Федеральное государственное автономное образовательное учреждение
высшего образования «Казанский (Приволжский) федеральный университет
«КФУ»

На правах рукописи

Алимова Ильсеяр Салимовна

Нейросетевой механизм кросс-внимания в задачах извлечения информации из текстов на примере биомедицинских данных

Специальность 05.13.11 —
«Математическое и программное обеспечение вычислительных машин,
комплексов и компьютерных сетей»

Диссертация на соискание учёной степени
кандидата технических наук

Научный руководитель:
кандидат физико-математических наук
Тутубалина Елена Викторовна

Казань — 2021

Оглавление

Стр.

Стр.

Стр.

Введение

В настоящее время в связи с бурным развитием сети Интернет и электронных коллекций научных публикаций накоплен огромный объем неструктурированной информации, представленной текстами на естественных языках. В связи с этим становится все более востребованной разработка автоматических методов обработки текстов с целью извлечения структурированных данных, которые в дальнейшем могут быть использованы для извлечение фактов, анализа мнений пользователей, поиска информации и других задач.

Важнейшими задачами автоматического извлечения информации из текстов являются классификация и извлечение отношений. Задача классификации текста заключается в группировке текстов по определенным заранее заданным классам. Задача классификации может рассматриваться на уровне документа, параграфа, предложения и сущности. Задача извлечения отношений заключается в определении семантических связей между двумя понятиями (сущностями).

Методы классификации текста и извлечения отношений применяются для решения широкого круга прикладных задач, включающих анализ мнений пользователей о продукте, фильтрации спама, подбор контекстной рекламы, автоматическое реферирование, тегирование контента на сайте, сортировка обращений пользователей в службы поддержки, построение лингвистических баз данных и т.д. Одной из наиболее значимых областей применения методов классификации и извлечения отношений являются задачи медицинской науки, в частности, задачи фармакологии и персонализированной медицины.

В области биомедицины классификация текста применяется для поиска новой информации о побочных реакциях лекарственных веществ, не указанных в инструкции, а также для обнаружения использовании лекарств с нарушением предписаний инструкции [1]. Задача извлечения отношений применяется для построения биомедицинских баз данных, определения действия лекарственных веществ по отношению к системам организма, излечения новых отношений между лекарствами и симптомами для построения гипотез. Информация, полученная в результате извлечения отношений, может также использоваться как часть входных данных для других задач, таких как извлечение событий, классификация диагнозов, принятие клинических решений и в вопросно-ответных системах.

Классификация текста и извлечение отношений между сущностями в биомедицинских текстах исследовались в трудах российских и зарубежных учёных, таких как Саркер А, Гонсалез Г., Гинн Р., Никфарджам А., Карими С., Патки А., Кириченко С., Лиу Х., Шелманов А., Браславский П. и других авторов. Перечисленными авторами разработаны основные теоретические аспекты анализа биомедицинских текстов на естественном языке. В работах [2—7] приведен развернутый обзор существующих автоматических методов классификации текста и поиска отношений между сущностями в биомедицинских текстах. Согласно рассмотренным исследованиям, подходы машинного обучения обладают большим потенциалом для исследуемых задач, однако для реальных биомедицинских приложений результаты необходимо улучшать [8]. Существующие методы разработаны для текстов определенного домена: социальные медиа, электронные медицинские карты (ЭМК) или научные тексты статей и тестируются в рамках одного текстового корпуса, что является существенным недостатком в вопросе применимости моделей для реальных практических задач [9]. Кроме того, исследования в основном ведутся для английского языка и почти отсутствуют работы для русского языка. Таким образом, подтверждена актуальность разработки методов классификации текста и извлечения отношений.

Объектом исследования являются неструктурированные тексты на естественном языке, включающие: отзывы пользователей о лекарственных препаратах, тексты твитов, аннотации научных статей, электронные медицинские карты (ЭМК). **Предметом** исследования выступают задачи классификации текста и извлечения отношений между сущностями.

Целью диссертационной работы является разработка методов и программных средств для классификации сущностей и извлечения отношений между сущностями из текстов. Разрабатываемые методы и программные средства должны удовлетворять следующим требованиям:

- более высокие оценки качества классификации сущностей и извлечения отношений предложенных методов по сравнению с существующими моделями;
- переносимость методов на тексты различных языков, в данной диссертационной работе рассматриваются тексты на русском и английском языках;

– переносимость методов на тексты различных доменов с различной языковой моделью, в данной работе рассматриваются тексты отзывов пользователей о лекарствах, твиты о здоровье, аннотации научных статей по теме биомедицины и ЭМК.

Для достижения поставленной цели необходимо было решить следующие **задачи**:

1. Разработать методы на основе нейронной сети с кросс-вниманием для задачи классификации сущностей;
2. Провести исследования для извлечения наиболее информативных признаков для задачи классификации сущностей;
3. Интегрировать наиболее информативные признаки в разработанную модель на основе нейронных сетей для задачи классификации сущностей;
4. Разработать методы на основе нейронной сети с кросс-вниманием для задачи извлечения отношений между сущностями;
5. Получить размеченные текстовые коллекции на русском и английском языках, состоящие из отзывов пользователей о лекарственных препаратах, твитов о здоровье, научных статей по теме биомедицины и ЭМК;
6. Реализовать предложенные методы в виде программного средства и провести экспериментальные исследования с целью определения качества работы методов и моделей с использованием коллекций текстовых документов.

Научная новизна данной диссертационной работы заключается в следующем:

1. Предложена модель классификации сущностей на основе нейронной сети с механизмом кросс-внимания, отличающаяся от существующих моделей набором дополнительных информативных признаков.
2. Предложена модель классификации отношений сущностей на основе нейронной сети с механизмом кросс-внимания, отличающаяся от существующих моделей разделением контекста и сущностей на отдельные подсети формирования контекстных векторных представлений.

Практическая значимость. В диссертации разработана программная система классификации сущностей и извлечения отношений, основанная на предложенных нейросетевых моделях с кросс-вниманием, которая предназначена для использования в качестве инструмента автоматического анализа

текстовых корпусов. Разработанная система может быть использована как для автоматического извлечения информации из биомедицинских текстов, так и из текстов другой тематики.

Методология и методы исследования. В данной диссертационной работе применялись методы обработки естественного языка, машинного обучения, теории вероятностей и оптимизации.

Основные положения, выносимые на защиту:

1. Предложен и реализован новый метод классификации сущностей, основанный на нейронной сети с механизмом кросс-внимания и набором информативных признаков.
2. Предложен и реализован новый метод извлечения отношений между сущностями, основанный на нейронной сети с механизмом кросс-внимания и с разделением контекста и сущностей на отдельные подсети формирования контекстных векторных представлений.
3. Разработано программное обеспечение CAFEC для задачи классификации сущностей и проведено экспериментальное исследование, обосновывающее улучшение качества предложенных методов по сравнению с существующими алгоритмами в рамках корпусов из одного домена и корпусов из разных доменов.
4. Разработан программный комплекс CARE для задачи извлечения отношений между сущностями и проведено экспериментальное исследование, обосновывающее улучшение качества предложенных методов по сравнению с существующими алгоритмами в рамках текстовых корпусов из разных доменов.

Достоверность подтверждается корректным применением выбранного математического аппарата, экспериментами, проведенными в соответствии с общепринятыми стандартами, взаимосвязью данных экспериментов и научных выводов, сделанных в работе, результатами апробации алгоритмов и разработанной программной системы.

Апробация работы. Основные результаты работы докладывались на следующих конференциях:

1. Летней школе по информационному поиску RuSSIR (г. Екатеринбург, Россия, 21–25 августа 2017 г.);
2. 6-й международной конференции по анализу изображений, сетей и текстов АИСТ (г. Москва, Россия, 27–29 июля 2017 г.);

3. 8-й открытой конференции ИСП РАН имени В.П. Иванникова (г. Москва, Россия, 30 ноября - 1 декабря, 2017 г.);
4. 52-й ежегодной конференции “ESCI Annual Scientific Meeting of the European Society for Clinical Investigation” (г. Барселона, Испания, 30 мая - 1 июня, 2018 г.);
5. Международной научной конференции “Artificial Intelligence and Natural Language Conference” (г. Санкт-Петербург, Россия, 17-19 октября, 2018 г.);
6. 9-й открытой конференции ИСП РАН имени В.П. Иванникова (г. Москва, Россия, 22 - 23 ноября, 2018 г.);
7. Восточноевропейской летней школе по машинному обучению EEML (г. Бухарест, Румыния, 1-6 июля, 2019 г.)
8. 8-й международной конференции по анализу изображений, сетей и текстов АИСТ (г. Казань, Россия, 17–19 июля 2019 г.);
9. 57-й конференции “Association for Computational Linguistics” (г. Флоренция, Италия, 28 июля - 3 августа 2019 г.);
10. 1-м международном саммите “EurNLP” (г. Лондон, Великобритания, 11 октября 2019 г.);
11. 22-й международной научной конференции “Data Analytics and Management in Data Intensive Domains” (г. Казань, Россия, 17-18 октября 2019 г.).

Личный вклад. Все представленные в диссертации результаты получены лично автором.

Публикации. Основные результаты по теме диссертации изложены в 14 печатных изданиях, 2 из которых изданы в журналах, рекомендованных ВАК, 10 — в периодических научных журналах, индексируемых Web of Science и Scopus, 2 — в тезисах докладов.

В работах [10—17] автором проведено исследование предметной области, выполнен основной объем теоретических и экспериментальных исследований, изложенных в публикациях, Тутубалиной Е.В. и Соловьеву В.Д. принадлежит постановка задачи и практические рекомендации для выполнения работы. В работе [18] автором диссертации был проведен анализ полученных результатов, осуществлен поиск соответствующих примеров и структурирование исследования. В работе [19] автором были проведены эксперименты по классификации твитов на наличие побочных эффектов. В работе [20] автор принимал участие

в аннотировании и подготовке финальной версии корпуса к публикации. В работах [21; 22] автором были проведены все экспериментальные исследования.

Объем и структура работы. Диссертация состоит из введения, четырех глав, заключения и одного приложения. Полный объём диссертации составляет 146 страниц, включая 22 рисунка и 28 таблиц. Список литературы содержит 182 наименования.

Глава 1. Анализ предметной области

Данная глава посвящена описанию основных подходов, применяемых для извлечения информации из текстов. Особое внимание уделяется рассмотрению существующих методов классификации и извлечения отношений из текстов биомедицинской тематики. Целью данной главы является анализ достоинств и недостатков существующих методов классификации текстов на уровне сущностей и извлечения отношений.

1.1 Методы автоматического извлечения информации из текста

Разработку системы извлечения информации из текста можно разделить на следующие четыре этапа: препроцессинг текста, извлечение признаков, выбор модели и оценка. В качестве входных данных модель принимает набор необработанных текстовых данных. Как правило, наборы текстовых данных содержат последовательности текстовых документов $D = \{X_1, X_2,..., X_N\}$, где X_i соответствует экземпляру входных данных (т. е. документу, текстовому сегменту). Препроцессинг текста включает в себя удаление шумовых данных, разбивку текста на токены и приведение слов к нормальной форме. На шаге извлечения признаков текстовые данные переводятся в числовые значения, которые характеризуют исходные данные. Один из самых важных шагов в задаче извлечения информации - это выбор лучшей модели. На последнем шаге проводится оценка качества разработанной системы извлечения информации.

Рассматриваемые в данной работе задачи относятся к задачам классификации текста, поэтому в данном разделе будет рассмотрен алгоритм построения классификатора текста. В задаче классификации текста каждый экземпляр помечен значением класса из набора k различных индексов с дискретными значениями.

1.1.1 Предварительная обработка текстовых данных

Предварительная обработка текстов (препроцессинг) - важные шаги для задачи классификации текста. В этом разделе представлены методы очистки наборов текстовых данных, которые удаляют неявный шум и позволяют извлечь более качественные признаки. Предварительная обработка текстов включает в себя следующие основные шаги: токенизация, удаление стоп слов, нормализация.

Токенизация - это метод предварительной обработки, который разбивает текст на слова, фразы, символы или другие значимые элементы, называемые токенами. Данный шаг применяется в большинстве задач автоматической обработки текстов.

Входные тексты и документы включают в себя множество слов, не имеющих важного значения для использования в алгоритмах классификации, таких как “а”, «примерно», «выше», «поперек», «после», «снова» и т.д. Самый распространенный метод - это удаление их из текстов и документов.

В тексте одно слово может появляться в разных формах (в единственном и множественном числе, в разных падежах, родах и т.д.), в то время как семантическое значение каждой формы одинаково. Нормализация заключается в приведении слов с разными словоформами, но с одинаковыми семантическими смыслами, в единую форму. Существует два основных способа нормализации: лемматизация и стемминг. При лемматизации слова приводятся основную словоформу: единственное число, именительный падеж для существительного и прилагательного, инфинитив для глагола и т.д. При нормализации из слова удаляется окончание и остается только основа слова.

1.1.2 Извлечение признаков

Существует множество исследований посвященных методам извлечения признаков из текста с целью решить проблему потери синтаксической и семантической связи между словами. Рассмотрим наиболее известные подходы

извлечения признаков, основанные на взвешивании слова в документе и векторном представлении слов.

Методы на основе взвешивания слов представляют каждый документ в виде вектора с длинной равной количеству слов в словаре корпуса. Вектор содержит значения веса слов в данном документе. Основной недостаток данного подхода в том, что наиболее распространенные слова языка могут получить больший вес в сравнении с другими словами.

Модель мешок слов (bag-of-words; BoW) - это сокращенное и упрощенное представление текстового документа на основе частоты слов. В данном методе подсчитывается количество вхождений каждого слова из словаря, где словарь является набором уникальных слов всех текстов обучающей выборки. Модель не учитывает порядок слов в тексте, грамматику и семантические отношения между словами, что является одним из ее главных недостатков. Кроме того, итоговый вектор, представляющий текст имеет большую размерность.

Модификацией подхода мешок слов является метод TF-IDF (Term Frequency-Inverse Document Frequency). Данный метод учитывает важность слова в контексте документа. Вес слова пропорционален частоте употребления этого слова в документе и обратно пропорционален частоте употребления слова во всех документах коллекции. Метрика вычисляется как произведение частоты термина в документе на обратную частоту термина во всех документах коллекции:

$$W(d,t) = \frac{n_t}{\sum_k n_k} \cdot log(\frac{N}{df(t)})$$

В формуле n_t - число вхождений слова t в документ, в знаменателе первого множителя - общее число слов в данном документе, N - количество документов в коллекции, $df(t)$ - количество документов, содержащих термин t. Не смотря на то, что метод TF-IDF решает проблему терминов, встречающихся во всех документах, у данного метода существуют другие недостатки. В частности, TF-IDF не учитывает сходство между словами в документе, поскольку каждое слово независимо представлено в виде индекса.

Методы на основе взвешивания слов в документе не учитывают семантику слов, то есть синонимичные друг другу слова в данном представлении будут представлены векторами далекими друг от друга в векторном пространстве. Эта особенность представляет серьезную проблему для понимания семантики предложения. Другая проблема рассмотренных ранее моделей - порядок слов

не учитывается. Для решения данных проблем были разработаны модели на основе векторного представления слов.

Векторное представление слов - это метод, при котором каждому слову или фразе из словаря сопоставляется вектор действительных чисел размерности N. Были предложены различные методы векторного представления слов для перевода слов в понятный для алгоритмов машинного обучения вход. Наиболее распространенные методы перевода слов в векторное представление: Word2Vec, Glove, FastText. В последнее время все большее распространение получают контекстно-зависимые модели векторного представления слов, такие как ELMo и BERT.

Подход Word2Vec основан на нейронной сети с одним скрытым слоем. Существует две вариации данного метода: Continuous bag-of-words (CBOW) и Skip-gram. В модели CBOW решается задача предсказания слова по его контексту. В модели Skip-gram по заданному слову предсказывается его контекст. На вход модели подаются слова, закодированные в виде “one-hot” представления. При таком представлении каждое слово кодируется бинарным вектором длины словаря, где на месте с порядковым номером данного токена в словаре стоит единица, а все остальные значения вектора равны нулю. Сеть проходит по всему тексту и учится строить предсказания для каждого локального контекста с заданным размером окна (количество слов слева и справа от предсказываемого слова). В качестве итоговых векторных представлений слов используются скрытые состояния сети.

В модели Glove предпринята попытка решить проблему прохождения по всем окнам локального контекста подряд и тем самым сжатия входных данных для подсчета общих статистик. Основная идея метода Glove заключается в приближении матрицы встречаемости слов. Обучающая функция выглядит следующим образом:

$$F(w_i, w_j, \widetilde{w}_k) = \frac{p_{ik}}{p_{jk}}$$

где w_i - вектор представления слова i, p_{ik} - вероятность появления слова i в контексте слова k.

Рассмотренные модели не решают проблему слов вне словаря (out of vocabulary; OOV), то есть если на вход модели будет подано слово, которого не было изначально в модели, вектор слова получить не удастся. В модели FastText решают данную проблему разбиением слов на символьные n-граммы.

Предположим, имеется словарь n-грамм размера G и дано слово w с векторным представлением z_g для каждой n-граммы z_g. Функция оценки в данном случае:

$$s(w,c) = \sum_{g \in g_w} z_g^T v_c$$

Описнанные модели не различают между собой разные значения слова и генерируют для них одинаковые вектора. Более современные векторные представления слов учитывают контекст при построении вектора для слова. Рассмотрим примеры таких моделей: ELMo (Embeddings from Language Models) и BERT (Bidirectional Encoder Representations from Transformers). Основная идея модели ELMo заключается в том, что результирующие векторы слов обучаются в двунаправленной языковой модели, основанной на сети с долгой краткосрочной памятью (bidirectional long short term memory; biLSTM). Общая формула для векторов модели ELMo вычисляется как сумма взвешенных скрытых слоев biLSTM и входного векторного представления слов, умноженная на весовой коэффициент специфичный для выбранной задачи:

$$ELMo_k^{task} = \gamma_k \cdot (s_0^{task} \cdot x_k + s_1^{task} \cdot h_{1,k} + s_2^{task} \cdot h_{2,k})$$

Здесь the s_i представляет нормализованные с помощью функции $softmax$ веса для скрытых слоев, γ_k специфичный для задачи коэффициент.

Векторное представление обучатся отдельно для каждой целевой задачи для которой модель будет использоваться. Для того, чтобы использовать модель в конкретной задаче, необходимо заморозить веса предобученной языковой модели и затем конкатенировать значения $ELMo_k^{task}$ для каждого токена с входным представлением конкретной задачи. таким образом, весовые коэффициенты γ_k и s_i подбираются в процессе обучения модели на специфичной задаче.

Модель BERT состоит из слоев кодировщика сети Трансформер. Слой кодировщика состоит из двух основных компонент: слоя самовнимания (self-attention) и слоя с прямой связью (feedforward). В слое с self-attention на основе каждого входного вектора создается три вектора: запрос (Q), ключ (K), значение (V). Эти векторы создаются умножением векторного представления слова на три матрицы (W^Q, W^K, W^V), которые подбираются в процессе обучения модели. На основе полученных векторов вычисляется вектор внимания:

$$softmax(\frac{QK^T}{\sqrt{d_k}})V$$

В данной формуле d_k - размер векторов ключа. Входной вектор подается нескольким идентичным слоям с self-attention. Такой подход позволяет распараллелить вычисления. Выходы слоев внимания конкатенируются и проходят через полносвязный слой.

На вход модели подаются слова, которые разбиваются с помощью алгоритма BPE ((Byte Pair Encoding)) на подслова. Данный алгоритм состоит из нескольких итераций на каждой из которых назначается новым токеном объединение двух существующих токенов, которые встречается совместно чаще других пар в корпусе. На первом шаге в качестве токенов берутся символы. Шаг со склеиванием токенов повторяется пока не достигнуто ограничение на размер словаря. Таким образом, решается проблема OOV.

Модель обучается на двух задачах моделирования языковой модели. Первая задача - предсказание замаскированных токенов. На вход сети подается текст, в котором какой-то процент слов заменен на служебный токен [MASK]. Сеть обучается предсказывать эти закрытые маской слова. Во второй задаче модель учится предсказывать является ли одно предложение логичным продолжением другого.

1.1.3 Существующие модели извлечения информации из текста

В этом разделе будут описаны классические методы на основе машинного обучения, такие как логистическая регрессия, наивный байесовский метод и метод опорных векторов. Далее будут рассмотрены алгоритмы классификации на основе деревьев, такие как дерево решений и случайный лес. Затем подходы на основе нейронных сетей.

Логистическая регрессия - один из самых ранних методов классификации [23]. Данная модель предсказывает вероятность принадлежности входного объекта к классу. В логистической регрессии строится линейный алгоритм классификации $a : X \to Y$, где X - множество входных объектов, Y - множество классов:

$$a(x, w) = sign(\sum_{j=1}^{n} w_j f_j(x) - w_0),$$

где w_j - вес j-го признака, w_0 - порог принятия решения, $w = (w_0, w_1, ..., w_n)$ - вектор весов. Задача обучения линейного классификатора заключается в том, чтобы по выборке X^m настроить вектор весов w. В логистической регрессии для этого решается задача минимизации с функцией потерь вида:

$$Q(w) = \sum_{i=1}^{m} ln(1 + exp(-y_i \langle x_i, w \rangle)) \underset{w}{\to} min$$

После того, как решение w найдено, становится возможным не только вычислять класс $a(x) = sign\langle x,w\rangle$ для произвольного объекта x, но и оценивать апостериорные вероятности его принадлежности классам:

$$P(y|x) = \sigma(y\langle x, w\rangle), y \in Y$$

где $\sigma(z) = \frac{1}{1+e^{-x}}$ - сигмоидная функция.

Логистическая регрессия хорошо подходит для прогнозирования категориальных результатов, однако требуется, чтобы каждая точка входных данных была независимой.

Наивный байесовский метод широко используется для задач категоризации документов с 1950-х годов [24]. Метод наивного байесовского классификатора теоретически основан на теореме Байеса и представляет собой генеративную модель, которая является наиболее традиционным методом категоризации текста. Формула для вычисления результирующего класса для текста:

$$c_{map} = \underset{c \in C}{argmax} \left[P(c) \prod_{i=1}^{n} P(w_i|c) \right]$$

где $P(c)$ — безусловная вероятность встретить документ класса c в корпусе документов, $P(w_i|c)$ — вероятность встретить слово w_i среди всех слов документов класса c, C - множество всех классов.

Одним из наиболее важных недостатков наивного байесовского классификатора является предположение о независимости признаков, поданных на вход, поскольку в тексте слова зависят от контекста.

Метод на основе опорных векторов (Support Vector Machines; SVM) был разработан Вапником и Червоненкисом в 1963 г [25]. Изначально SVM был разработан для задач бинарной классификации. Однако многие исследователи применяют данный метод для мультиклассовых задач, используя технику “один

против всех". Основная идея алгоритма заключается в построении гиперплоскости, заданной уравнением $\langle\vec{w},\vec{x}\rangle - b = 0$, такой, чтобы:

$$\begin{cases} \langle\vec{w},\vec{x}\rangle - b > 0, & \forall x \in C_1 \\ \langle\vec{w},\vec{x}\rangle - b > 0, & \forall x \in C_2 \end{cases}$$

где $\vec{x} = (x_1, x_2, ..., x_n)$ - вектор признаков объекта, $\vec{w} = (w_1, w_2, ..., w_n) \in \mathbb{R}^n$ и $b \in \mathbb{R}$ - обучаемые параметры гиперплоскости, C_1 и $_2$ - заданные классы.

Одним из недостатков SVM является неустойчивость к шуму: выбросы в исходных данных становятся объектами-нарушителями и напрямую влияют на построение разделяющей гиперплоскости.

Дерево решений (Decision Tree) для задач классификации было предложено Д. Морганом [26] и разработано Дж. Р. Куинланом [27]. Структура этого метода представляет собой иерархическую декомпозицию пространства входных данных. Набор правил по которым осуществляется декомпозиция можно представить в виде дерева. Основная проблема дерева решений заключается в выборе признаков, которые будут располагаться на каждом уровне дерева. Чтобы решить эту проблему было введено статистическое моделирование для выбора признаков в дереве. Для обучающего набора, содержащего p положительных и n отрицательных примеров:

$$H\left(\frac{p}{n+p}, \frac{n}{n+p}\right) = -\frac{p}{n+p} log_2\left(\frac{p}{n+p}\right) - \frac{n}{n+p} log_2\left(\frac{n}{n+p}\right)$$

Необходимо выбрать критерий A с k различными значениями, делящий обучающую выборку на подвыборки $\{E_1, E_2, ..., E_k\}$. При этом ожидается, что энтропия (EH) сохранится:

$$EH(A) = \sum_{k=1}^{K} \frac{p_i + n_i}{p+n} H\left(\frac{p}{n+p}, \frac{n}{n+p}\right)$$

Прирост количества информации (I) или уменьшение энтропии для этого атрибута составляет:

$$A(I) = H\left(\frac{p}{n+p}, \frac{n}{n+p}\right) - EH(A)$$

Критерий с наибольшим приростом количества информации выбирается как родительский.

Дерево решений очень быстрый алгоритм как для обучения, так и для прогнозирования. Однако данный алгоритм также чувствителен к небольшим выбросам в данных и может легко переобучаться. Данный алгоритм также показывает низкое качество предсказания для экземпляров вне обучающей выборки.

Метод случайных лесов (Random Forest; RF) основан на ансамбле деревьев решений и был разработан Т. Кам Хо [28] в 1995 году. После обучения всех деревьев финальное предсказание вычисляется с помощью функции голосования:

$$\sigma_V = \underset{i}{argmax} \sum_{j:j \neq j} I_{r_{ij} > r_{ji}}$$

при этом $r_{ij} + r_{ji} = 1$, а I - индикаторная функция.

Случайные лес очень быстро обучается для наборов текстовых данных по сравнению с другими методами, в частности, методами на основе глубокого обучения, но довольно медленно предсказывает метку класса после обучения. Таким образом, чтобы получить более быстрый алгоритм, количество деревьев в ансамбле необходимо уменьшить, так как большее количество деревьев в лесу увеличивает временную сложность на этапе прогнозирования.

Глубокие нейронные сети (Deep Neural Network; DNN) предназначены для обучения путем множественного соединения слоев так, что каждый отдельный слой соединен только с предыдущим и соединяется только со следующим [29]. Входные данные состоят из конкатенации входных векторов признаков с первым скрытым слоем DNN. Входной слой может быть создан с помощью TF-IDF, векторного представления слов или какого-либо другого метода извлечения признаков. Выходной слой равен по размеру количеству классов для мультиклассовой классификации или состоит из одного нейрона в случае бинарной классификации.

Реализация DNN - это дискриминативно обученная модель, которая использует стандартный алгоритм обратного распространения ошибки с использованием функций активации, например, сигмоиды или ReLU. Выходной слой для мультиклассовой классификации чаще всего имеет функцию активации softmax.

Для заданных пар $(x, y), x \in X, y \in Y$ необходимо чтобы сеть в процессе обучения установила взаимосвязь между входными и целевыми пространствами с помощью скрытых слоев. В задачах классификации текста входными данными является векторизованная текстовая строка.

Рекуррентная нейронная сеть (Recurrent Neural Network; RNN) является одной из разновидностей глубоких нейронных сетей [30; 31]. RNN способна учитывать информацию о предыдущих входных данных, что дает преимущество этому методу при обработке данных, представленных последовательностями. Таким образом, данный метод является мощным методом для классификации текста, поскольку учет предыдущих слов в тексте позволяет получить больше информации о семантике текста. Общая формула, представляющая принцип работы RNN:

$$x_t = F(x_{t-1}, u_t, \theta)$$

где x_t - состояние сети в момент времени t, u_t - вход сети в момент времени t. Представленную формулу можно переписать в терминах весов слоев следующим образом:

$$x_t = W_{rec}\sigma(x_{t-1}) + W_{in}u_t + b$$

где W_{rec} - рекуррентная матрица весов, W_{in} - матрица весов входного слоя, b - матрица сдвига, σ - обозначает поэлементную функцию.

Обычные RNN плохо запоминают информацию в длинных входных последовательностях, поскольку влияние скрытого состояния и входа на последующие состояния рекуррентной сети экспоненциально затухает. Существует две основные разновидности RNN, решающие данную проблему, - это сеть с долгой краткосрочной памятью и управляемый рекуррентный блок.

Сеть с долгой краткосрочной памятью (Long Short Term Memory; LSTM) была представлена Хохрайтером и Шмидхубером [32]. Сеть с короткой долгосрочной памятью - особая разновидность рекуррентных сетей, способная к обучению долговременным зависимостям. Поскольку слова в предложении сильно зависят друг от друга способность запоминать предыдущие состояния (слова) позволяет LSTM слоям идентифицировать эти зависимости и благодаря этому лучше интерпретировать семантику текста. Формально, для данного входного слова w_k, предыдущего состояния ячейки c_{k-1} и предыдущего скрытого состояния h_{k-1} текущее состояние сети c_k и скрытого слоя h_k вычисляется по следующим формулам:

$$i_k = \sigma(W_i^w \cdot w^k + W_i^h \cdot h^{k-1} + b_i)$$

$$f_k = \sigma(W_f^w \cdot w^k + W_f^h \cdot h^{k-1} + b_f)$$

$$o_k = \sigma(W_o^w \cdot w^k + W_o^h \cdot h^{k-1} + b_o)$$
$$\hat{c}^k = tanh(W_c^w \cdot w^k + W_c^h \cdot h^{k-1} + b_c)$$
$$c^k = f^k \odot c^{k-1} + i^k \odot \hat{c}^k$$
$$h^k = o^k \odot tanh(c^k)$$

где i - входной вектор f - вектор забывания, хранящий вес старой информации и o - выходной вектор обеспечивают взаимодействие между ячейками памяти и их окружением. σ обозначает логистическую функцию (сигмоиду). W и b обозначают веса матрицы и смещений слоя. Символ $\cdot$ обозначает стандартное умножение матриц, символ $\odot$ - поэлементное умножение матриц. На выходе слой выдает скрытые состояния $h = [h_1, h_2, ..., h_n]$.

Сеть с управляемым рекуррентным блоком (Gated Recurrent Unit; GRU) имеет особый механизм вентилей для рекуррентных нейронных сетей, представленный в 2014 году [33]. По сравнению с LSTM у данного механизма меньше параметров, так как отсутствует выходной вентиль. Кроме того, GRU отличается от LSTM тем, что не фильтрует внутреннюю память. Допустим, e_{i-1} -вектор, полученный на предыдущем шаге слоя GRU и t_i - текущее состояние, полученное применением механизма внимания к вектору памяти, тогда текущий вектор e_i будет высчитываться по формулам:

$$r = \sigma(W_r t_i + U_r e_{i-1})$$
$$z = \sigma(W_z t_i + U_z e_{i-1})$$
$$\tilde{e_i} = tanh(W_x t_i + W_g(r \odot e_{i-1}))$$
$$e_i = (1 - z) \odot e_{i-1} + z \odot \tilde{e_i}$$

где $W_r, W_z, U_r, U_z, W_g, W_x$ - веса слоя GRU. В качестве e_0 используется вектор из нулей.

Сверточная нейронная сеть (Convolutional Neural Networks; CNN) широко применяется для классификации текста [34]. На вход сети подается матрица, состоящая из слов теста, закодированных векторным представлением слов. Полученная матрица последовательно проходит через слои свертки и пулинга. Каждый слой свертки состоит из фильтра – матриц небольшой размерности (от 3×3, до 7×7). Фильтр последовательно скалярно умножается на фрагменты исходной матрицы, в результате чего образуется новая матрица,

называемая картой признаков. Таким образом, каждый нейрон в следующем слое связан не со всеми, а только с небольшим локализованным подмножеством нейронов в предыдущем слое, что позволяет выделить наиболее значимые признаки для каждого из фрагментов входной матрицы. Формально, свертка - это линейное преобразование входных данных. Если x^l - матрица признаков в слое под номером l, то результат двумерной свертки с ядром размера $2d+1$ и матрицей весов W размера $(2d+1) \times (2d+1)$ на следующем слое будет вычисляться по формуле:

$$y^l_{i,j} = \sum_{-d<=a,b<=d} W_{a,b} x^l_{i+a,j+b}$$

Для уменьшения вычислительной сложности, в CNN используются пулинговый слой (pooling layer), который уменьшает размер выхода предшествующего слоя. В пулинговом слое входная матрица делится на блоки заданного размера, и для каждого блока вычисляется некоторая функция. Чаще всего используется функция максимума или (взвешенного) среднего. Последний слои в CNN обычно полносвязный и в нем проводится классификация.

Модели на основе архитектуры **Трансформер** являются наиболее современными моделями для решения различного рода задач автоматической обработки текста, в том числе для классификации текста. Для построения классификатора на основе модели Трансформер в архитектуру сети добавляется финальный полносвязный слой, который получает на вход предобученные вектора из последнего слоя Трансформера. Архитектура модели Трансформер описана в разделе 1.1.2. Наиболее распространенная модель на базе архитектуры Трансформер - BERT. В данной модели в качестве векторного представления текста чаще всего используется вектор для служебного токена [CLS], который добавляется перед каждым входным текстом и формирует семантическое представление всего текста. Существует два способа обучения данной модели. В первом способе обучаются все веса модели, в том числе, и сети Трансформер. Такой способ более затратен по времени и вычислительным ресурсам, но при этом показывает более высокое качество. Во втором способе обучается только последний поносвязный слой, а веса трансформера остаются неизмененными. В данном случае модель обучается существенно быстрее, поскольку имеет меньшее количество параметров, но при этом качество модели существенно ниже, чем при первом способе обучения.

Одним из недостатков нейросетевых моделей является сложность интерпретации результатов. Еще один недостаток нейронных сетей заключается в том, что для них обычно требуется гораздо больше данных, чем для традиционных алгоритмов машинного обучения. Кроме того, огромный объем данных, необходимых для алгоритмов классификации нейронных сетей, повышает вычислительную сложность на этапе обучения. Несмотря на перечисленные недостатки, модели извлечения информации из текста, основанные на нейронных сетях, показывают более высокое качество в сравнении с классическими моделями классификации.

1.1.4 Оценка качества моделей

В данной секции будут рассмотрены наиболее распространенные способы определения качества моделей классификации - метод перекрестной проверки и метрики оценки классификации.

Обучение модели и оценка качества на одних и тех же данных может привести к ситуации, когда модель переобучается, то есть предсказывает метки класса с высоким качеством на используемых для обучения данных, но при этом на новых данных качество модели существенно снижается. Для решения данной проблемы набор данных делится на обучающие и тестовые данные. При этом риск остается переобучения модели на тестовых данных, поэтому в данных выделяется еще один набор - валидационных данных, на которых происходит подбор наилучших параметров модели. Однако при разбиении доступных данных на три части, сокращается количество данных, которые можно использовать для обучения модели, и результаты могут зависеть от конкретного случайного разбиения данных. Решением этой проблемы является процедура перекрестной проверки (cross-validation; CV). Данные разбиваются на k частей (фолдов). Модель оценивается на $k - 1$ частях и оценивается на оставшейся части данных. Процедура повторяется k раз, при этом модель оценивается каждый раз на разных частях данных. Итоговая метрика качества подсчитывается как усредненная метрика по всем частям данных. Такой подход имеет более высокую вычислительную сложность, но при этом данные задействуются более эффективно в отличие от подхода с фиксированным разбиением данных.

Таблица 1 — Матрица ошибок. y - истинная метка класса на объекте, $\hat{y}$ - ответ классификатора на этом объекте.

	$y = 1$	$y = 0$
$\hat{y} = 1$	истинно положительные (TP)	ложно положительные (FP)
$\hat{y} = 0$	ложно отрицательные (FN)	истинно отрицательные (TN)

В число стандартных метрик качества классификации входят: P - точность (precision), R - полнота (recall), F-мера (англ. F_1 - мера) и доля правильных ответов (Accuracy). Точность показывает на сколько можно доверять классификатору, полнота означает сколько экземпляров класса может выделить классификатор, F-мера является средним гармоническим между полнотой и точностью, доля правильных ответов показывает процент правильных ответов классификатора.

Для определения описанных метрик используется матрица ошибок (confusion matrix), представленная в таблице 1.

Полнота вычисляется как отношение истинно-положительных документов к общему количеству известных положительных документов по формуле:

$$R = \frac{TP}{TP + FN}$$

Точность вычисляется как отношение истинно-положительных релевантных документов к общему количеству определенных системой релевантных документов:

$$P = \frac{TP}{TP + FP}$$

F-мера рассчитывается по формуле:

$$F = \frac{2PR}{P + R}$$

Формула доли правильных ответов:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Метрика Accuracy не отражает реальное качество классификации в случае, когда в наборе данных присутствует дисбаланс классов. Метрики полноты и точности, в отличие от Accuracy, не зависят от соотношения классов и потому применимы в условиях несбалансированных выборок.

1.2 Классификация текстов биомедицинской тематики

Большинство работ по классификации текстов биомедицинской тематики посвящены извлечению упоминаний о побочных эффектах. Классификацию текстов биомедицинской тематики рассматривать в двух направлениях: (i) на уровне сообщения и (ii) на уровне сущности. В первом случае необходимо определить наличие упоминания побочного эффекта во фрагменте текста, например, предложении или тексте твита. Данный тип классификации необходим для очистки коллекции текста от нерелевантных документов. Во втором случае классификация применяется к результатам работы алгоритмов извлечения именованных сущностей.

1.2.1 Методы на основе словарей и правил

В исследованиях применяются различные подходы для классификации текста с целью выявления упоминаний побочных реакций. В ранних работах применялся подход, основанный на словарях [35—41]. Словари состоят из списков побочных реакций, извлеченных из инструкций по применению лекарств, записей о клинических испытаниях , отзывах пользователей в социальных сетях. Среди них наиболее известными и используемыми для английского языка являются: FAERS[1], COSTART[2], CHV[3], MedEffect[4], UMLS[5], MedDRA[6], SIDER[7]. Первые работы были ограничены количеством исследуемых лекарств и целевых побочных эффектов из-за ограничений терминов в словарях. Для преодоления этого ограничения начали использоваться методы на основе правил [41; 42]. Основная идея этих методов заключается в том, чтобы выделить наиболее распространенные конструкции предложений, которые могут свидетельствовать об описании побочных реакций. Однако, разработка правил является длительным

[1]http://www.fda.gov/Drugs/GuidanceComplianceRegulatoryInformation/Surveillance/AdverseDrugEffects
[2]http://bioportal.bioontology.org/ontologies/COSTART
[3]http://www.consumerhealthvocab.org/
[4]http://www.hc-sc.gc.ca/dhp-mps/medeff/index-eng.php
[5]http://www.nlm.nih.gov/research/umls/
[6]http://www.meddra.org/
[7]http://sideeffects.embl.de/

и трудоемким процессом, требующим наличие специалиста в данной предметной области и не масштабируемым для новых коллекций документов.

1.2.2 Методы на основе машинного обучения

Большинство работ описывают исследования с использованием методов машинного обучения. Например, в работах [43—49] используется метод опорных векторов (support vector machine; SVM), в статьях [41; 50] применяется метод условных случайных полей (conditional random fields; CRF), а в работе [51] метод случайных лесов (Random Forest; RF). В качестве признаков, подаваемых на вход алгоритмам машинного обучения, используются:

- n-граммы,
- части речи,
- принадлежность к семантическим типам из словаря UMLS,
- количество слов с отрицанием,
- принадлежность рассматриваемого термина к словарям побочных реакций,
- наличие в тексте названия лекарства,
- вектора Word2Vec,
- вектора кластеризации.

Ню и другие предложили классификатор на основе SVM со следующим набором признаков: униграммы и биграммы; наличие фраз, указывающих на изменения (увеличился, усилился, уменьшился, снизился и т.д.); наличие отрицания в тексте; категории из тезауруса UMLS [43]. Модель тестировалась на текстах отчетов клинических испытаний лекарственных препаратов. В работе рассматривалось 4 класса: позитивный (симптомы заболевания уменьшились или исчезли), нейтральный (существенных улучшений не замечено), негативный (обнаружились побочные эффекты) и отсутствие результата. Модель показала 73.13% F-меры для класса побочных эффектов. Ян и другие применили классификатор на основе SVM для извлечения информации о побочных эффектах из текстов отзывов пользователей о лекарственных препаратах с форумов ProzacAwareness и SSRIsex [46]. Классификатору подавались на вход три вида признаков: синтаксические, семантические и тональные. Кроме того,

в работе был применен метод обучения с частичным привлечением учителя. Разработанный классификатор показал 88.94% и 89.74% F-меры на корпусах ProzacAwarenes и SSRIsex соответственно при использовании 40% положительных примеров для обучения. В работе Патки и других проводилась бинарная классификация 10617 комментариев пациентов о лекарственных препаратах, собранных и размеченных вручную с ресурса DailyStrength с целью определить наличие или отсутствие побочных эффектов [49]. В качестве модели использовалась комбинация классификаторов, основанных на SVM и наивном байесовском классификаторе (Naive Bayesian Classifier; NB). Модель достигла 54% F-меры для положительного класса (побочный эффект в тексте присутствует) и 85.2% F-меры для отрицательно класса (побочный эффект отсутствует). На основе полученных результатов классификатора следующим шагом проводилась идентификация лекарственных препаратов с наиболее высоким количеством побочных эффектов.

1.2.3 Методы на основе нейронных сетей

В последнее время все большее распространение получают методы, основанные на нейронных сетях. В большинстве работ использовалась сверточная нейронная сеть (CNN) [52—56]. В работе [52] использовались четыре архитектуры, основанные на CNN: классическая CNN с одним сверточным слоем; сверточная рекуррентная нейронная сеть (CRNN) и рекуррентная сверточная нейронная сеть (RCNN), полученные конкатенацией сверточной нейронной сети с рекуррентной нейронной сетью в разной последовательности; сверточная нейронная сеть с вниманием (CNNA). Методы тестировались на корпусе твитов и корпусе текстов отчетов, собранных с ресурса MEDLINE. Эксперименты показали, что все методы превосходят результаты базового метода, основанного на NB классификаторе. Наиболее высокие результаты среди предложенных в статье архитектур нейронных сетей показала классическая CNN: 51% и 87% F-меры на корпусе твитов и MEDLINE соответственно. Шен и др. разработали многоканальную сверточную нейронную сеть для извлечения упоминаний о побочных эффектах из текстов социальных медиа [55]. На вход сети подавались два вида векторных представлений: на уровне слов и на уровне символов, кото-

рые проходили через сверточные слои. Дополнительно на вход сети подавались извлеченные из текста признаки: количество сущностей в предложении, мешок слов и тональные. Модель показала 74.4% F-меры на корпусе твитов из [47].

Становски и другие применили рекуррентную нейронную сеть, интегрированную с данными из DBPedia и Blekko [57]. Разработанная модель показала 83.9% и 93.4% F-меры на корпусах RASCAL и CADEC соответственно. Существует ряд работ, применяющих классическую схему для извлечения сущностей, состоящую из комбинации сети с долгой краткосрочной памятью и метода условных случайных полей (BiLSTM+CRF) [58—60]. Танг и другие протестировали модель BiLSTM+CRF для извлечения побочных эффектов на корпусе CADEC, состоящем из отзывов пользователей о лекарственных препаратах с ресурса askapatient.com [60]. Помимо векторного представления слов на вход модели подавались признаки, основанные на словарях побочных эффектов SIDER и ADRMine. Разработанная модель достигла 66.32% F-меры. Тутубалина и Николенко улучшили результаты на корпусе CADEC, добавив на вход модели BiLSTM+CRF посимвольное векторное представление слов, полученное с помощью CNN [58]. Данная модель достигла 70.65% F-меры. В работе Вунава и других модель BiLSTM+CRF достигла 63.51% F-меры на корпусе электронных карточек пациентов для задачи извлечения сущностей, обозначающих побочные эффекты [59]. Существуют исследования по применению рекуррентных нейронных сетей с механизмом внимания [61; 62]. Пандей и другие применили двунаправленную рекуррентную нейронную сеть с вниманием в комбинации с CRF для извлечения побочных эффектов [61]. Модель тестировалась на текстах с ресурса MEDLINE и электронных карточек пациентов из корпуса CRIS. Нейронная сеть достигла 82.68% и 87.16% F-меры на корпусах MEDLINE и CRIS соответственно. Рамамурфи и Муруган решали задачу извлечения побочных эффектов в два этапа: на первом этапе модель извлекала сущности из текста, на втором этапе сущность проходила через бинарный классификатор, определяющий, является ли она описанием побочного эффекта [62]. Для решения данной задачи в работе была использована двунаправленная сеть с долгой краткосрочной памятью и механизмом внимания. На вход сети подавались векторные представления слов, части речи и посимвольные векторное представления слов. Эксперименты проводились на корпусе размеченных аннотаций научных статей, собранных с ресурса PubMed. Модель показала 86.78% F-меры.

1.2.4 Классификация твитов по теме биомедицины

Существует большое количество работ по классификации текстов пользователей Твиттера о здоровье [19; 39; 45; 48; 51; 52; 63—68]. В работе [48] был представлен корпус из 10822 размеченных твитов, а также два метода бинарной классификации на основе наивного байесовского классификатора и метода опорных векторов. Методы тестировались на трех наборах данных: сбалансированный, когда количество экземпляров класса было одинаково, несбалансированный с 60% и с 70% твитов, не содержащих побочный эффект. Тексты представлялись в виде мешка слов. На самом несбалансированном наборе данных наивный байесовский классификатор показал наибольший результат 64.6% F-меры для положительного класса (побочный эффект присутствует в тексте) и 72.7% средней F-меры по двум классам. Вторая модель достигала 52.9% F-меры для класса положительного класса и 76.6% средней F-меры. В работе [39] был собран корпус из 6.9 миллионов твитов. Классификация проводилась на основе словарного подхода. Предложенный классификатор показал результат 78.4% F-меры.

В работе Биан и др. был собран корпус из двух миллиардов твитов [45]. С помощью методов, основанных на словарях, из корпуса были выделены твиты, относящиеся к теме здоровья. Для проверки качества модели были выделены твиты о 5-ти лекарственных препаратах от рака: Авастин, Мелфалан, Рупафин, Тамоксифен, Доцетаксел. Для классификации использовалась модель на основе метода опорных векторов, со следующими признаками: мешок слов, количество хештегов, тегов, обозначающих ответ на твит, слов с отрицанием, ссылок, предлогов, названий лекарств или их синонимов, терминов с уникальными идентификаторами словаря UMLS в каждом семантическом типе и каждой семантической группе. Разработанная модель достигла точности 74%.

Плачоурас и др. собрали собственную коллекцию твиттов и разработали классификатор, основанный на методе опорных векторов с набором признаков в виде n-грамм, стандартных признаков для Твиттера и частеречных признаков [69]. Данный подход показал 60.4% F-меры.

С 2016-го года в соревновании SMM4H проводится задача по поиску побочных эффектов в сообщениях из Твиттера [70—73]. В рамках соревнования присутствовали задачи классификации на уровне всего твита и на уровне сущ-

ности. Победители первого соревнования использовали комбинацию из девяти классификаторов, основанных на модели RF [51] со следующим набором признаков: 1, 2, 3 - граммы, появление вместе лекарства и побочного эффекта в тексте, наличие отрицания и оценка тональности. В качестве набора данных для каждого классификатора на вход подавались все положительные примеры и такое же количество случайным образом выбранных отрицательных примеров, что позволило участникам решить проблему несбалансированности классов. Описанная система получила 41.95% F-меры. Участники, занявшие второе место использовали также ансамбль классификаторов, состоящий из четырех моделей: (1) классификатор, основанный на словаре побочных эффектов, (2) метод максимальной энтропии с n-граммами и метрикой TF-IDF в качестве признаков, (3) метод максимальной энтропии с n-граммами и коэффициентами NB классификатора (4) метод максимальной энтропии с распределенным представлением слов в качестве признаков [63]. Данный подход достиг 41.82% F-меры. На третьем месте оказалась система, использовавшая метод опорных векторов с мешком слов, признаками тональности, концептами медицинского тезауруса UMLS и наличием смайлов в качестве признаков [64]. Данная система получила 35.8% F-меры. На четвертом месте оказалась модель, использовавшая также метод опорных векторов с n-граммами, словарями, полярностью в качестве признаков, а также признаками на основе тематического моделирования. Модель показала 33% F-меры.

Дальнейшие эксперименты на корпусе твитов соревнования позволили улучшить результаты, достигнутые участниками. В работе [52] применялись сверточная рекуррентная нейронная сеть и сверточная сеть с вниманием. Эксперименты проводились на двух наборах данных: твитов из соревнования SMM4H 2016-го года [70] и отчетов системы MEDLINE [74]. Сверточная рекуррентная нейронная сеть показала 51% F-меры на корпусе твитов и 87% F-меры на корпусе MEDLINE, модель с вниманием показала 49% и 83% F-меры соответственно. Таким образом, был получен прирост на 9% в сравнении с лучшими результатами соревнования. Ли и др. применил ансамбль сверточных нейронных сетей с частичным обучением на твитах, содержащих побочные эффекты, статьях Википедии о лекарственных препаратах и текстах из метатезауруса UMLS. Данный подход превзошел предыдущий на 13.5% F-меры. Даи и другие применили метод опорных векторов с набором лингвистических, тональных, словарных признаков, а также признаков, основанных на тематическом мо-

делировании [75]. Самый высокий результат модели - 55.9% F-меры. Однако, провести сравнительную оценку данного метода с участниками соревнования не представляется возможным, так как результаты представлены на обучающем корпусе соревнования, а не на тестовом.

В соревновании SMM4H 2017-го года в задаче классификации на уровне твитов первое место заняла система, использовавшая метод опорных векторов в качестве модели [65]. Однако, в отличие от предыдущего года, набор признаков был более обширным: n-граммы, доменно-зависимые векторные представления слов и кластеры, наличие отрицания, специфичные признаки для анализа твитов (наличие слов в верхнем регистре, намеренное дублирование букв в слове, наличие смайлов), признаки, основанные на различных словарях побочных эффектов, набор базовых тональных признаков. Модель получила 43.5% F-меры и таким образом улучшила результаты предыдущего соревнования на 1.55%. В задаче классификации на уровне сущностей лучшие результаты показала система, использовавшая ансамбль сверточных нейронных сетей [66]. Система достигла 69.3% F-меры.

В 2018-м году лучшие результаты по извлечению побочных эффектов на уровне твитов получили системы на основе нейронных сетей [72]. Первое место заняла система с использованием сверточной нейронной сети для представления твита и технологии “multi-head self-attention” [67]. Система получила 52.2%. Минард и другие применили двунаправленную сеть с долгой краткосрочной памятью (bidirectional long short term memory; BiLSTM), которая показала 47.8% F-меры [68]. Команда из университета Цюриха применила полносвязную нейронную сеть, которая получила 44.5% F-меры [76]. В соревнованиях SMM4H 2019-го года участники также преимущественно применяли нейронные сети, при этом классические подходы, основанные на моделях SVM или CRF, использовались в качестве базовых моделей для сравнения [73]. В отличие от предыдущего года, в качестве входов нейронных сетей преимущественно применялись векторные представления слов, предварительно обученных с помощью модели BERT [77], а не классической word2vec. В задаче извлечения упоминаний побочных эффектов на уровне сущностей первое место заняла система, применившая комбинацию модели BioBERT и CRF [19]. Разработанная модель достигла 65.8% F-меры при нестрогом совпадении извлеченной и эталонной сущностей и 46.4% F-меры при строгом совпадении сущностей.

1.3 Извлечение отношений из текстов биомедицинской тематики

Различные подходы были предложены для задачи извлечения отношений между сущностями в текстах биомедицинской тематики [78—84]. Ввиду высокой сложности биомедицинских текстов в большинстве работ поиск отношений осуществляется в рамках одного предложения. Подходы, используемые в системах извлечения отношений, варьируются от методов на основе простого лингвистического анализа до построения систем, основанных на правилах и машинном обучении. Основываясь на используемых подходах, методы извлечения отношений можно классифицировать на пять групп: поиск совместной встречаемости, основанные на шаблонах, основанные на правилах, основанные на классических моделях машинного обучения и основанные на нейронных сетях. Далее представлены наиболее общие характеристики этих методов.

1.3.1 Методы на основе совместной встречаемости сущностей

Поиск совместной встречаемости наиболее простой метод для поиска отношений между сущностями в рамках одного предложения, аннотации или документа [85]. Метод основан на гипотезе, что если две сущности довольно часто встречаются вместе, то вероятнее всего между ними существует связь. Такие подходы, как правило, показывают высокую полноту, однако, сильно проигрывают другим подходам по показателям точности, поскольку предложения в биомедицинских текстах длинные, имеют сложную структуру и включают в себя сразу несколько сущностей, при этом лишь малая часть из них взаимосвязана между собой. Например, в корпусе AIMED [86] лишь 17% пар белков из одного предложения находятся в отношении белок-белок. Помимо этого, еще одним недостатком данного подхода является отсутствие возможности определить тип отношения. Преимуществом данного подхода является простота его реализации, отсутствие необходимости лингвистического анализа и размеченной выборки. В связи с этим данные подходы используются в качестве базового метода для оценки эффективности разработанного метода на новом корпусе данных [87—89].

1.3.2 Методы на основе шаблонов и правил

Подход, основанный на шаблонах, базируется на поиске соответствия лингвистическим шаблонам, представленным в виде регулярных выражений. Шаблоны могут быть выявлены вручную или автоматически. Пример шаблона может выглядеть следующим образом: Б1 глагол Б2, где Б1, Б2 - это названия белков, а название глагола выбирается из заранее определенного списка глаголов, идентифицирующих связь между белками, например: подавляет, активирует, связывается. Системы, основанные на выявлении шаблонов вручную, могут выявить ограниченное число отношений, поэтому результаты таких систем невысоки [90]. Более поздние системы стали строить шаблоны с использованием частей речи и синтаксического анализа предложений, однако такой подход плохо масштабируем на более сложные предложения [91]. Помимо этого, более глубокий анализ предложений порождает большое количество шаблонов, что снижает качество работы системы. Кроме того, выявленные вручную шаблоны плохо масштабируемы на другие корпуса. Оценка качества подобных систем показывает, что они имеют высокую точность, однако, низкую полноту [92]. Единственным преимуществом данных систем является отсутствие необходимости иметь заранее размеченную выборку, однако, сам метод трудоемкий и затратный по времени, поэтому на практике он почти не применяется.

Автоматическое извлечение шаблонов позволяет минимизировать время разработки модели. Существует два подхода к автоматическому извлечению шаблонов: используя заранее известные связанные пары сущностей или извлеченные непосредственно из корпуса. При первом подходе сначала выделяются шаблоны, позволяющие извлечь отношения между заранее известными парами сущностей. На основе полученных шаблонов из текста извлекаются новые пары сущностей. Далее для новых пар сущностей повторяется первый шаг и так до тех пор, пока не удастся извлечь новые шаблоны [93—95]. Необходимость маленького набора исходных отношений является основным преимуществом описанного подхода. Однако при таком способе получения шаблонов генерируется множество шаблонов, приносящих шум в выходные данные. В противоположность этому методу, подход, основанный на извлечении шаблонов из размеченного корпуса, требует наличие большего количества начальных отношений. Не смотря на то, что автоматическое извлечение шаблонов может

увеличить полноту, при этом снижается точность из-за большого количества шаблонов, вносящих шум в выходные данные [96—99]. Однако данные системы более масштабируемы на новые корпуса данных.

Подход, основанный на правилах, использует наборы правил для извлечения отношений между сущностями [100; 101]. Аналогично подходу на основе шаблонов, правила извлекаются вручную или автоматически. Некоторые правила дополняют шаблонные подходы, добавляя ограничения, которые позволяют выявить отрицания и направление отношений [96; 102; 103]. В других системах правила выражаются набором некоторых процедур или эвристических алгоритмов [104—106]. В дальнейшем эти правила применяются к синтаксической структуре предложения, в частности, к синтаксическому дереву, для извлечения отношений.

1.3.3 Методы на основе машинного обучения

Благодаря появлению все большего количества размеченных корпусов биомедицинских текстов методы машинного обучения стали наиболее распространенными для задачи извлечения отношений. Большинство подходов основаны на обучении с учителем, при этом задача поиска отношений рассматривается как задача бинарной классификации. Несмотря на большое разнообразие моделей машинного обучения, наибольшее распространение получил метод SVM [37; 107—109]. Методы на основе машинного обучения для задачи извлечения отношений можно разделить на две группы: методы на основе ядра и методы на основе признаков.

Методы на основе ядра используют в качестве представления входных данных синтаксические деревья и графы зависимостей и на основе них оценивают сходство между двумя экземплярами входных данных [79; 110—115]. В качестве ядер используются следующие представления: мешок слов, поверхностное лингвистическое представление, поддерево синтаксического разбора, граф. Ниже представлено подробное описание каждого из методов.

– Ядро мешка слов использует вектор неупорядоченных слов входа и далее подсчитывается схожесть полученных векторов [86].

- Поверхностное лингвистическое ядро состоит из двух частей: глобальное ядро и ядро локального контекста. Глобальное ядро использует три вектора в виде мешка слов: слова до, после и между сущностями и подсчитывает общие слова в двух рассматриваемых предложениях. Ядро локального контекста использует в качестве признаков заданное количество слов слева и справа от сущностей [116].
- Ядро на основе поддерева синтаксического разбора использует синтаксическое дерево разбора и подсчитывает количество общих поддеревьев в двух предложениях [112; 114].
- Графовое ядро вычисляет сходство между двумя входными графами, подсчитывая вес общих путей в графах. В качестве графов используются синтаксические графы зависимостей и отношений в предложении [82].

Поскольку каждое ядро направлено на рассмотрение ограниченного числа характеристик предложения, в моделях используются комбинации различных ядер [117]. Такой подход улучшает качество поиска отношений, однако требует больших вычислительных ресурсов. Лиу и другие разработали систему AZDrugMiner для сбора информации о побочных эффектах с форумов пациентов [37]. Система состоит из нескольких модулей: сбор текстовых данных; классификация текста с целью отфильтровать релевантные документы; выделение именованных сущностей, обозначающих побочные эффекты и названия лекарственных препаратов; извлечение отношений между побочными эффектами и лекарствами. Модуль извлечения отношений состоит из трех компонентов: генерация признаков, функция ядра, классификатор. в качестве признаков использовалось кратчайшее расстояние между сущностями по дереву синтаксического разбора предложения с указанием части речи для каждого токена пути. Функция ядра основывалась на подсчете общих признаков у пары отношений. В качестве классификатора использовалась модель SVM. Проведенные эксперименты показали, что результаты работы модели имеют высокую корреляцию с данными из системы FAERS.

В методах, основанных на признаках, каждый вход представляется в виде вектора признаков. Такой подход направлен на извлечении наиболее информативных признаков, способствующих улучшению качества классификации. Различные признаки были предложены для моделей классификации [78; 118—123], наиболее часто используемые среди них:

- мешок слов: вектор признаков, состоящий из слов до, после и между сущностями;
- часть речи: вектор признаков, состоящий из частей речи слов до, после и между сущностями;
- кратчайший путь: закодированный кратчайший путь от одной сущности к другой по дереву синтаксического разбора;
- графовые признаки: состоят из меток и весов графов зависимостей между сущностями в предложении;
- расстояние между сущностями: количество слов между сущностями, количество слов-индикаторов между сущностями, например, специфичных глаголов, которые указывают на существование связи.

В целях улучшения качества в классификаторах используются комбинации перечисленных признаков.

В 2014 году было организовано соревнование BioCreative V [124]. Одной из задач соревнования было извлечение отношений между химическими веществами и заболеваниями из текстов аннотаций научных статей с ресурса PubMed. Первое место в соревновании получила система, состоящая из двух SVM-классификаторов, один из которых обучался на уровне предложения, другой - на уровне документов [107]. Оба классификатора получали на вход признаки, основанные на знаниях: совстречаемость сущностей в базах данных CTD, MEDI и SIDER и путь по дереву тезауруса MeSH[8] между сущностями. Кроме того, в качестве признаков была извлечена статистика упоминаний сущностей во всем документе корпуса соревнования и заголовке. Дополнительно для классификатора, обучающегося на уровне предложения, на вход подавались слова из контекста с номером позиции. Описанная система показала 57.03% F-меры. Понс и другие разработали классификатор на основе SVM с признаками на основе знаний и синтаксического дерева разбора и статистическими [108]. Классификатор показал 52.6% F-меры и занял второе место в соревновании. Ле и др. предобработали исходные тексты модулем разрешения кореференции и применили SVM классификатор с признаками на основе токенов, частей речи и кратчайшим путем по дереву синтаксического разбора между сущностями [109]. Модель показала 51.6% F-меры. Результаты F-меры остальных участников варьируются от 32.01% до 51.0%.

[8] https://www.nlm.nih.gov/mesh/meshhome.html

1.3.4 Методы на основе нейронных сетей

Современные подходы по извлечению отношений преимущественно основаны на применении нейронных сетей. Наибольшее распространение для данной задачи получила CNN [125—127]. Зенг и др. представили модель CNN, которая извлекает отношения на основе удаленного обучения (англ. distant supervision) [125]. В данной модели CNN создает семантическое представление предложения. Модель достигла 78,3% точности на наборе данных Freebase и корпусе New York Times. Однако метод учитывает только те пары сущностей, которые упоминаются в одном предложении и отбрасывает большое количество возможных отношений среди сущностей, находящихся в разных предложениях. Янкай и др. модифицировали описанную модель, добавив механизм выборочного внимания для слов в предложении [126]. Модель достигла 72.2% точности.

Несколько моделей были протестированы на корпусе данных с соревнования SemEval-2010. Зенг и др. разработали CNN, которая принимает на вход векторные представления слов и производит лексические признаки на уровне предложения автоматически [127]. Данная архитектура извлекает отношения без сложной предварительной обработки текста. Модель показала 82,7% F-меры. Жанг и др. применили BiLSTM с дополнительными признаками, полученными из лексических ресурсов и систем по обработке текстов, включая: тезаурус WordNet, синтаксический парсер и систему по извлечению именованных сущностей [128]. Модель показала 84,3% F-меры. Жоу и др. дополнили BiLSTM слоем с механизмом внимания [128]. Модель принимает на вход необработанные тексты без дополнительных признаков и при этом показывает результаты на том же уровне, что у BiLSTM (84% F-меры). Жи-Ху и др. исследовали проблему пристрелочной классификации отношений [129]. Авторы предложили многоуровневую составляющую и агрегирующую сеть, которая интерактивно кодирует экземпляры входных данных. Модель получила самые высокие результаты 92.66% точности на наборе данных FewRel, который состоит из 70,000 предложений для 100 отношений, полученных из Википедии.

1.3.5 Извлечение отношений из текстов электронных медицинских карт

Первые работы по извлечению отношений из текстов электронных медицинских карт (ЭМК) пациентов появились в 2008-м году. Робертс и др. предложили метод на основе машинного обучения для извлечения отношений из историй болезней онкобольных пациентов [130]. Метод основан на SVM с лексическими и синтаксическими признаками, полученными для каждой пары сущностей. Система показала 70% F-меры. Данная система была встроена в платформу clinical E-Science (CLEF), целью которой является извлечение важной информации из ЭМК. Система также включает в себя автоматическое распознавание сущностей. CLEF была использована для извлечения 6 миллионов отношений из более полумиллиона документов пациентов.

Одна из задач соревнования i2b2 2010 была посвящена определению типов отношений между медицинскими проблемами, тестами и методами лечения в клинических медицинских записях [131]. В рамках задачи было необходимо классифицировать отношения между парами заданных сущностей из одного предложения. Модель, основанная на семантических признаках из тезисов Medline и синтаксического дерева разбора предложения, показала наивысшие результаты среди других участников - 73.7% F-меры [132]. Модель, разработанная командой NRC Канада, достигла 73.1% F-меры [133]. Модель основана на методе максимальной энтропии со следующим набором признаков: основанным на дереве разбора; поточечная взаимная информация между двумя сущностями, подсчитанная на аннотациях Medline; словоформа; концепты и признаки на уровне контекста, параграфа и документа. Кроме того, авторы применили балансировку категорий и частичное обучение. Система, получившая третье место, адаптировала гибридный подход, который комбинирует машинное обучение и метод на основе шаблонов [134]. Авторы обучили SVM с третями типами признаков: словоформенные, лексические и синтаксические. Система показала 70.9% F-меры. Остальные участники применили методы машинного обучения с учителем, которые показали результаты F-меры, варьирующиеся от 70.2% до 65.6% [135—139]. Одна из основных проблем, с которой столкнулись участники, - это несбалансированное количество примеров для каждого типа отношений. Разработанные участниками классификаторы могли классифицировать с высо-

кой точностью классы с большим количеством примеров, используя признаки на основе текстов. Однако для распознавания остальных типов необходима была разработка дополнительных лингвистических правил.

Дальнейшие исследования на корпусе соревнования i2b2 были посвящены разработке гибридных систем, основанных на комбинации правил и методах машинного обучения. Д'Соуза и Нг применили комбинацию метода на основе правил и машинного обучения с большим набором признаков [140]. Этот подход уменьшил количество ошибок на 17-24% по сравнению с предыдущим лучшим результатом на корпусе соревнования i2b2. Саху и др. исследовали возможности CNN для извлечения отношений на корпусе i2b2 [141]. Модель получает на вход целое предложение и генерирует признаки для каждого слова. Полученные вектора проходят через сверточный, полносвязный и $softmax$ слои. Результаты показывают, что CNN может извлекать глобальные признаки, которые учитывают контекст, что способствует улучшению результатов. Лв и др. использовали условные случайные поля и применили модель глубокого обучения автокодировщик для оптимизации признаков [142].

В 2018-м году было организовано соревнование по извлечению информации из ЭМК MADE [142]. Задачей соревнования было извлечение информации о побочных эффектах и отношений между лекарствами, их атрибутами и заболеваниями. В отличие от соревнования i2b2 в тексте были выделены только сущности. Таким образом, участникам необходимо было сначала определить пары сущностей-кандидатов и затем определить, есть ли связь между ними. Система, выигравшая соревнование, применила алгоритм случайного леса со следующими признаками: (i) тип сущностей-кандидатов и словоформа, (ii) количество сущностей между ними и их типы, (iii) токены и части речи слов между сущностями-кандидатами и рядом с ними [143]. Согласно таблице результатов соревнования описанная система показала 86.8% усредненной микро F-меры. Дандала и др. использовали BiLSTM с с механизмом внимания и заняли второе место с усредненной F-мерой 84% [144]. Команда, занявшая третье место, применила SVM модель [145]. Классификатор принимал на вход четыре типа признаков: позиция, расстояние, мешок слов и мешок сущностей, и показал 83.1% усредненной микро F-меры. Маге и др. разработали классификатор на основе алгоритма случайного леса с типами сущностей, количеством слов в сущностях, количеством слов между сущностями, усредненным векторным представлением каждой из сущностей и индикатором присутствия сущностей

в одном предложении [146]. Этот подход показал 81.6% усредненной микро F-меры. Таким образом, большинство участников применяли классические модели машинного обучения и только одна команда использовала нейронную сеть, при этом результаты на одном уровне.

Мункхдалай и др. провели дополнительные эксперименты на корпусе MADE и исследовали три модели машинного обучения для извлечения отношений: (i) SVM, (ii) сквозная глубокая нейронная сеть, (iii) базовая система на основе правил с обучением [147]. Авторы использовали следующие признаки для классификатора SVM: типы сущностей, количество сущностей, токены между сущностями, n-граммы между двумя сущностями и соседними токенами, символьные n-граммы именованных сущностей. В качестве нейронных сетей применялась сеть BiLSTM с вниманием. Лучшие результаты показал классификатор SVM - 89.1% усредненной микро F-меры, в то время как нейронная сеть показала только 65.72% F-меры.

Шелманов и др. разработали корпус аннотированных клинических текстов на русском языке, в котором размечены отношения между заболеванием, частью тела, степенью тяжести и курсом лечения [148]. В статье также сравнивались четыре метода на основе машинного обучения для извлечения отношений: SVM с линейным ядром, SVM с радиальной базисной функцией в качестве ядра, RF, AdaBoost, основанный на деревьях решений. Проведенные эксперименты показали, что для типа отношений степень тяжести наилучшие результаты F-меры показал метод RF (87.5%). Для типа курс самого высокого результата F-меры достиг линейный SVM (95.7%). Для типа часть тела наиболее высокие результаты показал SVM с радиальной базисной функцией в качестве ядра (83.3%).

1.4 Выводы к первой главе

В настоящий момент большинство исследований по классификации текстов и извлечению отношений между сущностями сводятся к использованию методов машинного обучения, где требуется сформировать вектор признаков, или к применению нейронных сетей. Среди недостатков рассмотренных моделей для задачи извлечения информации из биомедицинских текстов можно

выделить три основных. Во-первых, методы разработаны и протестированы для текстов определенного домена: социальные медиа, ЭМК или научные тексты статей. Различие лексиконов и синтаксических конструкций предложений в текстах с разных ресурсов может существенно повлиять на качество результатов модели. Во-вторых, существующие модели тестируются в рамках одного корпуса и практически отсутствуют работы, в которых обучение и оценка моделей проводилась бы на разных корпусах. В-третьих, существующие методы, в основном, разработаны для английского языка, при этом для русского языка исследования в данной области практически не проводились.

Глава 2. Метод классификации сущностей

В данной главе будет дана формальная постановка задачи классификации сущностей, описание разработанных методов для решения поставленной задачи, полученные результаты и выводы. Для решения поставленной задачи была разработана модель LSTM+CA+feat, основанная на нейронной сети, принимающей на вход векторное представление слов и набор извлеченных из текста признаков. Архитектура нейронной сети LSTM+CA+feat состоит из слоя LSTM и механизма кросс-внимания. Исходные тексты сущностей и контекстов кодируются в виде векторного представления слов и подаются на вход LSTM слоям. Далее вычисляется среднее значение векторов скрытых состояний сущности и контекста. На основе полученных значений вычисляется два вектора внимания: среднего значения сущности относительно выхода слоя LSTM для контекста и наоборот среднего значения контекста относительно выхода слоя LSTM для сущности. Полученные вектора внимания конкатенируются с извлеченными признаками и передаются на вход полносвязному слою с функцией активации softmax для дальнейшей классификации. Оценка разработанной модели проводилась на корпусах, состоящих из текстов аннотаций научных статей по теме медицины, текстов электронных карточек пациентов, твитов по теме здоровья и отзывов пользователей о лекарственных препаратах. На первом этапе модель сравнивалась с несколькими базовыми методами, при этом оценка проводилась в рамках одного корпуса. Далее проводилась кросс-доменная оценка разработанной модели для англоязычных корпусов и сравнение с лучшей базовой моделью, выявленной на первом этапе.

2.1 Формальная постановка задачи

Каждая текстовая коллекция состоит из набора текстовых документов: $D = \{d_1, d_2, \ldots, d_n\}$, где $d_i = \{s_{i1}, s_{i2}, \ldots, s_{|di|}\}$, s_{ij} - предложение документа. Каждое предложение состоит из последовательностей токенов: w_{ij}^k: $s_{ij} = \{w_{ij}^1, w_{ij}^2, \ldots, w_{ij}^n\}$, включающих в себя слова, числа, знаки препинания. Формально, обозначим как $e_{ij} = \{w_{ij}^u, \ldots, w_{ij}^v\}$, $u, v \in [1, k]$ - сущность, состоящую

из одного или более токенов в рамках одного предложения. Постановка задачи звучит следующим образом: необходимо определить класс сущности $a(e_{ij})$ для всех сущностей из документов контрольной выборки $d_{ij} \in [1, ..., |d_i|], i \in [1, ..., |D|]$. В данной работе рассматривается бинарная классификация, то есть $a \in [0, 1]$, где $a(e_{ij}) = 1$ (ADR), если e_{ij} обозначает побочный эффект и $a(e_{ij}) = 0$ (non-ADR) в противном случае.

Побочный эффект (side effect), согласно определению Всемирной организации здравоохранения (ВОЗ), – любой непреднамеренный эффект фармацевтического продукта (лекарственного средства), развивающийся при использовании его у человека в обычных дозах и обусловленный его фармакологическими свойствами [149]. Например, в предложении "это лекарство очень хорошее, боли почти прошли", сущность "боль" - показание к приему лекарства, соответственно, a("боль") $= 0$. При этом в предложении "Не рекомендую лекарство, если вы не готовы справиться с болью, которое оно вызывает", сущность "болью" обозначает побочный эффект (a("болью") $= 0$). Примеры высказываний с $a(e_{ij}) = 1$ (сущности выделены курсивом):

- Несколько лет назад попробовала этот препарат, был жуткий *стресс*, *недосып*.
- Через 2 дня появилось *покраснение* и *отечность*, была небольшая *болезненность*.
- Но эти таблетки надо пить с осторожностью, потому что вызывают *привыкание*.
- Но после приема таблетки, начинала стремительно *подниматься температура*...
- Эффекта ноль, только *клонит в сон*.

2.2 Модель LSTM+CA+feat для задачи классификации сущностей

Для решения поставленной задачи была разработана модель LSTM+CA+feat, основанная на нейронной сети с кросс-вниманием LSTM+CA (Cross-attention LSTM), принимающей на вход векторное представление слов и набор извлеченных из текста признаков. Механизм кросс-внимания показал свою эффективность для различного рода задач: анализа тональности [150],

рекомендация заголовков [151], понимание языка [152], генерации текста к картинке [153].

2.2.1 Архитектура нейронной сети LSTM+CA

Архитектура нейронной сети LSTM+CA состоит из слоя с короткой долгосрочной памятью (англ. long short-term memory; LSTM) и механизма кросс-внимания. Общая архитектура сети представлена на рисунке 2.1. Исходные тексты сущностей и контекстов кодируются в виде векторного представления слов $[w_c^1, w_c^2, ..., w_c^n]$ и подаются на вход LSTM слоям.

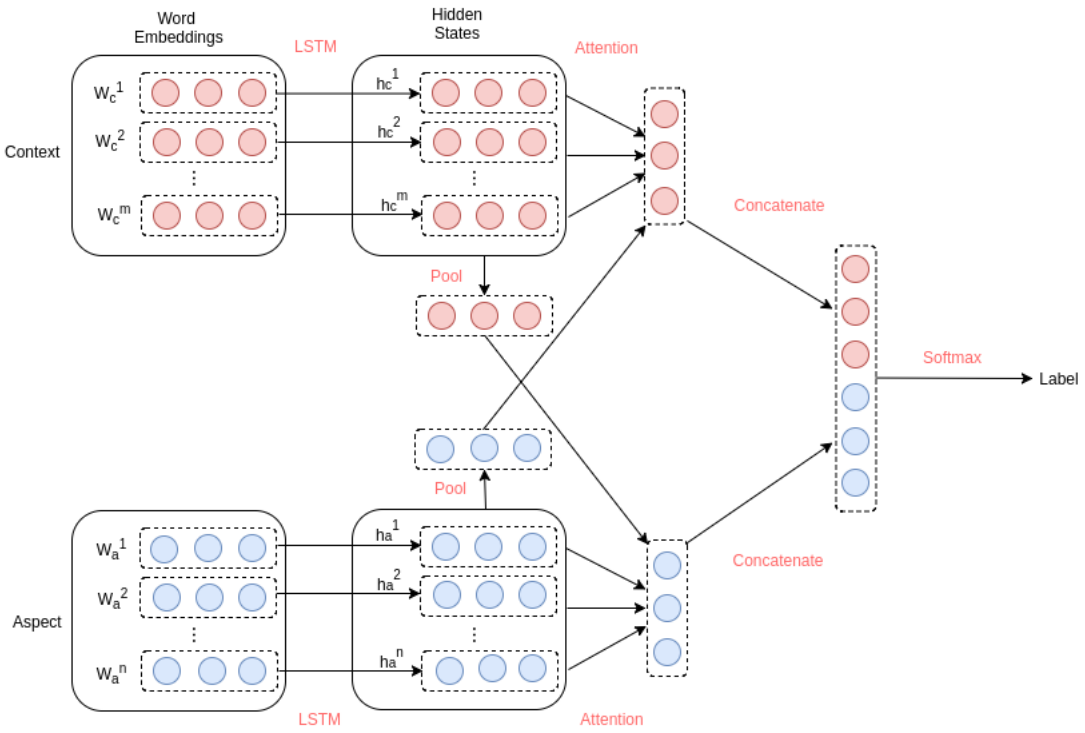


Рисунок 2.1 — Общая архитектура модели LSTM+CA для классификации сущностей

Слои с короткой долгосрочной памятью - особая разновидность рекуррентных слоев, способные к обучению долговременным зависимостям. Поскольку слова в предложении сильно зависят друг от друга способность запоминать предыдущие состояния (слова) позволяет LSTM слоям идентифицировать эти зависимости и благодаря этому лучше интерпретировать семантику текста. Формально, для данного входного слова w_k, предыдущего состояния ячейки

c_{k-1} и предыдущего скрытого состояния h_{k-1} текущее состояние сети c_k и скрытого слоя h_k вычисляется по следующим формулам:

$$i_k = \sigma(W_i^w \cdot w^k + W_i^h \cdot h^{k-1} + b_i)$$
$$f_k = \sigma(W_f^w \cdot w^k + W_f^h \cdot h^{k-1} + b_f)$$
$$o_k = \sigma(W_o^w \cdot w^k + W_o^h \cdot h^{k-1} + b_o)$$
$$\hat{c}^k = tanh(W_c^w \cdot w^k + W_c^h \cdot h^{k-1} + b_c)$$
$$c^k = f^k \odot c^{k-1} + i^k \odot \hat{c}^k$$
$$h^k = o^k \odot tanh(c^k)$$

,

где i - входной вектор f - вектор забывания, хранящий вес старой информации o - выходной вектор, которые обеспечивают взаимодействие между ячейками памяти и их окружением. σ обозначает логистическую функцию (сигмоиду). W и b обозначают веса матрицы и смещений слоя. Символ $\cdot$ обозначает стандартное умножение матриц, символ $\odot$ - поэлементное умножение матриц. На выходе LSTM слой выдает скрытые состояния $[h_c^1, h_c^2, ..., h_c^n]$ и $[h_e^1, h_e^2, ..., h_e^n]$ для контекста и сущности.

Далее вычисляется среднее значение векторов скрытых состояний сущности и контекста:

$$c_{avg} = \sum_{i=1}^{n} \frac{h_c^i}{n}$$
$$e_{avg} = \sum_{i=1}^{n} \frac{h_e^i}{n}$$

На основе полученных значений вычисляется два вектора кросс-внимания: среднего значения сущности относительно выхода слоя LSTM для контекста и наоборот среднего значения контекста относительно выхода слоя LSTM для сущности. Для вектора контекста $[h_c^1, h_c^2, ..., h_c^n]$, полученного на предыдущем шаге, генерируется вектор внимания α с использованием среднего значения вектора целевой сущности e_{avg}:

$$\alpha_i = \frac{exp(\gamma(h_c^i, e_{avg}))}{\sum_{j=1}^{n} exp(\gamma(h_c^j, e_{avg}))}$$

где γ - функция оценки, которая показывает степень важности h_c^i в контексте и вычисляется по формуле:

$$\gamma(h_c^i, e_{avg}) = tanh(h_c^i \cdot W_c \cdot e_{avg}^T + b_c)$$

где W_a и b_a - матрица веса слоя и матрица смещения соответственно. $tanh$ - обозначает нелинейную функцию гиперболического тангенса. e_{avg}^T - транспонированная матрица e_{avg}.

Аналогично вычисляется вектор внимания β для сущности:

$$\beta_i = \frac{exp(\gamma(h_e^i, c_{avg}))}{\sum_{j=1}^{m} exp(\gamma(h_e^j, c_{avg}))}$$

$$\gamma(h_e^i, c_{avg}) = tanh(h_e^i \cdot W_a \cdot c_{avg}^T + b_a)$$

На основе полученных векторов внимания для слов вычисляется векторное представление контекста c_r и целевой сущности e_r. учитывающее внимание:

$$c_r = \sum_{i=1}^{n} \alpha_i h_c^i$$

$$e_r = \sum_{i=1}^{n} \beta_i h_e^i$$

Полученные вектора представления контекста c_r и целевой сущности конкатенируются в один вектор d для дальнейшей классификации. Далее также, как и в предыдущей модели используется нелинейный слой с логистической функцией softmax для классификации.

В процессе обучения модели необходимо оптимизировать параметры: из LSTM слоя - $[W_i^w, W_f^w, W_o^w, W_c^w, W_i^h, W_f^h, W_o^h, W_c^h, b_i, b_f, b_o, b_c]$, из слоя внимания - $[W_a, b_a]$, из слоя softmax - $[W_l, b_l]$ и вектора в первом слое векторного представления слов. Обозначим все перечисленные параметры через Θ. В качестве функции потерь используется перекрестная энтропия с L_2 регуляризацией:

$$J = -\sum_{i=1}^{C} g_i log(y_i) + \lambda_r(\sum_{\theta \in \Theta} \theta^2)$$

где $g_i \in R^C$ - верные ответы, представленные в виде бинарного вектора со значением 1 на месте правильного класса, $y_i \in R^C$ - оценочные вероятности для

каждого класса, которые выдает модель в качестве ответа, λ_r - коэффициент регуляризации для L_2.

Для вычисления градиентов используется метод обратного распространения ошибки:

$$\Theta = \Theta - \lambda_l \frac{\partial J(\Theta)}{\partial(\Theta)}$$

где λ_l - скорость обучения.

2.2.2 Выбор наиболее информативных признаков

Для выявления наиболее значимых признаков для задачи классификации сущностей был реализован классификатор на основе SVM. Исходный код модели доступен в открытом репозитории[1].

Описание признаков

В качестве входных данных для модели были сгенерированы два вида признаков: на уровне сущностей, когда признак основывается исключительно на словах, входящих в сущность, и контекстные, когда признак базируется на заданном количестве слов слева и справа от сущности.

Контекстные признаки:

1. *Мешок слов* (bow) представляет из себя бинарный вектор равный по длине количеству уникальных слов в тексте. Каждый элемент вектора идентифицирует наличие слова в рассматриваемом тексте, 1 - если слово есть в контексте, 0 - в противном случае.
2. *Вектор частей речи* (pos). Для каждого контекста было посчитано количество существительных, глаголов, наречий и прилагательных.

[1] https://github.com/Ilseyar/adr_classification

3. *Тональные признаки* (sent). В качестве признаков были подсчитаны классические числовые характеристики, применяемые в задачах анализа тональности [154]:
   - количество токенов, входящих в состав словаря и имеющую тональную окраску не равную 0
   - сумма всех значений сентиментов
   - максимальное значение сентиментов
   - значение сентимента последнего слова в контексте
   - сумма позитивных сентиментов, находящихся в утвердительной части контекста
   - сумма негативных сентиментов, находящихся в утвердительной части контекста
   - сумма позитивных сентиментов, находящихся отрицательной части контекста
   - сумма негативных сентиментов, находящихся в отрицательной части контекста

   Под отрицательной частью контекста понимаются слова, стоящие после частиц, выражающих отрицание, таких как не, нет для русского и no, not, don't для английского языка.
4. Поточечная взаимная информация (Pointwise mutual information, PMI) (pmi) применяется в анализе текстов для определения вероятности совместной встречаемости слов. В анализе тональности данная метрика позволяет определить насколько часто данное слово встречается в позитивном или негативном контексте. Значение PMI вычисляется по формулам:

$$SentimentScore(w) = PMI(w, positive) - PMI(w, negative)$$

$$PMI(w, positive) = log_2 \frac{freq(w, positive) \cdot N}{freq(w) \cdot freq(positive)}$$

$$PMI(w, negative) = log_2 \frac{freq(w, negative) \cdot N}{freq(w) \cdot freq(negative)}$$

   где freq(w, positive) - это частота встречаемости слова w в позитивном контексте, а freq(w, negative) - в негативном.
5. *Наличие лекарства или побочного эффекта* (drug adr). Этот признак представляет собой бинарный вектор длины два. Первый компонент

вектора указывает на наличие в контексте названия лекарства, второй - присутствие побочных эффектов. Названия лекарств были использованы из системы FDA[2], а побочные эффекты из словаря COSTART[3]. Данный признак применялся только для английского языка.

Признаки на уровне сущностей:

– *Векторное представление слов* (emb). В качестве признака был использован усредненный вектор всех слов сущности, полученный из предобученной модели векторного представления слов.
– *Векторное представление на основе кластеров* (cls). Каждое слово было представлено в виде бинарного вектора длины 150, где 1 обозначает принадлежность к кластеру с соответствующим номером.
– *Семантические типы из унифицированной системы медицинского языка* (umls). UMLS - это онтология, объединяющая в себе медицинские термины и иерархические связи между ними. Семантические типы обеспечивают категоризацию терминов словаря. Примерами таких категорий могут быть: лекарство, медицинское устройство, витамин. В качестве признака было использовано количество слов из контекста, входящих в различные категории.

Для определения частей речи использовались библиотеки nltk [155] и Texterra [156] для английского и русского языков соответственно. Тональные признаки основаны на наиболее распространенных тональных словарях SentiWordNet [157], MPQA Subjectivity [158], Bing Liu's [159] для английского языка и RuSentiLex [160] для русского языка. В словаре SentiWordNet для каждого слова указаны значения сентиментов - два числа от 0 до 1, показывающие на сколько слово выражает позитив или негатив. В остальных словарях указана только полярность слова или выражения: позитивный, негативный или нейтральный. В этом случае значение сентимента считалось равным 1 - для позитивной, 0 - для нейтральной и -1 - для негативной полярности.

Признаки PMI, представление на основе кластеров, векторное представление слов были обучены на корпусе Health для английского языка и на неразмеченной части корпуса RuDReC для русского языка. Корпус Health [161] состоит из 2.5 миллионов отзывов пользователей о лекарственных препаратах,

[2]https://www.fda.gov/
[3]http://www.nlm.nih.gov/research/umls/sourcereleasedocs/current/CST/

собранных с различных форумов, связанных со здоровьем: askapatient.com[4], dailystrength.org[5], drugscom.com[6], webmd.com[7] и с сайта Амазон[8]. Для русского языка были использованы отзывы о лекарственных и косметических средствах из неразмеченной части корпуса RuDReC. Данные отзывы были собраны с сайта Отзовик и ПроТаблетки, количество отзывов - 247282.

Для английского языка использовалась модель векторного представления слов, представленная в [161]. Параметры модели: длина вектора - 200, размер локального контекста - 10, количество отрицательных примеров - 5, алгоритм - непрерывный мешок слов. Для русского языка использовалась модель Ruwikiruscorpora, обученная на Национальном корпусе русского языка и текстах Википедии. Модель обучалась со следующими параметрами: длина вектора - 300, размер локального контекста - 2, частотный порог - 5, алгоритм - Skipgram.

В качестве кластеров для английского языка были использованы кластеры, вычисленные в [161] с использованием иерархического алгоритма Брауна. Для русского языка кластера были обучены аналогичным образом на неразмеченной части корпуса RuDReC, состоящей из отзывов о лекарственных препаратах и косметических средств.

Результаты оценки информативности признаков

Эксперименты проводились на корпусах CADEC [162], и RuDReC [15]. Модели оценивались на 5-ти фолдовой кросс-валидации. Результаты представлены в таблице 2. используются стандартные метрики качества для классификации текстов: P - точность (precision), R - полнота (recall), F-мера.

Согласно полученным результатам классификатор показал максимальный результат 80.3% F-меры на корпусе CADEC и 77.6% F-меры на корпусе RuDReC. Наиболее информативные признаки для англоязычного корпуса: мешок слов, части речи, тональные, вектора кластеризации и признаки на основе

[4]http://www.askapatient.com/
[5]https://www.dailystrength.org/
[6]https://www.drugs.com/
[7]http://www.webmd.com/
[8]http://jmcauley.ucsd.edu/data/amazon/

Таблица 2 — Оценка информативности признаков на корпусах CADEC и RuDReC для метода на основе SVM по усредненным метрикам: P (точность), R (полнота) и F (F-мера).

	CADEC			RuDReC		
Features	**P**	**R**	**F**	**P**	**R**	**F**
bow	.827	.740	.775	.637	.631	.632
bow, pos	.824	.743	.776	.635	.642	.636
bow, pos, sent	.823	.745	.777	.633	.639	.634
bow, pos, sent, cls	.832	.776	.788	.646	.652	.648
bow, pos, sent, cls, umls	**.844**	.773	**.803**	.684	.691	.686
bow, pos, sent, cls, umls, pmi	.839	.770	.799	.705	.719	.710
bow, pos, sent, cls, umls, pmi, emb	.822	**.777**	.797	**.772**	**.785**	**.776**

концептов из словаря UMLS. Для русскоязычного корпуса также эффективными оказались признаки PMI и векторное представление слов. Тональный признак для русского языка оказался менее эффективным и его добавление в модель привело к снижению результатов F-меры на 0.2%. Наибольший прирост на корпусе CADEC был получен при добавлении признака UMLS (+1.5%). На корпусе RuDReC добавление векторного представления слов привело к максимальному приросту результатов по метрике F-меры (+6.6%).

2.2.3 Общая архитектура модели LSTM+CA+feat

Общая архитектура модели LSTM+CA+feat, представленная на рисунке 2.2, основана на нейронной сети LSTM+CA с набором дополнительных признаков, которые конкатенируются с финальным полносвязным слоем сети. В качестве признаков были использованы наиболее информативные признаки для задачи классификации биомедицинских сущностей: мешок слов, части речи, кластеры Брауна, основанные на тональных словарях и поточечная взаимная информация. Эксперименты показали, что векторное представление слов также дает прирост при классификации сущностей, однако, поскольку векторное представление слов используется для представления сущности и контекста, бы-

ло решено не добавлять данный признак дополнительно в модель. Исходный код модели доступен в открытом репозитории[9].

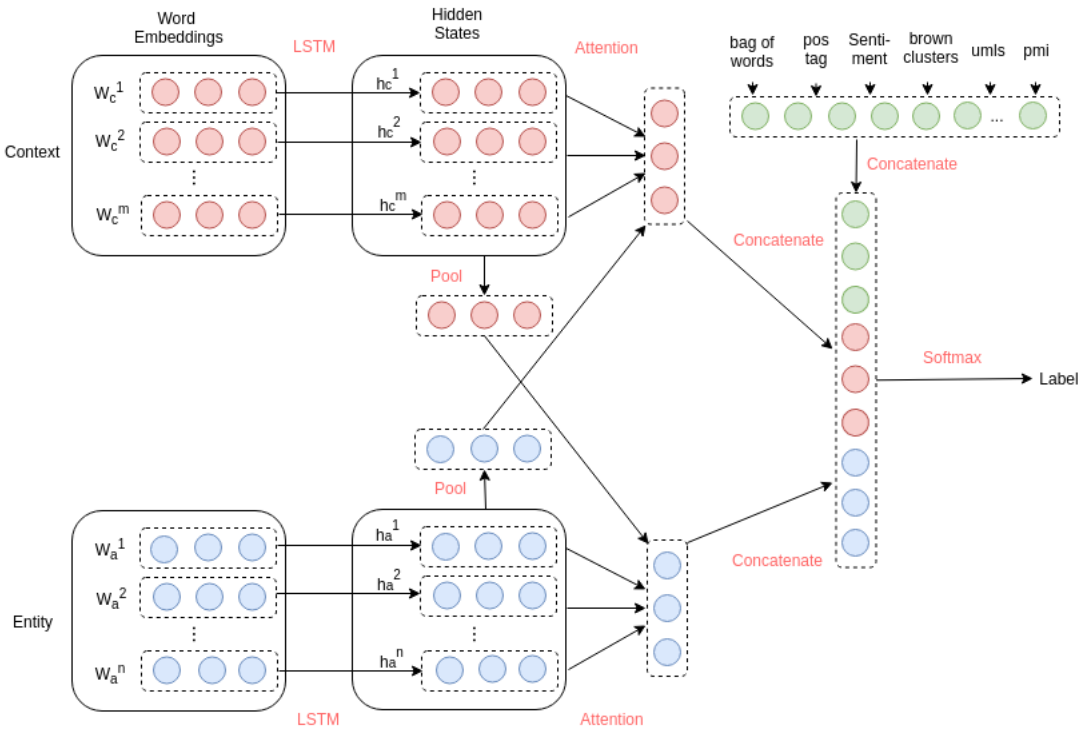


Рисунок 2.2 — Общая архитектура модели LSTM+CA+feat для классификации сущностей

2.3 Наборы данных

Эксперименты по оценке эффективности методов классификации проводились на пяти существующих англоязычных корпусах: CADEC [162], Твиттер [163], MADE [164], TwiMed [165], PsyTAR [166] и русскоязычном корпусе отзывов о лекарственных препаратах RuDReC [15].

Корпус **CADEC** состоит из размеченных отзывов пользователей о лекарственных препаратах с форума askapatient.com. В корпусе присутствуют отзывы о 12 лекарственных препаратах, разделенных на две группы: группа противовоспалительных лекарств, содержащих в своем составе Диклофенак, и лекарство для снижения холестерина Липитор. В разметке текстов участвовало 14 аннотаторов. Мера согласованности аннотаторов при строгом совпадении

[9]https://bitbucket.org/Ilseyar/entity_classification/src/master/

сущностей для отзывов о Лекарстве Диклофенак - 46.6%, для отзывов о Липиторе - 74.2%. В корпусе размечены 5 видов аннотаций: лекарство (drug), побочный эффект (adverse), заболевание (disease), симптом (symptom) и другие медицинские термины, не вошедшие в описанные категории (finding). Аннотацией лекарство отмечены все названия лекарственных препаратов в тексте. Все побочные эффекты, связанные с лекарством, отмечены аннотацией побочный эффект. Аннотацией заболевание обозначены показания к применению. Симптом обозначает сопутствующие признаки болезни. Аннотации заболевание и симптом были сгруппированы вместе с аннотацией, обозначающей другие медицинские термины в одну группу. Примеры предложений с сущностями разных типов:

- ADR: Pain feels better, but the *appetite increase* worries me. (Боль проходит, но *повышение аппетита* меня беспокоит.)
- non-ADR: I can finally clean my house without *pain*. (Наконец-то я могу убираться дома без *боли*.)

Корпус **Твиттер** содержит твиты пользователей на тему здоровья. Каждый твит отмечен в зависимости от того имеется ли в нем упоминание о побочном эффекте или нет. Два аннотатора независимо друг от друга размечали твиты пользователей под контролем эксперта в области фармакологии. Метрика согласованности аннотаторов - 81%. Политика Твиттера не позволяет хранить и распространять твиты в открытом доступе. Создатели корпуса предоставляют только идентификатор пользователя и твита по которым можно загрузить исходный текст. В связи с этим часть твитов (36%) не удалось загрузить. Во время предобработки из текстов твитов были удалены все ссылки, упоминания пользователей и ретвиты. В корпусе размечено четыре вида сущностей: побочный эффект (adverse drug reaction), положительный эффект (beneficial effect), показание к применению (indication) и другие упоминания симптомов (other). Сущности типа положительный эффект, показание к применению и другие упоминания симптомов были объединены в один класс. Примеры предложений с сущностями разных типов:

- ADR: @live_imagine Anyone taking Zolpidem suffer with *headaches* weeks after starting the med? (Кто-нибудь, принимающий Золпидем, страдает от *головной боли* через несколько недель после начала приема лекарства?)

– non-ADR: Paroxetine 40mg / day, since I started I ve been able to go outside without having a *panic attack* which is nice. (Пароксетин 40 мг / день, с тех пор как я начал, я смог выходить на улицу без *панической атаки*, что приятно.)

Корпус **MADE** состоит из обезличенных записей электронных карточек пациентов, больных раком. Корпус был создан для соревнования по обработке естественного языка, в задачи которого входило извлечение медицинских терминов, побочных эффектов и отношений между ними. Корпус содержит 1089 документов. В процессе разметки участвовали несколько аннотаторов, в том числе врачи, биологи, лингвисты и кураторы биомедицинских баз данных. Каждый документ был размечен двумя аннотаторами, один из которых выполнял первоначальную разметку, второй проверял разметку и вносил необходимые правки. Соглашение между пятью аннотаторами, рассчитанное для трех документов, составило 0.424, что находится в пределах допустимого диапазона соглашения. Каждая запись аннотирована 9-ю видами сущностей: лекарство (drug), доза (dose), частота приема (frequency), продолжительность курса (route), продолжительность (duration), тяжесть заболевания (severity), побочный эффект (adr), показания к применению (indication) и другие симптомы (SSLIF). Аннотации, обозначающие симптомы (SSLIF) и показания к применению (indication) были объединены в один класс, не относящийся к побочным эффектам. Примеры предложений с сущностями разных типов:

– ADR: The patient has chemotherapy induced *peripheral neuropathy*. (Пациент имеет вызванную химиотерапией *периферическую невропатию*.)
– The patient's *neuropathy* is well controlled on Katena. (*Невропатия* пациента хорошо контролируется Катеной.)

Корпус **TwiMed** состоит из двух частей: твитов пользователей (TwiMed-Twitter) и текстов статей с ресурса PubMed (TwiMed-PubMed). Корпус содержит следующие аннотации: болезнь, симптом и лекарство. В разметке приняли участие шесть аннотаторов из различных областей: три фармаколога, два активных пользователя социальных медиа и один носитель языка. Метрика согласованности между аннотаторами и золотым стандартом составила 75%, максимальная метрика согласованности была достигнута одним из фармакологов - 87%. Если отношение между лекарством и болезнью было размечено как негативное, то болезнь отмечалась как побочный эффект. Примеры предложений с сущностями разных типов для части, состоящей из статей PubMed:

– Docetaxel - induced *photolichenoid eruption.* (Доцетаксел вызвал *фотолихеноидную сыпь.*)
– Stevens Johnson syndrome in a *bipolar* patient treated with lamotrigine. (Синдром Стивенса-Джонсона у пациента с *биполярным расстройством*, получавшего ламотриджин.)

Примеры предложений с сущностями разных типов для части, состоящей из твитов:

– tamoxifen makes me feel like i'm *going nuts*, *mood wings*, *anger*, *depression.* (тамоксифен заставляет меня чувствовать, что я *схожу с ума*, *перепады настроения*, *гнев*, *депрессия.*)
– feeling awful today. *Nerve pain* everywhere. (чувствую себя ужасно сегодня. *Нервная боль* повсюду.)

PsyTAR (Psychiatric Treatment Adverse Reactions) - первый открытый корпус отзывов пользователей о психотропных препаратах, собранный с форума askaPatient.com. Набор данных содержит 887 отзывов (6004 предложения) о четырех психотропных лекарственных препаратах из двух классов:

1. Сертралин (Zoloft) и Эсциталопрам (Lexapro) из класса селективных ингибиторов обратного захвата серотонина;
2. Венлафаксин (Effexor) и Дулоксетин (Cymbalta) из класса селективных ингибиторов обратного захвата серотонина.

Все отзывы были размечены вручную четырьмя типами сущностей:

– Побочный эффект (adverse drug reactions; ADR);
– Синдром отмены (withdrawal symptoms; WD);
– Показания к применению (drug indications; DI);
– Признак/симптом/болезнь (sign/symptoms/illness; SSI).

В процессе разметки корпуса участвовало 4 разметчика, два из которых студенты медицинских учреждений, остальные два разметчика имели опыт работы в сфере здравоохранения. Мера согласованности аннотаторов для всего корпуса составила 0.86, при этом разброс метрики составлял от 0.81 для сущностей типа WD до 0.91 для сущностей типа DI. Для задачи классификации сущностей сущности типа абстинентный синдром, показания к применению и признак/симптом/болезнь были объединены в одну группу не побочных эффектов. Примеры предложений с сущностями разных типов:

– One of the side effects is *Alopecia.* (Одним из побочных эффектов является *Алопеция.*)

Таблица 3 — Общая статистика по корпусам. ADR - количество классов с побочным эффектом, non-ADR - количество классов с отсутствием побочного эффекта

Корпус	**Источник**	**Кол-во документов**	**ADR**	**non-ADR**	**Макс. длина предложения**	**Средняя длина предложения**
Twitter	Твиттер	645	569	76	37	22
CADEC	Отзывы	1231	5770	550	236	28
MADE	ЭМК	876	1506	37077	173	21
TwiMed-Pubmed	Научные статьи	1000	264	983	150	39
TwiMed-Twitter	Твиттер	637	329	308	42	27
PsyTAR	Отзывы	891	4525	2987	264	32
RuDReC	Отзывы	400	1500	558	165	19

– Now my *anxiety* is up but my *depression* is decreasing. (Теперь мое *беспокойство* возрастает, но моя *депрессия* уменьшается.)

Корпус **RuDReC** содержит отзывы пользователей о лекарственных препаратах с русскоязычного форума otzovik.com. В данной работе использовалась первая версия корпуса. В корпусе содержится 400 отзывов, в каждом из которых отмечены три категории сущностей: название лекарства, заболевание и побочный эффект. В категорию заболевание относятся сущности, обозначающие: название заболевания, показания к применению, симптомы, позитивные динамики после (во время) приема препарата, показания к применению. Класс побочных эффектов помимо сущностей, конкретно указывающих на побочный эффект, включает в себя сущности, обозначающие случаи ухудшения состояния после пройденного курса лечения препарата, негативные динамики после какого-то периода применения и случаев, когда препарат не действует после пройденного курса.

– Мне так плохо было от них *голова просто взрывалась*, *недомогание*, *усиление головокружения* я думала умру!
– Как то слегла с *простудой*, *температурой* и *насморком* в постель.

Общая статистика для всех корпусов представлена в таблице 3, примеры выделенных сущностей различных классов вместе с контекстом представлены в таблице 2. Как видно из статистики, корпуса CADEC и MADE содержат большее кол-во аннотаций, чем остальные корпуса. Наибольшее количество документов в корпусе CADEC - 1231, наименьшее количество в корпусе RuDReC

- 400. Максимальная средняя длина предложения в корпусе TwiMed-PubMed - 39 слов, минимальная в корпусе RuDReC - 19 слов. Стоит отметить, что все корпуса, кроме TwiMed-Twitter не сбалансированы. MADE - наиболее несбалансированный корпус, всего 4% примеров из класса ADR.

2.4 Оценка эффективности разработанной модели

В данном параграфе описаны базовые подходы, использованные для сравнения, а также результаты проведенных экспериментов. Все модели были оценены на 5-фолдовой кросс валидации с помощью стандартных метрик оценки качества классификации: точность (P), полнота (R) и F-мера – среднее гармоническое между точностью и полнотой.

2.4.1 Базовые методы для сравнения

Для оценки эффективности разработанной модели классификации сущностей результаты сравнивались с существующими базовыми подходами классификации текста, показавшими наилучшие результаты в задаче классификации для биомедицинских текстов:

1. метод на основе опорных векторов с набором наиболее информативных признаков, выявленных в пункте (SVM+feat)2.2.2;
2. метод опорных векторов, представленный в работе [47];
3. сверточная нейронная сеть (CNN) [167];
4. рекуррентная сверточная нейронная сеть (RCNN) [168];
5. сверточная рекуррентная нейронная сеть (CRNN) [52];
6. сверточная сеть с вниманием (CNNA) [52];
7. сверточная нейронная сеть с двунаправленной короткой долгосрочной памятью (CNN-BiLSTM) [54];
8. модель на основе архитектуры Трансформер BioBERT [169].

Далее приводится подробное описание каждого метода.

Набор экспериментов показал, что признаки на основе униграмм, бинрамм, частей речи, тональности, векторов кластера и семантических типов из словаря UMLS являются наиболее эффективными для классификации побочных эффектов. Совокупность данных признаков подавалась на вход модели SVM+feat.

Метод на основе SVM, предложенный в работе [47] основан на линейном ядре. Признаки, использованные в данном методе, включают: 1,2,3-грамы, синсеты, тональные признаки, замена фраз, словари побочных эффектов, тематические, длина сущности в словах, наличие сравнительной и превосходной степени прилагательного и модальных глаголов. Синсеты содержат синонимы прилагательных, глаголов и наречий предложения, полученных из тезауруса WordNet. Для тональных признаков использовались следующие словари: SentiWordNet [157], MPQA Subjectivity Lexicon [158], Bing Liu's dictionary [159]. Признак с заменой фраз состоит из вектора длины 4, элементы которого показывают количество слов, принадлежащих наборам слов, обозначающим 'меньше', 'больше', 'хорошо', 'плохо' соответственно. В модели использовались следующие словари побочных эффектов: SIDER[10], Consumer Health Vocabulary[11], COSTART[12] и DIEGO LAB[13]. Признаки, основанные на словарях побочных эффектов, состоят из двух параметров: первый показывает принадлежность токенов входного текста словарю, а второй количество токенов из словаря. В экспериментах был использован публично доступный код из репозитория[14]. Метод тестировался на трех корпусах и показал следующие результаты: 53.8% на корпусе, состоящем из твитов, 67.8% на корпусе DailyStrength, 81.2% на корпусе ADE [47].

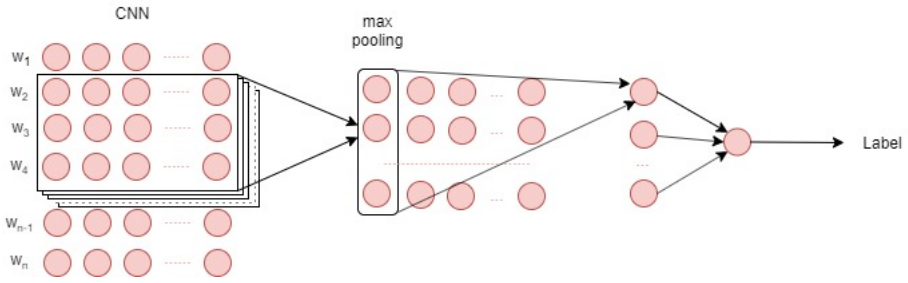


Рисунок 2.3 — Общая архитектура модели RCNN.

[10]http://sideeffects.embl.de/
[11]http://www.consumerhealthvocab.org/
[12]1 https://www.nlm.nih.gov/research/umls/sourcereleasedocs/current/CST/
[13]http://diego.asu.edu/Publications/ADRClassify.html
[14]https://bitbucket.org/asarker/adrbinaryclassifier/downloads/

Сверточная нейронная сеть (**CNN**) является одним из базовых методов классификации текстов на уровне предложения [167; 170]. Huynh и др. протестировал модель CNN для задачи классификации текстов с целью извлечения упоминаний побочных эффектов из текстов [52]. Модель показала 51% F-меры на корпусе, состоящем из твитов пользователей о здоровье [47] и 87% F-меры на корпусе ADE, состоящем из текстов с ресурса MEDLINE [171]. Сверточная нейронная сеть превзошла результаты метода максимальной энтропии и подхода на основе правил. В работе [53] CNN превзошла более комплексную архитектуру нейронной сети, состоящей из нескольких сверточных слоев, на 3.1% F-меры на корпусе ADE. Общая архитектура сети представлена на рисунке 2.3.

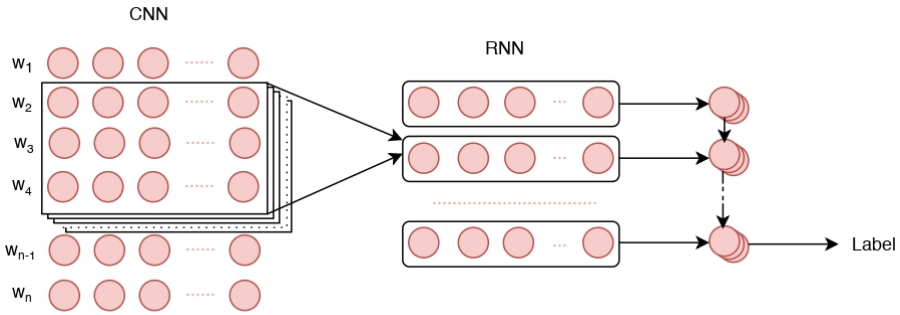


Рисунок 2.4 — Общая архитектура модели RCNN.

Рекуррентная сверточная нейронная сеть (**RCNN**) [168] также, как и CNN, принимает на вход текст, закодированный векторным представлением слов, который проходит через сверточный слой. Однако перед слоем максимального пула имеется дополнительный рекуррентный слой. В данной архитектуре в качестве рекуррентного слоя использовались управляемые рекуррентные блоки (Gated Recurrent Units; GRU). В качестве функций активации в сверточных слоях использовалась функция ReLU. Модель RCNN показала 49% F-меры на корпусе, состоящем из твитов и 83% на корпусе ADE [52]. Общая архитектура сети RCNN представлена на рисунке 2.4.

Сверточная рекуррентная нейронная сеть (**CRNN**) [52] аналогична модели RCNN за исключением того, что рекуррентный слой следует перед сверточным. В качестве рекуррентного слоя также использовался GRU. В архитектуре CRNN использовались те же параметры, что в RCNN. Модель достигла 51% на корпусе твитов и 84% на корпусе ADE [52]. Общая архитектура сети CRNN представлена на рисунке 2.5.

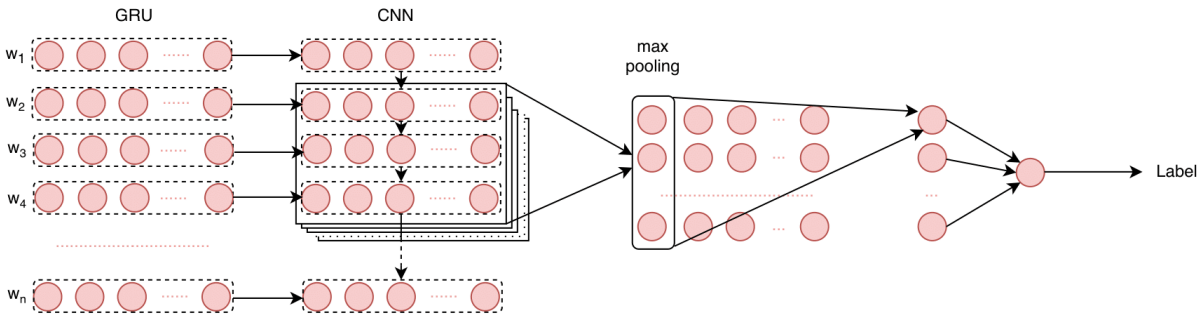


Рисунок 2.5 — Общая архитектура модели CRNN.

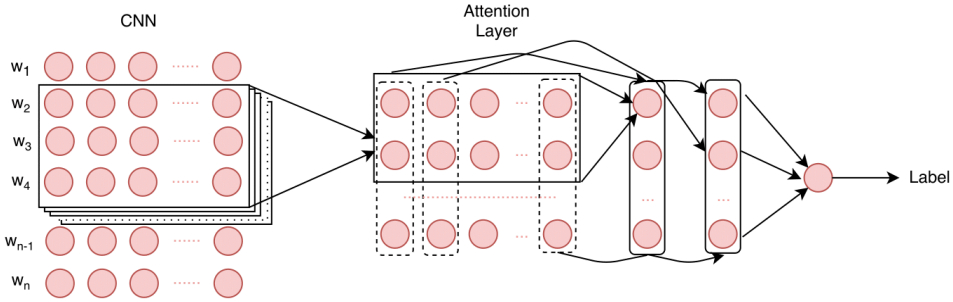


Рисунок 2.6 — Общая архитектура модели CNNA.

Сверточная сеть с вниманием (**CNNA**) разработана на основе CNN, но в дополнении имеет еще один сверточный слой, следующий за первым сверточным слоем [52]. Выходные данные второго сверточного слоя нормализуются при помощи функции softmax, полученные значения называются весами внимания. Веса внимания затем перемножаются с выходными данными первого сверточного слоя. Результат перемножения передается в выходной слой для дальнейшей классификации. Модель CNNA показала одинаковые с RCNN результаты: 49% F-меры на корпусе, состоящем из твитов и 83% на корпусе ADE [52]. Общая архитектура сети CNNA представлена на рисунке 2.6.

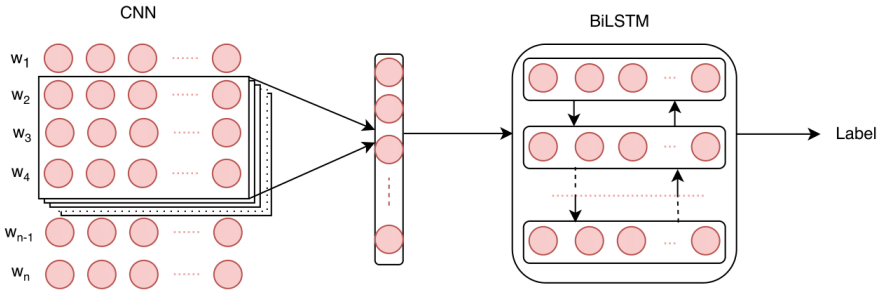


Рисунок 2.7 — Общая архитектура модели CNN-BiLSTM.

Жанг и др. представили модель (**CNN-BiLSTM**), которая является вариацией модели RCNN, однако в качестве рекуррентного слоя в архитектуре CNN-LSTM используется двунаправленная LSTM (Bidirectional LSTM; BiLSTM) [54]. Модель CNN-LSTM была протестирована для задачи классификации с целью извлечения побочных эффектов на корпусе отзывов о лекарственных препаратах с сайта askapatient.com [54]. Результаты экспериментов показали, что модель с результатом 85.57% F-меры превосходит базовые архитектуры нейронных сетей. Общая архитектура сети CNN-BiLSTM представлена на рисунке 2.7.

BERT [77] - это контекстно-зависимая модель представления слов, основанная на двунаправленной многослойной архитектуре нейронной сети - Трансформер [172]. Модель BERT предобучалась на большом корпусе статей Википедии и текстах книг. Для корпусов на английском языке применялась модель BioBERT, для корпуса на русском применялась модель RuBERT. Модель **BioBERT** (Bidirectional Encoder Representations from Transformers for Biomedical Text Mining) обучалась дополнительно на текстах аннотаций научных статей с ресурсов PubMed и PMC [169]. BioBERT значительно превосходит предыдущие современные модели в разнообразных задачах по анализа биомедицинского текста: распознавание биомедицинских именованных сущностей (улучшение метрики F1 на 0,62%), извлечение биомедицинских отношений (улучшение метрики F1 на 2,80%) и биомедицинские вопросно-ответные системы (улучшение метрики среднеобратного ранга на 12,24%) [169]. BioBERT достигла 43.17% F-меры в задаче классификации текстов твитов о здоровье [173]. Модель RuBERT (BERT-base, Multilingual Cased) была инициализирована весами мультиязычной модели MultiBERT, предварительно обучена на русскоязычных текстах Википедии и имеет 12 слоев множественного внимания, и в общей сложности содержит 180 миллионов параметров [174].

2.4.2 Параметры моделей

Входной текст кодировался векторным представлением использовалась модель векторного представления слов, обученная на записях из социальных медиа [161]. Векторное представление слов было получено с использованием

модели word2vec, обученной на неразмеченном корпусе, состоящем из 2.5 миллиона англоязычных отзывов пользователей о лекарственных препаратах. Длина векторов 200. Статистика покрываемости корпусов словами из модели векторного представления слов: CADEC - 93.5%, Twitter - 80.4%, MADE - 62.5%, TwiMed-Twitter - 81.2%, TwiMed-PubMed - 76.4%, RuDReC - 95%. Для слов, отсутствующих в модели, генерируется вектор случайных чисел с нормальным распределением и значениями, ранжирующимися в рамках значений векторов модели векторного представления слов. Все базовые модели обучались на 30 эпохах, модель LSTM+CA - на 15 эпохах, модель LSTM+CA+feat - на 10 эпохах.

2.4.3 Результаты оценки в рамках одного корпуса

Таблица 4 — Результаты классификации на корпусе Twitter.

	non-ADR			**ADR**			**Макро**		
Модель	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
SVM [Sarker et. al.]	.209	.093	.127	.887	.954	.919	.548	.523	.523
SVM+feat	.602	.520	.554	.602	.520	.554	.769	.736	.749
CNN	.222	.107	.143	.891	.975	.931	.557	.541	.537
RCNN	.334	.160	.213	.895	.959	.926	.615	.559	.569
CRNN	.380	.133	.190	.893	.968	.929	.636	.551	.559
CNNA	.201	.107	.139	.890	.964	.925	.545	.535	.532
CNN-BiLSTM	.367	.107	.143	.889	.961	.923	.628	.534	.533
BioBERT	.378	.187	.250	.898	.959	.927	.638	.573	.589
LSTM+CA	.654	.627	.634	.951	.957	.954	.802	.792	.794
LSTM+CA+feat	**.736**	**.653**	**.687**	**.954**	**.964**	**.959**	**.845**	**.809**	**.823**

Результаты экспериментов представлены в таблицах 4-10. Разработанная модель LSTM+CA+feat превзошла результаты остальных моделей на всех корпусах. Среди базовых подходов на англоязычных корпусах наилучшие результаты на корпусах CADEC и Twitter достигла модель SVM+feat, а на остальных корпусах - модель BioBERT. На корпусе RuDReC лучшие результаты средней F-меры среди базовых методов были получены моделью CNN-BiLSTM (78.4%). Максимальный прирост F-меры модели LSTM+CA+feat по сравнению с BioBERT достигнут на корпусе Twitter (+23.4%). Данный результат

Таблица 5 — Результаты классификации на корпусе CADEC.

	non-ADR			ADR			Макро		
Модель	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
SVM [Sarker et. al.]	.452	.227	.297	.914	.969	.941	.683	.598	.619
SVM+feat	.659	.620	.638	.964	.969	.967	.811	.795	.802
CNN	.557	.450	.488	.948	.964	.956	.753	.707	.722
RCNN	.596	.462	.505	.95	.965	.957	.773	.713	.731
CRNN	.666	.444	.516	.949	.976	.962	.808	.710	.739
CNNA	.624	.410	.486	.945	.976	.960	.785	.693	.723
CNN-BiLSTM	.626	.533	.561	.956	.966	.961	.791	.750	.761
BioBERT	.701	.422	.527	.947	.983	.965	.824	.702	.746
LSTM+CA	**.699**	.637	.662	.966	**.972**	**.969**	**.832**	.805	.815
LSTM+CA+feat	.681	**.665**	**.673**	**.969**	.971	.970	.821	**.814**	**.821**

Таблица 6 — Результаты классификации на корпусе MADE.

	non-ADR			ADR			Макро		
Модель	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
SVM [Sarker et. al.]	.798	.819	.805	.743	**.639**	**.674**	.770	.729	.739
SVM+feat	**.984**	.981	.982	.551	.582	.562	.767	**.782**	.772
CNN	.974	**.996**	.985	.746	.337	.457	.86	.667	.721
RCNN	.980	.989	.984	.649	.485	.535	.815	.737	.760
CRNN	.976	**.996**	.986	.787	.379	.488	**.882**	.687	.737
CNNA	.974	.995	.984	.715	.318	.414	.844	.657	.699
CNN-BiLSTM	.979	.992	.985	.726	.444	.517	.852	.718	.751
BioBERT	.902	.953	.926	.742	.547	.642	.822	.750	.784
LSTM+CA	.982	.991	.986	.740	.524	.585	.861	.758	.786
LSTM+CA+feat	.981	.993	**.987**	**.749**	.519	.589	.865	.756	**.788**

показывает, что модель LSTM+CA+feat способна лучше обучиться на несбалансированных корпусах малого размера, чем модель BioBERT. Наименьший прирост был достигнут на корпусе MADE (+0.4%). На корпусе RuDReC модель LSTM+CA+feat превзошла результаты CNN-BiLSTM на 8% F-меры. Прирост результатов на остальных корпусах по сравнению с базовой моделью варьируется от 0.7% до 2.8%.

Добавление признаков в модель LSTM+CA позволило повысить качество классификации модели на всех корпусах. Наибольший прирост по средней F-мере при добавлении признаков в модель был получен на корпусах PsyTAR

Таблица 7 — Результаты классификации на корпусе TwiMed-Twitter.

	non-ADR			ADR			Макро		
Модель	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
SVM [Sarker et. al.]	.638	.535	.565	**.843**	.843	**.839**	.741	.689	.702
SVM+feat	.779	.707	.739	.752	.810	.778	.766	.758	.758
CNN	.689	.686	.677	.726	.708	.708	.708	.697	.692
RCNN	.700	.750	.721	.757	.698	.722	.728	.724	.722
CRNN	.688	.707	.688	.729	.692	.701	.708	.699	.694
CNNA	.713	.618	.637	.697	.741	.701	.705	.679	.669
CNN-BiLSTM	.742	.718	.720	.758	.761	.753	.750	.739	.736
BioBERT	.814	.750	.781	.786	.843	.813	.800	.796	.797
LSTM+CA	.802	**.825**	.813	.836	.813	.824	.819	.819	.819
LSTM+CA+feat	**.838**	.800	**.816**	.823	**.849**	.834	**.830**	**.825**	**.825**

Таблица 8 — Результаты классификации на корпусе TwiMed-PubMed.

	non-ADR			ADR			Макро		
Модель	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
SVM [Sarker et. al.]	.875	.877	.871	.609	.470	.511	.742	.673	.691
SVM+feat	.925	.955	.939	.799	.681	.728	.862	.818	.834
CNN	.854	.986	.915	.850	.364	.503	.852	.675	.709
RCNN	.866	.978	.918	.835	.438	.570	.851	.708	.744
CRNN	.848	**.993**	.914	**.916**	.342	.491	.882	.668	.703
CNNA	.825	.990	.899	.859	.220	.343	.842	.605	.621
CNN-BiLSTM	.904	.977	.938	.873	.612	.710	.888	.794	.824
BioBERT	.934	.910	.932	.845	**.782**	**.812**	.901	.861	.872
LSTM+CA	.936	.977	**.956**	.878	.738	.792	**.907**	.858	.874
LSTM+CA+feat	**.939**	.970	.954	.871	.753	.803	.905	**.862**	**.878**

(+7.6%) и Twitter (+2.9%). На остальных корпусах прирост по F-мере варьируется от 0.2% до 0.9% F-меры. Стоит отметить, что модель с признаками обучается быстрее и требует на 5 эпох меньше по сравнению с моделью без признаков.

По метрике средней точности модель LSTM+CA+feat показала результат ниже модели LSTM+CA на корпусах CADEC и TwiMed-PubMed на 1.1% и 0.2% соответственно, на остальных корпусах разработанная модель показывает лучшие результаты. Наибольший прирост точности по сравнению с моделью LSTM+CA был достигнут на корпусе PsyTAR (+5.4%), наименьший на кор-

Таблица 9 — Результаты классификации на корпусе PsyTAR.

	non-ADR			ADR			Макро		
Модель	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
SVM [Sarker et. al.]	.694	.570	.612	.841	.866	.850	.768	.718	.731
SVM+feat	.863	**.910**	**.886**	.851	.781	.815	.857	.845	.850
CNN	.841	.813	.826	.880	.897	.888	.860	.855	.857
RCNN	.808	.865	.834	**.908**	.862	.884	.858	.864	.859
CRNN	**.902**	.710	.788	.835	.945	.885	.868	.828	.837
CNNA	.839	.808	.823	.876	.898	.887	.858	.853	.855
CNN-BiLSTM	.834	.835	.834	.891	.889	.890	.863	.862	.862
BioBERT	.866	.854	.860	.904	.913	.909	.884	**.882**	.881
LSTM+CA	.848	.712	.752	.841	.915	.873	.844	.814	.812
LSTM+CA+feat	.895	.808	.849	.899	**.956**	**.927**	**.898**	.878	**.888**

Таблица 10 — Результаты классификации на корпусе RuDReC.

	non-ADR			ADR			Макро		
Модель	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
SVM+feat	.889	.858	.873	.654	.712	.680	.772	.785	.776
CNN	.822	.941	.877	.745	.454	.557	.783	.698	.717
RCNN	.860	.905	.881	.705	.604	.649	.782	.754	.765
CRNN	.835	.899	.865	.663	.520	.580	.749	.710	.723
CNNA	.752	**.989**	.853	.156	.112	.130	.454	.551	.492
CNN-BiLSTM	.875	.901	.887	.714	.656	.681	.794	.778	.784
RuBERT	.867	.914	.888	.752	.620	.664	.809	.767	.776
LSTM+CA	.917	.931	.923	.810	**.773**	.787	.863	.852	.855
LSTM+CA+feat	**.919**	.942	**.928**	**.837**	**.773**	**.798**	**.877**	**.857**	**.864**

пусе MADE (+0.4%). По метрике полноты модель LSTM+CA+feat превзошла остальные модели на всех корпусах, кроме MADE (-0.2%), где наилучшие результаты показала модель LSTM+CA, и PsyTAR (-0.4%), где самых высоких результатов достигла модель BioBERT. Наибольший прирост средней полноты по сравнению с моделью LSTM+CA был достигнут на корпусе Twitter (+1.7%), наименьший на корпусе TwiMed-PubMed (+0.4%).

Модель LSTM+CA+feat превзошла результаты остальных моделей по метрике F-меры для класса ADR на корпусах: Twitter (95.9%), PsyTAR (92.7%) и RuDReC (79.8%). На корпусе TwiMed-PubMed самые высокие результаты показала модель BioBERT (81.2% в сравнении с 80.3%). На корпусах MADE и

TwiMed-Twitter максимальный результат показала модель SVM, предложенная Саркером (67.4% и 83.9%) F-меры. На корпусе CADEC результаты оказались ниже результатов модели LSTM+CA всего на 0.1%. По метрике точности для класса ADR наибольший прирост относительно модели LSTM+CA был получен на корпусах RuDReC (+2.7%) и MADE (+0.9%), на корпусах Twitter и CADEC прирост составил +0.3%. На корпусах TwiMed-Twitter и TwiMed-PubMed модель LSTM+CA превзошла модель LSTM+CA+feat на 1.3% и 0.7% соответственно, а на корпусе PsyTAR модель RCNN показала максимальный результат (90.8%) на 0.7% больше, чем модель LSTM+CA+feat. По метрике полноты наибольший прирост по сравнению с моделью LSTM+CA достигнут на корпусе PsyTAR (+4.1%), на корпусах, состоящих из твитов: TwiMed-Twitter и Twitter, прирост составил +3.6% и +0.7% соответственно. На корпусе Twitter-PubMed модель BioBERT показала самый высокий показатель полноты для класса ADR и обошла модель LSTM+CA+feat на 2.9%. На корпусах CADEC и MADE LSTM+CA превзошла модель LSTM+CA+feat на 0.1% и 0.5% соответственно. На корпусе RuDReC модели LSTM+CA и LSTM+CA+feat превзошли базовые модели, показав результат 77.3%.

Полученные результаты доказывают, что разработанная модель LSTM+CA+feat показывает более высокое качество в задаче классификации сущностей в сравнении с остальными существующими моделями. Добавление признаков позволяет улучшить метрики классификации модели LSTM+CA. Модель LSTM+CA+feat показывает высокие результаты для текстов различных доменов на русском и английском языках, а также несбалансированных наборах данных с небольшим количеством обучающих примеров.

2.4.4 Результаты кросс-доменной оценки

В таблице 11 представлены результаты кросс-доменной оценки моделей в задаче классификации сущностей. В данном наборе экспериментов модели обучались на одном корпусе (source), а оценивались на другом (target). В качестве базовой модели для сравнения была выбрана модель BioBERT, поскольку данная модель показала наилучшие результаты среди базовых моделей при оценке в рамках одного корпуса. Согласно полученным результатам модели по-

Таблица 11 — Результаты кросс-доменной оценки моделей для задачи классификации по метрике средней F-меры. Результаты в рамках одного домена на диагонали. Среднее - это усредненное значение F-меры для всех кросс-доменных экспериментов. Средняя потеря - это разница между результатами в рамках одного домена и среднего значения. Жирным шрифтом выделены максимальные значения каждой метрики.

Target→ Source↓	**Модель**	**Twitter**	**CADEC**	**MADE**	**TwiMed-Twitter**	**TwiMed-PubMed**	**Psy-TAR**	**Сред-нее**	**Средняя потеря**
Twitter	LSTM+CA+feat	**.794**	**.580**	**.108**	**.638**	.414	**.558**	**.459**	-.334
	BioBERT	.589	.531	.048	.626	**.493**	.473	.434	-.155
CADEC	LSTM+CA+feat	**.719**	**.815**	**.284**	**.722**	.670	**.617**	**.602**	-.213
	BioBERT	.592	.746	.203	.666	**.781**	.553	.559	-.187
MADE	LSTM+CA+feat	.242	.112	**.786**	.429	.696	**.491**	**.394**	-.392
	BioBERT	**.269**	**.129**	.784	**.513**	**.743**	.319	**.394**	-.390
TwiMed-	LSTM+CA+feat	.653	.647	**.247**	**.819**	.765	**.697**	**.602**	-.217
Twitter	BioBERT	**.659**	**.676**	.119	.797	**.837**	.623	.583	-.214
TwiMed-	LSTM+CA+feat	**.518**	.570	**.365**	.728	**.874**	**.754**	**.587**	-.287
PubMed	BioBERT	.491	**.662**	.170	**.735**	.872	.679	.547	-.325
PsyTAR	LSTM+CA+feat	**.577**	.643	**.293**	.602	.600	.812	**.543**	-.269
	BioBERT	.533	**.667**	.168	**.624**	**.711**	**.881**	.541	-.340

казывают существенно ниже качество в случаях, когда обучающий и тестовый набор данных из разных доменов, чем когда оценка проводится в рамках одного корпуса. Падение показателя макро F-меры варьируется от 18.7% до 39.2%. Стоит отметить, что при обучении моделей на корпусе MADE наблюдается наибольшая потеря в качестве при кросс-доменной проверке: -39.2% для модели LSTM+CA+feat и - 39% для модели BioBERT. Такой результат обусловлен разницей между языковой моделью, используемой в текстах ЭМК пациентов и текстами социальных медиа.

Модель LSTM+CA+feat превосходит модель BioBERT по среднему значению F-меры на всех корпусах, кроме MADE, где модели показывают одинаковую среднюю F-меру - 39.4%. Наибольший прирост модели LSTM+CA+feat по сравнению с BIoBERT в среднем был достигнут при обучении моделей на корпусе CADEC (+4.3%), наименьший при обучении на корпусе PsyTAR (0.2%). Стоит отметить, что при кросс-доменной оценке модель BioBERT превзошла модель LSTM+CA+feat на корпусе TwiMed-PubMed, независимо от корпуса, на котором модели обучались. Это обусловлено тем, что изначально модель

BioBERT обучалась на текстах научных статей. Максимальный прирост модели LSTM+CA+feat по сравнению с BioBERT был достигнут в эксперименте TwiMed-PubMed-MADE (+19.5%).

Полученные результаты показывают, что модель LSTM+CA+feat более эффективна в случае, когда обучающие и тестовые данные из разных доменов, что в свою очередь доказывает большую возможность практического применения модели LSTM+CA+feat по сравнению с BioBERT.

2.4.5 Оценка модели, обученной на всех корпусах

Исследования показывают, что увеличение объема обучающей выборки приводит к улучшению качества классификации. В связи с этим были проведены дополнительные эксперименты, в которых на каждом фолде модель LSTM+CA обучалась на совокупности обучающих данных шести англоязычных корпусов и затем предсказывала метки для тестовых подвыборок каждого корпуса. Результаты представлены в виде гистограммы на рисунке 2.8.

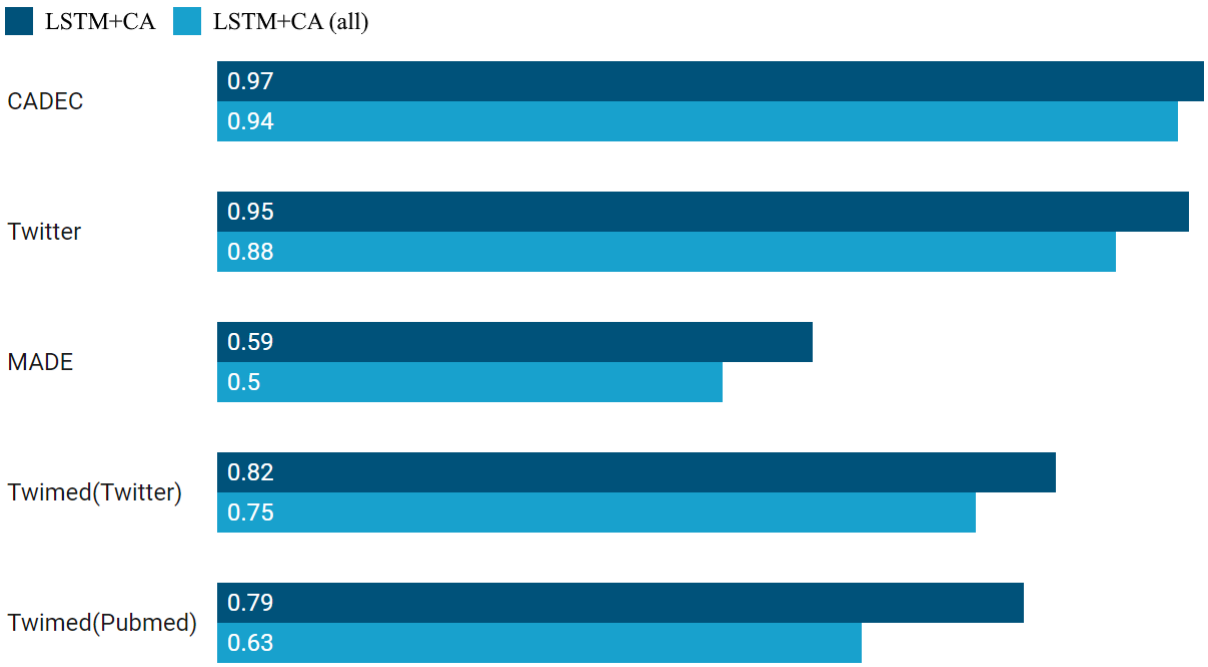


Рисунок 2.8 — Результаты F-меры для класса ADR модели LSTM+CA, обученной на одном корпусе (LSTM+CA) и модели, обученной на совокупности англоязычных корпусов (LSTM+CA (all)).

Результаты показывают, что обучение на совокупности корпусов не дает прироста в результатах. Отсутствие увеличения результатов может быть обусловлено двумя факторами. Во-первых, все корпуса содержат информацию о различных лекарственных препаратах, которые могут вызывать различные побочные эффекты. Во-вторых, лексика, используемая в корпусах существенно различается, в научных статьях и картах пациентов структура текста более формализована, в то время как в социальных медиа используется разговорный стиль и может присутствовать большое количество грамматических ошибок. В таблице 12 представлена статистика пересечения слов в корпусах, подтверждающая данную гипотезу. Корпуса MADE и TwiMed-PubMed имеют наибольший процент пересечения (66.5%). Корпус PsyTAR пересекается больше всего с корпусами, состоящими из твитов: Twitter (65.33%) и TwiMed-Twitter (64.53%). В то же время пересечение корпуса PsyTAR с корпусом CADEC составляет всего 41.79%, несмотря на то, что оба корпуса состоят из отзывов с одного и того же форума askapatient.com. Однако при этом в корпусах собраны отзывы о различных лекарственных препаратах, что могло послужить причиной большой разницы в используемой лексике. Корпус CADEC больше всего пересекается с корпусом Twitter (60.82%).

Таблица 12 — Процент пересечения уникальных слов корпусов. Жирным шрифтом выделены максимальные значения.

Корпус	**CADEC**	**MADE**	**PsyTAR**	**TwiMed-PubMed**	**TwiMed-Twitter**	**Twitter**
CADEC	100.00	16.98	41.79	41.70	60.15	60.82
MADE	16.98	100.00	56.85	**66.50**	61.15	57.06
PsyTAR	41.79	56.85	100.00	45.25	**64.53**	**65.33**
TwiMed-PubMed	41.70	**66.50**	45.25	100.00	50.49	43.24
TwiMed-Twitter	60.15	61.15	64.53	50.49	100.00	39.18
Twitter	**60.82**	57.06	**65.33**	43.24	39.18	100.00

2.5 Оценка модели с различными векторными представлениями слов

Выбор векторного представления слов может существенно повлиять на результаты работы модели. Вектора, обученные на текстах из наиболее близкого домена способствуют более высокому качеству работы моделей. В данном разделе описаны эксперименты и результаты оценки модели LSTM+CA с различными векторными представлениями.

Для русскоязычного корпуса были выбраны две существующие модели векторного представления слов: Ruscorpora и Tayga. В дополнение были обучены несколько моделей на текстах отзывов о различных продуктах из неразмеченной части корпуса RuDReC:

- лекарственные препараты (Drugs),
- косметические средства (Beauty),
- медицинские учреждения (Hospitals),
- лекарственные препараты и косметические средства (Drugs+Beauty),
- лекарственные препараты, косметические средства и медицинские учреждения (Drugs+Beauty+Hospitals).

Модель Ruscorpora обучена на национальном корпусе русского языка с помощью алгоритма непрерывный мешок слов (Continuous Bag of Words; CBOW). Модель обучалась со следующими параметрами: размерность векторов - 300, размер окна - 20, частотный порог - 5, размер словаря - 189193. Модель Tayga обучалась на текстах корпуса Тайга, включающем в себя почти 5 миллиардов слов. Модель обучалась с применением алгоритма fasttext со следующими параметрами: размерность векторов - 300, размер окна - 10, частотный порог - 5, размер словаря - 192415.

Модель Drugs, Beauty и Hospitals обучались на отзывах о лекарственных препаратах, косметических средствах и медицинских учреждениях. Отзывы о лекарственных препаратах были собраны с сайтов Отзовик и ПроТаблетки. Суммарное количество отзывов - 247282. Отзывы о косметических средствах были собраны с сайта Отзовик. Суммарное количество отзывов о косметических средствах - 466199. Модель Hospitals обучалась на 635371 тексте отзывов о медицинских учреждениях, собранных с сайта “Справочник предприятий Москвы и

Московской области” [15]. Общее количество слов в корпусе - 58353690. Параметры моделей: размерность векторов - 200, размер окна - 10, частотный порог - 5.

Результаты работы модели LSTM+CA с разными входными векторными представлениями слов представлены в таблице 13. Из результатов видно, что модели, обученные на корпусах текстов общей тематики показали результаты ниже, чем модели, обученные на отзывах из медицинского домена. Модели с векторами Drugs и Beauty показали значение F-меры на одном уровне - 83.6%, в то время как модель с векторами Hospitals показала на 0.2% F-меры ниже. Модель с векторами, обученными на корпусах Drugs и Beauty, показала наивысший результат среди всех сравниваемых моделей - 85.5% F-меры. Данный показатель превзошел результаты модели с векторами, обученными на всех корпусах отзывов, на 1%.

Таблица 13 — Результаты классификации на корпусе RuDReC для модели LSTM+CA для различных векторных представлений слов.

	non-ADR			**ADR**			**Макро**		
Метод	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Ruscorpora	.877	.889	.882	.696	.655	.665	.786	.772	.774
Tayga	.852	.930	.888	.772	.555	.624	.812	.742	.756
Drugs	**.922**	.895	.907	.746	**.795**	.765	.834	.845	.836
Beauty	.904	.925	.914	.796	.735	.758	.850	.830	.836
Hospitals	.919	.895	.906	.744	.790	.762	.831	.842	.834
Drugs+Beauty	.917	**.931**	**.923**	**.810**	.773	**.787**	**.863**	**.852**	**.855**
Drugs+Beauty+Hospitals	.918	.916	.915	.789	.778	.775	.853	.847	.845

2.6 Выбор оптимального семантического представления контекста относительно сущности

Алгоритмы обучения с учителем для задачи классификации на уровне сущностей сталкиваются с тремя основными проблемами: (i) представление контекста сущности, (ii) генерация представления классифицируемой сущности с привязкой к контексту, (iii) определение наиболее важных слов, которые

[15] https://moskva.rosfirm.ru/catalog

влияют на присвоение класса. В данном случае в качестве контекста рассматривается предложение, в котором встретилась сущность. Контекст играет одну из ключевых ролей в задаче идентификации побочных эффектов. Например, в предложении: “Он не мог спать прошлой ночью из-за боли” (оригинал на английском: “He was unable to sleep last night because of pain”) сущность “не мог спать” была следствием боли, поэтому она классифицируется как не побочный эффект, а в предложении “стало невозможно ходить без трости, не могу спать, проблемы с почками (моча, такая как пиво)” (оригинал на английском: “Became unable to walk without a cane, unable to sleep, kidney problems (urine like root beer)”), сущность «не могу спать» - это побочный эффект от приема лекарства. Данный параграф посвящен оценке различного представления контекста и сущности относительно друг друга.

2.6.1 Описание моделей

Существующие работы показали успешность применения ряда архитектур нейронных сетей, основанных на сетях с короткой долгосрочной памятью (англ. long short-term memory; LSTM) [32]. В качестве моделей для сравнения с выбранной архитектурой LSTM+CA были взяты следующие архитектуры нейронных сетей:

1. сеть с короткой долгосрочной памятью (англ. long short-term memory; LSTM) - базовая модель, которая использует все предложение, закодированное векторным представлением слов, в качестве входа;
2. модель с заданной целью (англ. Target-Dependent LSTM; TD_LSTM) [175] которая использует два слоя LSTM для моделирования правого и левого контекста относительно сущности;
3. сеть с глубокой памятью (Deep Memory Network; MemNet) [176], которая применяет несколько раз механизм внимания к входному слою векторного представления слов, выход последнего из которых передается в слой с логистической функцией для предсказания класса;
4. сеть с рекуррентным механизмом внимания к памяти (Recurrent Attention Memory; RAM) [177] расширяет модель MemNet дополни-

тельными слоями LSTM и многократным применением механизма внимания к выходам этих слоев.

Далее приводится подробное описание моделей.

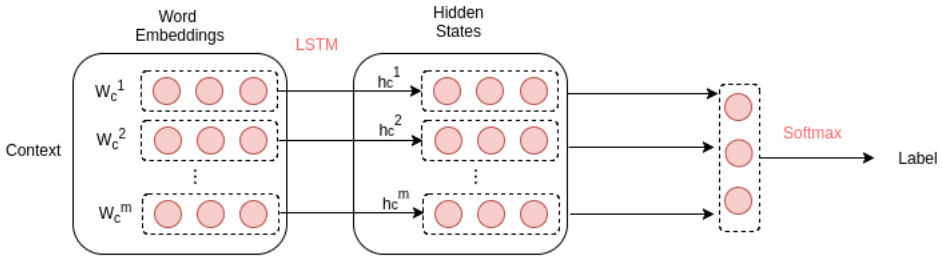


Рисунок 2.9 — Общая архитектура модели LSTM.

LSTM - классическая нейронная сеть, являющаяся разновидностью рекуррентных нейронных сетей, была представлена в [32]. Сеть состоит из трех слоев: входного, слоя с короткой долгосрочной памятью и выходного. Общая архитектура сети представлена на рисунке 2.9.

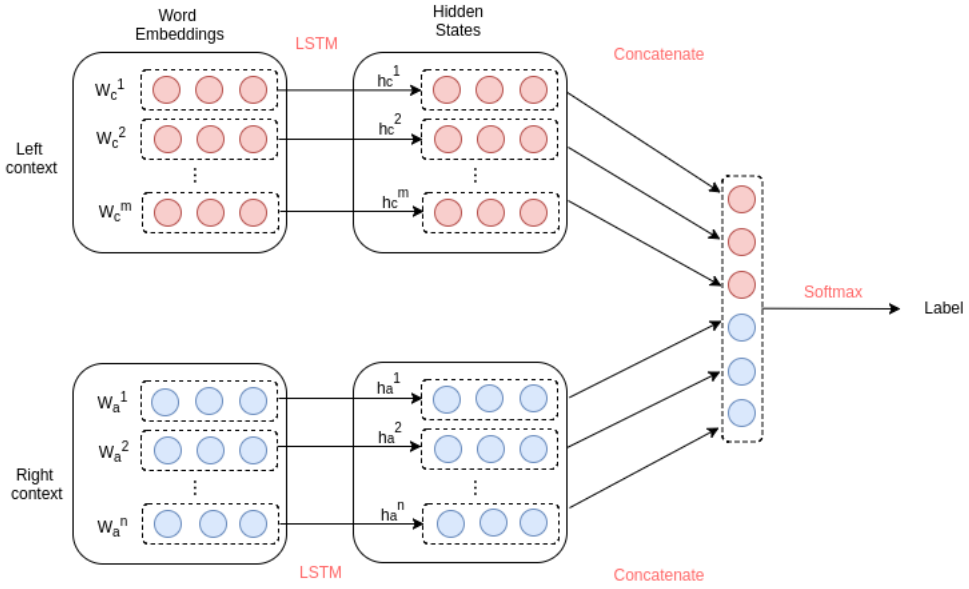


Рисунок 2.10 — Общая архитектура модели TD_LSTM.

Модель **TD_LSTM** была предложена в работе [176] и является расширением предыдущей модели. Модель состоит из двух частей, каждая из которых обрабатывает левый и правый контексты соответственно. Аналогично с предыдущей моделью входные тексты попадают в слой векторного представления слов, выходы которого передаются в LSTM слой. Вектора скрытых состояний LSTM слоев для левого и правого контекстов конкатенируются в один вектор. К полученному вектору, так же как и в предыдущей модели, применяется

нелинейный слой с функцией $softmax$ и вычисляется класс с наибольшей вероятностью. Общая архитектура сети представлена на рисунке 2.10.

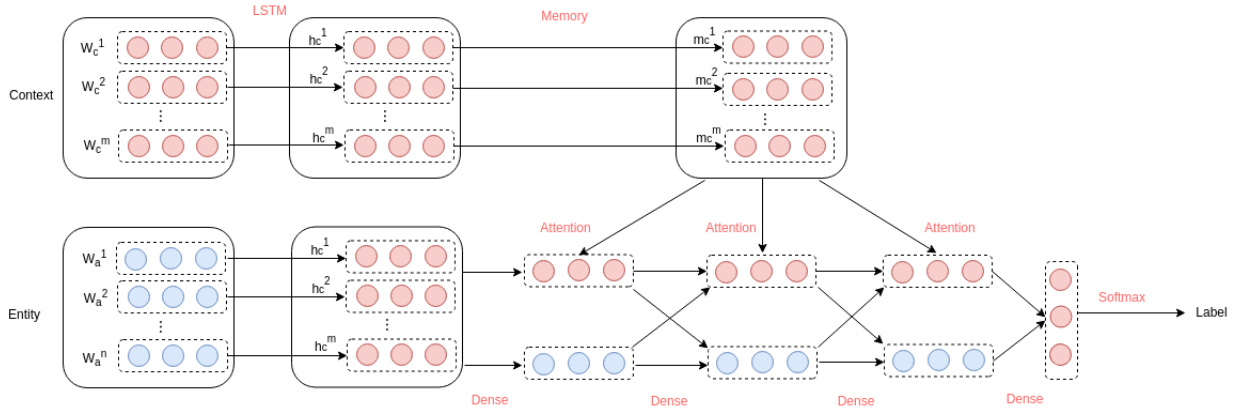


Рисунок 2.11 — Общая архитектура модели RAM.

Сеть с **RAM** рекуррентным механизмом внимания к памяти была представлена Ченом с соавторами [177]. В данной модели используются двунаправленные слои LSTM. В двунаправленном LSTM происходит два прохода по слою: прямой и обратный. Прямой проход описан в модели LSTM, обратный - по содержанию аналогичен прямому, однако на вход принимаются развернутые в обратном порядке вектора слов. На выходе из данного слоя получаются два вектора скрытых состояний: полученный прямым проходом $\overrightarrow{h^i}$ и обратным $\overleftarrow{h^i}$. На вход сети подается контекст и целевая сущность. Оба входа представляются в виде векторного представления слов, а затем подаются на вход слою с двунаправленной LSTM. Выход слоя для контекста сохраняется во внешнюю память в виде векторов $M = \{m_1, m_2, \ldots, m_n\}$, где m_i состоит из объединенных векторов скрытых состояний двунаправленной LSTM для контекста: $\overrightarrow{h_c^i}$ и $\overleftarrow{h_i}$. Полученные вектора памяти и объединенные вектора скрытых слоев LSTM для целевой сущности подаются на вход слоям с механизмом внимания. На каждом шаге выход из слоя внимания проходит через управляемые рекуррентные блоки (англ. Gated Recurrent Unit; GRU). GRU - механизм вентилей для рекуррентных нейронных сетей, представленный в 2014 году [33]. По сравнению с LSTM у данного механизма меньше параметров, так как отсутствует выходной вентиль. Допустим, e_{i-1} - вектор, полученный на предыдущем шаге слоя GRU и t_i - текущее состояние, полученное применением механизма внимания к вектору памяти, тогда текущий вектор e_i будет высчитываться по формулам:

$$r = \sigma(W_r t_i + U_r e_{i-1})$$

$$z = \sigma(W_z t_i + U_z e_{i-1})$$

$$\tilde{e_i} = tanh(W_x t_i + W_g(r \odot e_{i-1}))$$

$$e_i = (1 - z) \odot e_{i-1} + z \odot \tilde{e_i}$$

где $W_r, W_z, U_r, U_z, W_g, W_x$ - веса слоя GRU. В качестве e_0 используется вектор из нулей. Для вычисления значения t_i используется вектор из памяти m_j и предыдущий выход работы слоя GRU e_{i-1}. На первом шаге подсчитывается функция оценки для каждого слова в памяти:

$$\gamma_j^i = W_a^i(m_j, e_{i-1}, h_e) + b_a^i$$

где h_e - выход двунаправленного LSTM слоя для контекста, а матрицы W_a и b_a веса слоя внимания. Далее подсчитывается нормализованная оценка внимания:

$$\alpha_k^i = \frac{exp(\gamma_k^i)}{\sum_{j=1}^{n} exp(\gamma_k^i)}$$

Значение, подающееся на вход GRU слою вычисляется по формуле:

$$t_i = \sum_{j=1}^{n} \alpha_j^i m_j$$

Значение, полученное после заданного количества итераций подается на вход полносвязному слою с функцией softmax для дальнейшей классификации. Оптимизация и подсчет функции потерь в процессе обучения происходит аналогичным с моделью LSTM+CA образом.

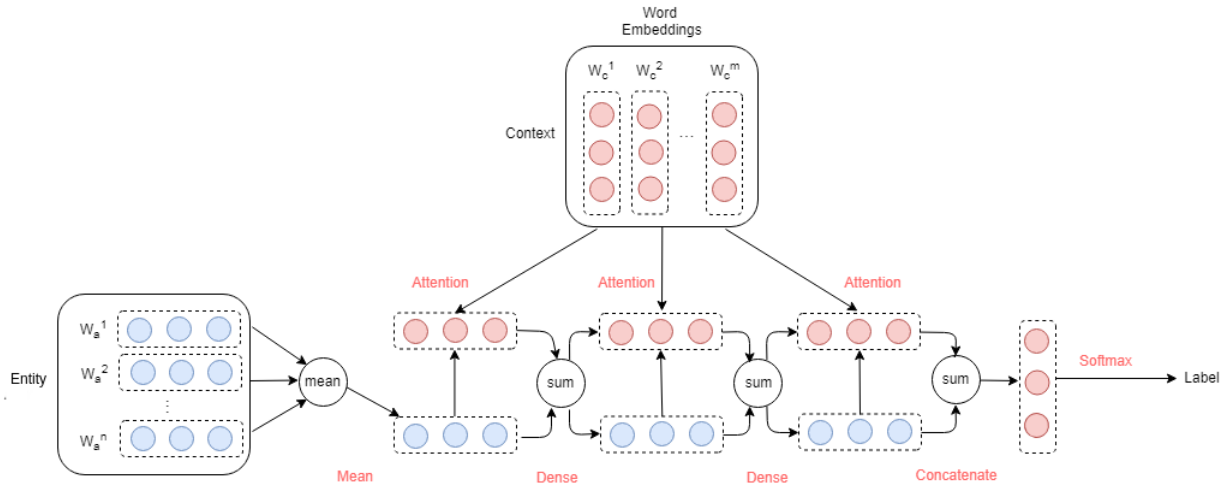


Рисунок 2.12 — Общая архитектура модели MemNet.

Модель **MemNet** была представлена Тангом с соавторами [176]. Данная модель состоит из двух главных частей: модуля памяти, который хранит в себе

входные данные для контекста в виде распределенного представления слов и механизма внимания. На вход слою с вниманием подаются сущность в виде векторного представления слов и вектора, сохраненные в памяти. Выход из слоя памяти суммируется с векторами памяти и подается в следующий слой с механизмом внимания. Общая архитектура сети представлена на рисунке 2.12.

На вход сети так же как и модели LSTM+CA подаются контекст $[w_c^1, w_c^2, ..., w_c^n]$ и целевая сущность $[w_e^1, w_e^2, ..., w_e^n]$. Контекст кодируются в векторное представление слов. Матрица, полученная для контекста сохраняется во внешней памяти $m = R^{dxn}$, где d - размерность векторов. Целевая сущность представляется одним вектором v_e, полученным усреднением векторов слов, входящих в эту сущность. Полученная матрица m и вектор целевой сущности v_e подаются на вход слою внимания, который на выход отдает вектор a. Для вычисления этого вектора на первом шаге вычисляется функция оценки:

$$\gamma_i = tanh(W_a \cdot m_i \cdot v_e + b_a)$$

Полученные значения $\gamma_1, \gamma_2, \ldots, \gamma_n$ подаются на вход слою с функцией $softmax$ для подсчета коэффициентов важности каждого из векторов контекста:

$$\alpha_i = \frac{exp(\gamma_i)}{\sum_{j=1}^{n} exp(\gamma_j)}$$

На основе полученных значений вычисляется итоговый вектор внимания:

$$a = \sum_{i=1}^{n} \alpha_i m_i$$

Далее операция с применением внимания повторяется еще два раза, но уже для вектора a и матрицы памяти m. Полученный в результате всех итераций вектор внимания a отдается полносвязному слою с функцией $softmax$ для классификации. Оптимизация и подсчет функции потерь в процессе обучения происходит аналогичным с моделью LSTM+CA образом.

2.6.2 Результаты оценки моделей

Для кодирования входного текста векторным представлением использовалась модель векторного представления слов, обученная на записях из социальных медиа [161]. Для русского языка использовалась модель Ruwikiruscorpora,

Таблица 14 — Макро усредненная F-мера для сравниваемых моделей с механизмом памяти и внимания.

Модель	**Twitter**	**Cadec**	**MADE**	**TwiMed-Twitter**	**TwiMed-PubMed**	**PsyTAR**	**RuDReC**
LSTM+CA	.794	**.815**	**.817**	**.819**	**.874**	812	**.855**
LSTM	.613	.784	.771	.700	.839	.861	.703
TD_LSTM	.758	.772	.750	.730	.709	**.878**	.735
RAM	**.834**	.734	.761	.780	.789	.809	.622
MemNet	.763	.758	.760	.795	.811	.638	.671

обученная на Национальном корпусе русского языка и текстах Википедии. Для обучения каждой модели на каждом из корпусов было использовано 15 эпох, размер входного блока 128 для корпусов CADEC и MADE и 32 для остальных корпусов, количество скрытых состояний - 300, шаг обучения (learning rate) - 0.01, l2 регуляризация со значением 0.001. В ходе экспериментов модели с данным набором параметров показали наиболее высокие результаты. Для реализации моделей был использован публично доступный код из репозитория [16]. Все модели были оценены на 5-фолдовой кросс валидации.

Результаты оценки макро усредненной F-меры представлены в таблице 14, метрики усредненной полноты и точности, а также метрики по классам представлены в приложении А . Из результатов видно, что на всех корпусах, кроме Twitter и PsyTAR лучшие результаты по макро F-мере показала модель LSTM+CA. Наиболее значимый прирост качества по сравнению с другими моделями был получен на корпусах TwiMed-Twitter и Twitter-Pubmed, где модель LSTM+CA достигла 81.9% и 87.4% макро F-меры соответственно. На корпусе Twitter лучшие результаты показала модель RAM с усредненной макро F-мерой 83.4%, на корпусе PsyTAR модель TD_LSTM с результатом 87.8% F-меры.

Исходя из полученных результатов, можно сделать вывод, что разделение входного предложения на правый и левый контекст относительно выделенной сущности улучшило качество классификации для корпусов, состоящих из твитов, а также на текстах отзывов корпуса PsyTAR и RuDReC. Это следует из того, что модель TD_LSTM превзошла модель LSTM на данных корпусах. Наибольшее прирост результатов усредненной F-меры при разделении контекстов был достигнут на корпусе Twitter (+14.5%), наименьший - на корпусе PsyTAR

[16] https://github.com/songyouwei/ABSA-PyTorch

(+1.7%), на корпусах TwiMed-Twitter и Отзовик - 3% и 3.2%, соответственно. Для остальных корпусов разделение контекста не смогло улучшить результатов. На корпусе TwiMed-PubMed LSTM превзошла модель TD_LSTM на 7% по метрике F-меры. На остальных корпусах результаты сравнимы и отличаются всего на 2%.

Сравнение результатов работы моделей RAM и MemNet показывают, что наличие LSTM слоя перед слоем с памяти оказалось эффективно только на двух корпусах Twitter и PsyTAR, где RAM показала существенно выше результаты F-меры (83.4% и 80.9% соответственно) по сравнению с MemNet (76.3% и 63.8%).

Превосходство LSTM+CA по сравнению с RAM и MemNet на пяти из семи корпусов также показывает, что наличие дополнительной памяти далеко не всегда дает преимущество. Результаты модели LSTM+CA также превзошли результаты модели TD_LSTM, что показывает эффективность слоя кросс-внимания. Преимущество модели LSTM+CA по сравнению с моделью LSTM показывает необходимость разделения контекста и сущности.

2.7 Апробация модели на большом корпусе отзывов о лекарственных препаратах

В данном пункте описана апробация модели на большом корпусе отзывов о лекарственных препаратах. Оценка качества модели на реальных данных проводилась по следующему алгоритму:

1. С помощью автоматических методов были извлечены сущности, описывающие состояние здоровья, в большом корпусе текстов о лекарственных препаратах.
2. Размеченные сущности были классифицированы с помощью разработанной модели LSTM+CA+feat для выделения сущностей, описывающих побочный эффект.
3. Для отзывов о трех лекарственных препаратах были выбраны все побочные эффекты, которые извлекла система.
4. Извлеченные побочные эффекты сопоставлялись с побочными эффектами, описанными в инструкциях к лекарственным препаратам.

Разметка проводилась на корпусе отзывов, собранных с сайта Отзовик. Для разметки сущностей использовалась модель BERT [77]. Обе модели BioBERT и LSTM+CA+feat были обучены на четырехстах размеченных текстах корпуса Отзовик. Для анализа были выбраны три лекарственных препарата: Антисептический крем Судокрем - 70 отзывов, мазь для наружного применения Astellas Пимафукорт - 13 отзывов, антибиотик Азитромицин - 18 отзывов.

На рисунках 2.13, 2.14, 2.15 представлены результаты. На рисунках в центральном круге - название лекарственного препарата, на остальных кругах - побочные эффекты из инструкций к лекарственным препаратам, в прямоугольниках извлеченные сущности, описывающие побочные эффекты с указанием количества упоминаний в текстах. В категорию Остальное определены сущности, которые не удалось сопоставить с побочными эффектами, указанными в инструкции.

Рисунок 2.13 — Отображение извлеченных из отзывов сущностей о побочных эффектах на побочные эффекты, указанные в инструкции, для лекарства Судокрем.

Для Судокрема наиболее распространенный побочный эффект - покраснения, который в большинстве случаев описывается сущностью, совпадающей с показаниями в инструкции - “покраснения” (32), в одном случае уточняется - “покраснения кожи” и в одном отзыве - “краснота”. В нескольких отзывах встретились побочные эффекты, связанные с зудом и высыпаниями. В категории побочных эффектов, которые не были найдены в инструкции попали сущности:

“аллергия”, “раздражение кожи”, “раздражение”. Стоит отметить, что данные понятия могут включать в себя такие симптомы, как покраснение, зуд и высыпания, соответственно, сопоставить их с определенным пунктом из инструкции к препарату затруднительно, однако нельзя сказать, что данные сущности были выделены ошибочно.

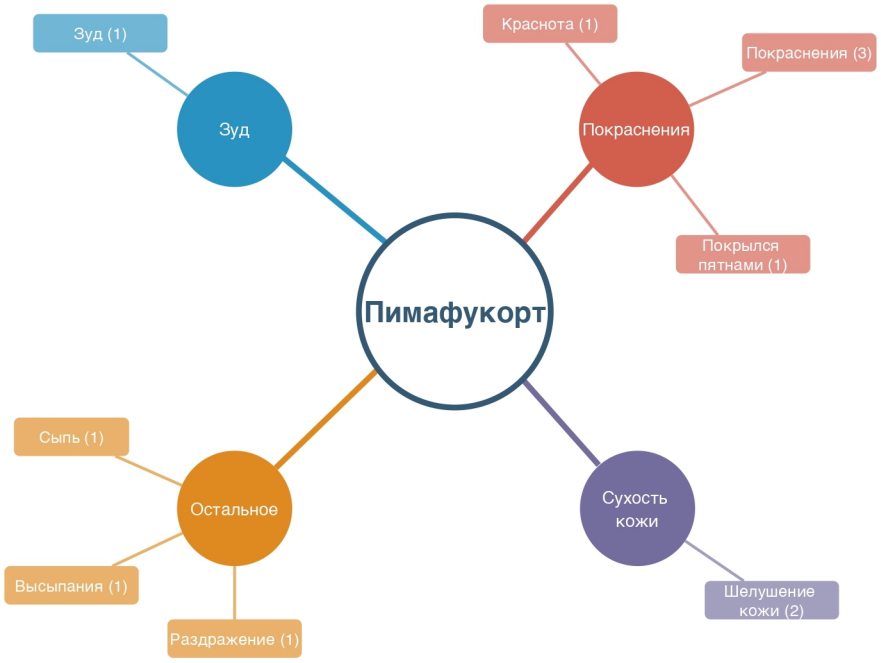


Рисунок 2.14 — Отображение извлеченных из отзывов сущностей о побочных эффектах на побочные эффекты, указанные в инструкции, для лекарства Пимафукорт.

Похожая картина наблюдается для препарата Пимафукорт. Больше всего упоминается побочные эффекты, описывающие покраснения. В двух отзывах встретился побочный эффект - сухость кожи, который описан сущностью “шелушение кожи”, и в одном отзыве зуд. Сущности “сыпь”, “высыпание” и “раздражение” не были найдены в инструкции, но при этом они могут быть отнесены к классу аллергических реакций, описанных в инструкции.

Для препарата Азитромицин наиболее характерными оказались побочные эффекты, связанные с работой желудочно-кишечного тракта: боль в животе, дискомфорт в желудке, тошнота и т.д. В трех отзывах пользователи жаловались на “шум в ушах”, при этом в инструкции данный побочный эффект характеризуется как редкий. В категорию сущностей, не найденных в побочных эффектах из инструкции, были отнесены: “кашлять”, “не мог уснуть”, “температура” и “учащенное сердцебиение”. Данные сущности вероятно описывают симптомы

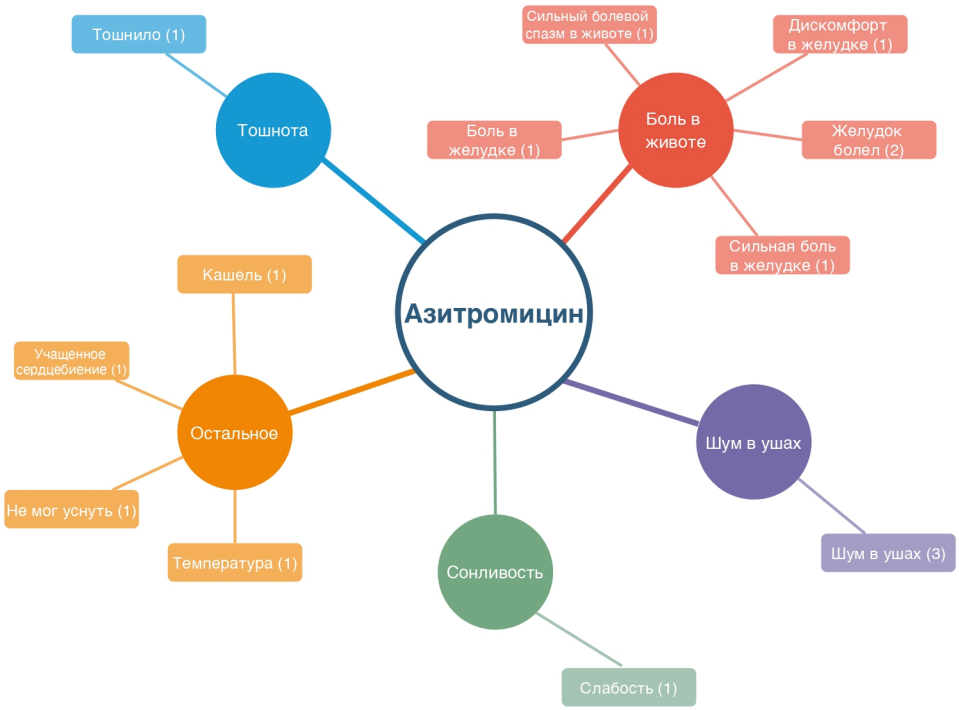


Рисунок 2.15 — Отображение извлеченных из отзывов сущностей о побочных эффектах на побочные эффекты, указанные в инструкции, для лекарства Азитромицин.

заболеваний из показаний к применению, например, инфекции верхних и нижних отделов дыхательных путей, лор-органов.

Анализ результатов показывает, что модель способна выделить сущности, которые описывают побочные эффекты, присутствующие в инструкции к препаратам. При этом поиск точного совпадения слов из инструкции приведет к потере важных данных, так как пользователи используют термины, отличные от медицинских, как например “шелушение кожи” вместо “сухость кожи”. Таким образом, подтверждается эффективность и актуальность разработанной модели.

2.8 Выводы ко второй главе

В данной главе рассматривалась задача классификации сущностей. Для решения поставленной задачи была разработана модель LSTM+CA+feat, основанная на нейронной сети с интерактивным вниманием, принимающей на вход векторное представление слов и набор наиболее информативных призна-

ков. Качество разработанной модели оценивалось на пяти англоязычных и одном русскоязычном корпусе, состоящих из текстов отзывов о лекарственных препаратах, ЭМК пациентов и текстов научных статей. Проведенные эксперименты показали преимущество разработанной модели по сравнению с базовыми подходами, основанными на современных моделях нейронных сетей. В дополнение была проведена качественная оценка качества работы модели на большом корпусе не размеченных текстов отзывов пользователей о лекарственных препаратах, которая показала, что модель способна извлекать побочные эффекты, описанные в инструкциях к препаратам.

Глава 3. Метод извлечения отношений между сущностями

В данной главе будет дана формальная постановка задачи извлечения отношений между сущностями, описание разработанных методов для решения поставленной задачи для русского и английского языков, полученные результаты и выводы. Разработанная модель LSTM+CA основана на нейронной сети LSTM с механизмом внимания, принимающей на вход векторное представление сущностей и контекста между сущностями. Выходы всех входных слоев проходят через LSTM слои, которые формируют семантическое представление входных текстов. Далее к полученным выходам LSTM слоев применяется механизм внимания: вычисляется два вектора внимания относительно каждой сущности и контекста. Полученные вектора представления контеста конкатенируются и подаются на вход полносвязному слою с функцией активации softmax для дальнейшей классификации. Оценка разработанной модели проводилась на корпусах, состоящих из текстов аннотаций научных статей по теме медицины и текстов электронных карточек пациентов. На первом этапе модель сравнивалась с несколькими базовыми методами, при этом оценка проводилась в рамках одного корпуса. Следуя последним исследованиям по извлечению отношений в биомедицинских текстах [9] На втором этапе проводилась кросс-доменная оценка разработанной модели для англоязычных корпусов и сравнение с лучшей базовой моделью, выявленной на первом этапе.

3.1 Формальная постановка задачи

Каждая текстовая коллекция состоит из набора документов $D = \{d_1, d_2, \ldots, d_n\}$. В каждом документе выделен некоторый набор сущностей $e^1 = \{w_1^1, w_2^1, ..., w_{|e^1|}^1\}$, $e^2 = \{w_1^2, w_2^2, ..., w_{|e^2|}^2\}$, ... $e^n = \{w_1^n, w_2^n, ..., w_{|e^n|}^n\}$. Сущности e_i и e_j находятся в отношении, если в тексте указано на их семантическое взаимодействие или влияние друг на друга. Постановка задачи звучит следующим образом: для каждой пары сущностей, принадлежащих одному документу: $e_i^k, e_j^k \in d_k, i,j \in [1, |e^k|], i \neq j$ необходимо определить, есть ли между ними отношение $r(e_i^k, e_j^k) = 1$ или нет $r(e_i^k, e_j^k) = 0$. Например, в предложении “Пациент

принимал 4 дня Руксиенс и в последний день Циклофосфамид" сущности "4 дня" и "Руксиенс" связаны друг с другом, в то время как сущности "4 дня" и "Циклофосфамид" не связаны. Таким образом, r("4 дня", "Руксиенс") = 1, а r("4 дня", "Циклофосфамид") = 0.

Примеры сущностей, для которых $r(e_i^k, e_j^k) = 1$ (сущности выделены курсивом):

- Lyrica and was thought perhaps that this had exacerbated her *autoimmune hemolytic anemia*.
- He finished *6-month* course of *Coumadin* after initially being treated with Lovenox.
- *Velcade* and *thalidomide* will be held for 1 month.

3.2 Модель LSTM+CA для извлечения отношений

Общая архитектура сети LSTM+CA для извлечения отношений представлена на рисунке 3.4. Сеть LSTM+CA имеет три входных слоя. Два входных слоя принимают на вход закодированные векторным представлением сущности относительно которых производится классификация на наличие отношения. Третий входной слой принимает на вход контекст между сущностями, закодированный векторным представлением слов и признаком позиции. Признак позиции (position embedding) широко используется в задаче извлечения отношений и основан на предположении, что близкие к целевым сущностям слова, обычно более информативны при определении связи между сущностями. Для определения признака позиции строятся два вектора, в которых определяется относительное расстояние от каждого токена в контексте до каждой сущности. Если токен в контексте стоит в тексте после сущности, то относительная позиция является положительным числом, в противном случае позиция характеризуется отрицательным числом. На основе полученных относительных позиций строились векторные представления каждой позиции длины 5. Изначально вектора инициализировались случайным образом, затем наиболее оптимальные значения векторов подбирались в процессе обучения сети.

Выходы всех входных слоев проходят через LSTM слои, которые формируют семантическое представление входных текстов. Далее к полученным

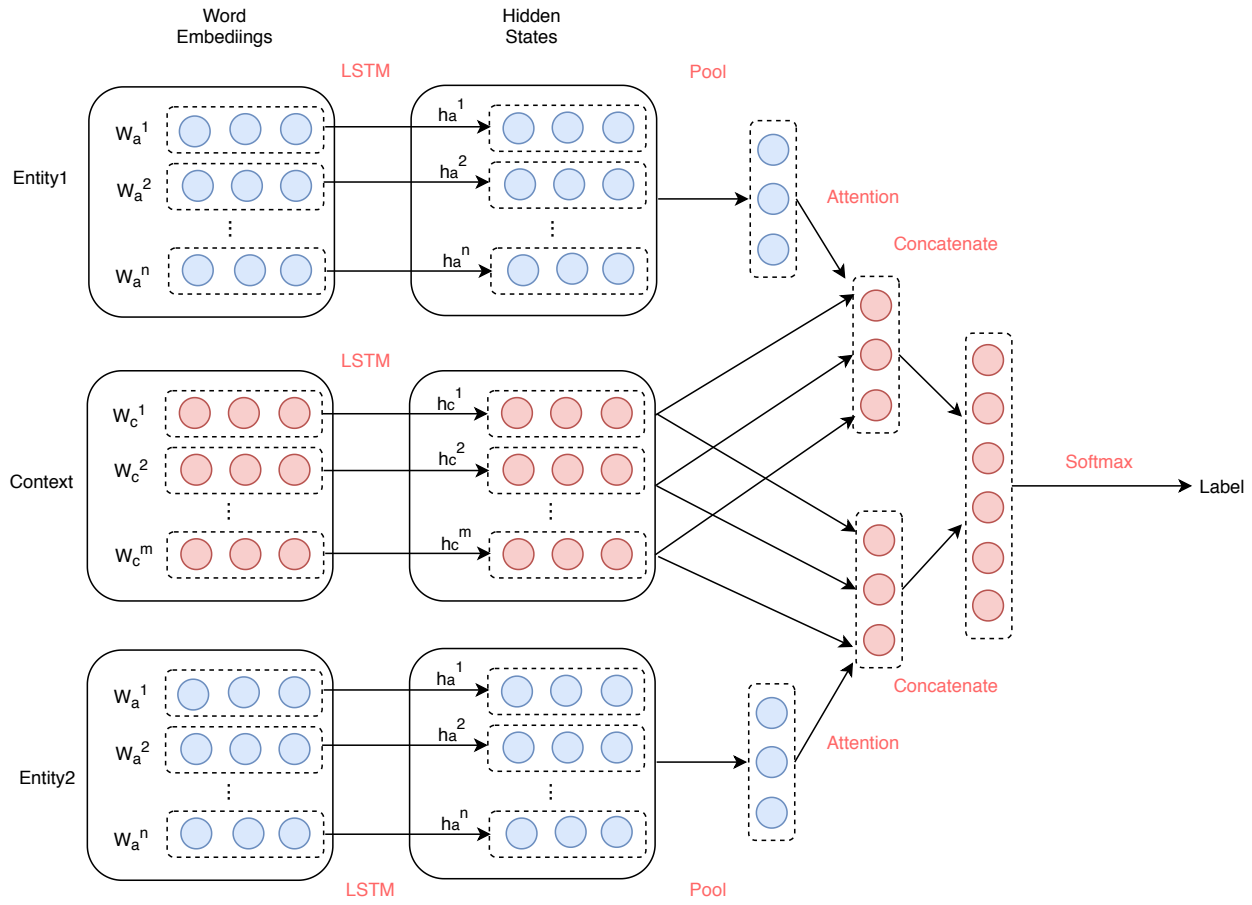


Рисунок 3.1 — общая архитектура модели LSTM+CA для извлечения отношений

выходам LSTM слоев применяется механизм внимания. Механизм внимания реализован следующим образом. Пусть $[h_c^1, h_c^2, ..., h_c^n]$ - скрытое представление контекста, полученное из LSTM слоя, $h_{e_1} = [h_{e_1}^1, h_{e_1}^2, ..., h_{e_1}^n]$ и $h_{e_2} = [h_{e_2}^1, h_{e_2}^2, ..., h_{e_2}^n]$ - скрытые представление сущностей. Для каждой сущности вычисляется среднее значение:

$$e_{1avg} = \frac{1}{m} \sum_{i=1}^{n} h_{e_1}^i$$

$$e_{2avg} = \frac{1}{m} \sum_{i=1}^{n} h_{e_2}^i$$

Для вектора контекста $[h_c^1, h_c^2, ..., h_c^n]$ генерируется вектор внимания $\alpha 1$ относительно каждой сущности с использованием среднего значения вектора сущности e_{1avg} и e_{2avg}. Вектор внимания для первой сущности вычисляется по формуле:

$$\alpha_{1i} = \frac{exp(\gamma(h_c^i, e_{1avg}))}{\sum_{j=1}^{n} exp(\gamma(h_c^j, e_{1avg}))}$$

где γ - функция оценки, которая показывает степень важности h_c^i в контексте и вычисляется по формуле:

$$\gamma(h_c^i, e_{1avg}) = tanh(h_c^i \cdot W_a \cdot e_{1avg}^T + b_a)$$

где W_a и b_a - матрица веса слоя и матрица смещения соответственно. $tanh$ - обозначает нелинейную функцию гиперболического тангенса. e_{1avg}^T - транспонированная матрица e_{1avg}. Аналогичным образом вычисляется вектор внимания контекста относительно второй сущности:

$$\alpha_{2i} = \frac{exp(\gamma(h_c^i, e_{2avg}))}{\sum_{j=1}^{n} exp(\gamma(h_c^j, e_{2avg}))}$$

$$\gamma(h_c^i, e_{2avg}) = tanh(h_c^i \cdot W_a \cdot e_{1avg}^T + b_a)$$

На основе полученных векторов внимания для слов вычисляется векторное представление контекста c_r относительно сущностей:

$$c_{e_1} = \sum_{i=1}^{n} \alpha_{1i} h_c^i$$

$$c_{e_2} = \sum_{i=1}^{m} \alpha_{2i} h^i$$

Полученные вектора представления контекста c_{e1} и c_{e2} конкатенируются в один вектор d для дальнейшей классификации в линейном слое с функцией активации $softmax$. Исходный код модели доступен в открытом репозитории[1].

3.3 Наборы данных

Эксперименты по оценке эффективности методов извлечения отношений проводились на четырех англоязычных корпусах: MADE [164] и i2b2 [131], состоящих из текстов электронных карточек пациентов, и на корпусах CDR [124] и PHAEDRA [178], состоящих из аннотаций научных статей медицинской тематики, и на размеченном русскоязычном корпусе текстов клинических карточек

[1] https://bitbucket.org/Ilseyar/relation-extraction/src/master/

Таблица 15 — Общая статистика по корпусам с размеченными отношениями.

Корпус	Кол-во отношений			Среднее расстояние (в токенах)			Максимальное расстояние (в токенах)		
	Обуч.	Оцен.	Всего	Обуч.	Оцен.	Всего	Обуч.	Оцен.	Всего
MADE	23036	4109	27145	30.6	26.0	29.9	981	868	981
CDR	3001	1512	4513	13.8	15.8	14.8	394	349	394
PHAEDRA	888	248	1136	13.5	16.4	15.0	137	262	262
i2b2	3120	6293	9413	3.7	3.7	3.7	73	61	73

[148]. Общая статистика корпусов представлена в таблице 15. Статистика включает в себя: количество отношений, среднее и максимальное значение длины контекста между сущностями (в токенах).

Согласно статистике только в корпусе i2b2 кол-во отношений в тестовой выборке превосходит кол-во отношений в обучающей выборке. Для остальных корпусов доля отношений в тестовой выборке варьируется от 15% (MADE) до 34% (CDR). Корпус MADE включает в себя наибольшее количество отношений (27145), в то время как корпус PHAEDRA содержит наименьшее количество отношений (1136). Наибольшее расстояние между сущностями в корпусе MADE (29.9 - в среднем и 981 - максимум), в то время как в корпусе i2b2 наименьшее расстояние между сущностями (3.7-в среднем и 73 - максимум).

Корпус **MADE** состоит из текстов электронных карточек 21 пациента, больного раком [164]. Электронные карточки включают discharge summaries, consultation reports, and other clinic notes. Общее количество документов в корпусе – 1 089, при этом 876 документов были выделены для обучающей выборки, а оставшиеся 213 для тестовой. Корпус размечали несколько аннотаторов: биологи, лингвисты и кураторы биомедицинских баз данных. Каждый документ размечался двумя аннотаторами, один из которых размечал исходный текст, а второй проверял, правил и предоставлял финальную разметку. Согласованность между пятью аннотаторами, подсчитанная на трех документах – 0.424, что попадает в диапазон приемлемого значения метрики согласия.

Каждый документ аннотирован следующими типами сущностей: drug (лекарство), adverse drug reaction или ADR (побочный эффект), indication (показания к применению), dose (доза), frequency (частота приема), duration (продолжительность приема), route (курс), severity (степень тяжести) и SSLIF (другие симптомы и заболевания). В корпусе размечено 7 типов отношений: drug—ADR (adverse), sslif – severity (severity), drug – route (route), drug – dose

(do), drug – duration (du), drug—frequency (fr), drug – indication (reason). Общая статистика корпуса представлена в таблице 16. Из таблицы 16 видно, что больше всего отношений типа: drug—dose, drug—indication и drug—frequency. Два типа отношений (reason и adverse) имеют максимальное кол-во расстояние между сущностями (более 900 символов), что усложняет выявление отношений между данными типами сущностей.

Таблица 16 — Общая статистика корпуса MADE.

Тип отношения	**Кол-во отношений**			**Среднее расстояние**			**Максимальное расстояние**		
	Обуч.	Оцен.	Всего	Обуч.	Оцен.	Всего	Обуч.	Оцен.	Всего
do	5176	866	6042	8.4	7.7	8.3	215	143	215
reason	4523	870	5393	89.3	63.8	85.2	981	868	981
fr	4417	729	5146	17.7	18.6	17.8	201	178	201
severity	3475	557	4032	2.6	1.8	2.5	259	188	259
adverse	1989	481	2470	59.4	45.6	56.7	937	718	937
route	2550	455	3005	13.5	12.9	13.4	191	137	191
du	906	147	1053	18.5	15.0	18.0	272	121	272
Всего	23 036	4109	27 145	30.6	26.0	29.9	981	868	981

Корпус **CDR**, разработанный в рамках соревнования BioCreative V [124], состоит из аннотаций научных статей, собранный с ресурса PubMed. В тексте выделены сущности, обозначающие заболевания (Disease) и химические вещество (Chemical), и отношения между этими сущностями. Корпус разбит на три подвыборки – обучающую, тестовую и валидационную. Каждая подвыборка состоит из 500 документов. В данной работе обучающая и валидационная выборки объединены в одну общую выборку, оценка модели проводится на оценочной части корпуса. Общая статистика представлена в таблице 15 (строка CDR).

Корпус **PHAEDRA** состоит из 597 аннотаций научных статей, собранных с ресурса MEDLINE [178]. В корпусе размечены три вида сущностей: лекарства, показания к применению и медицинские субъекты, которые связаны с заболеванием. Корпус содержит три вида бинарных отношений между сущностями:

– Subject_Disorder - отношение между медицинскими субъектами, обозначающими пациентов, и заболеваниями;

– is_equivalent - связывает разные названия одного и того же понятия (полное название и аббревиатуры или сокращения, название бренда и общее название для лекарства);
– Coreference - связывает сущности, размеченные аннотаторами, с неспецифическими общими выражениями, которые нельзя выделить в качестве сущности, например: это лекарство, такие расстройства и т.д.

В разметке корпуса принимали участие два аннотатора. Метрика согласованности для каждого типа отношений: Subject_Disorder - 69.3%, is_equivalent - 80.4%, Coreference - 50.91%.

Общая статистика корпуса отношений представлена в таблице 17. Согласно статистике больше всего в корпусе отношений с типом Subject_Disorder - 55% от общего числа отношений в корпусе. В типе отношений Coreference расстояние между сущностями существенно больше (39.7% в среднем), чем в остальных двух типах: Subject_Disorder (3.7%) и is_equivalent (1.6%).

Таблица 17 — Общая статистика корпуса PHAEDRA.

	Кол-во отношений			**Среднее расстояние**			**Максимальное расстояние**		
Тип отношения	Обуч.	Оцен.	Всего	Обуч.	Оцен.	Всего	Обуч.	Оцен.	Всего
Subject_Disorder	493	130	623	3.7	3.6	3.7	42	22	42
is_equivalent	229	67	296	1.1	2.1	1.6	15	23	23
Coreference	166	51	217	35.7	43.6	39.7	137	262	262
Всего	888	248	1136	13.5	16.4	15	137	262	262

Корпус **i2b2** был разработан в рамках соревнования i2b2/VA challenge в 2010-м году [131]. Корпус включает 871 размеченных документов, содержащих отчеты о выписках и истории болезней. В корпусе выделены три типа сущностей:

– Medical problem - медицинская проблема, включающая в себя название заболевания и его симптомы;
– Treatment - назначенное лечение;
– Test - медицинские исследования, проводимые для диагностирования заболевания, например, анализы, кардиограмма, рентген и т.д.

Между сущностями размечено три вида отношений:

– Medical problem—treatment (медицинская проблема - лечение) включает случаи, в которых лечение улучшило, ухудшило или вызвало медицин-

скую проблему, а также случаи, когда лечение было назначено или не назначено в связи с возникшей медицинской проблемой;

– Medical problem—test (медицинская проблема - тест) включает упоминания, где тесты были назначены или будут проведены для диагностирования медицинской проблемы;
– Medical problem—medical problem (медицинская проблема - медицинская проблема) включает медицинские проблемы, которые описывают или раскрывают аспекты другой медицинской проблемы или вызывают другие медицинские проблемы.

Согласно статистике, представленной в таблице 18, отношения типа Medical problem—treatment и Medical problem—test преобладают по количеству в корпусе. В отношении типа Medical problem—test самый большой по количеству токенов контекст (73), в то время как у типа отношений Medical problem—medical problem - самый короткий максимальный контекст между сущностями. В среднем в типе отношений Medical problem—treatment минимальный по длине контекст (2.8), в то время как в типе отношений Medical problem—test - максимальный (4.8).

Таблица 18 — Общая статистика корпуса i2b2.

Тип отношения	**Кол-во отношений**			**Среднее расстояние**			**Максимальное расстояние**		
	Обуч.	Оцен.	Всего	Обуч.	Оцен.	Всего	Обуч.	Оцен.	Всего
Medical problem—treatment	1206	2447	3653	2.7	2.8	2.8	58	61	61
Medical problem—test	1159	2398	3557	4.5	4.8	4.7	73	46	73
Medical problem—medical problem	755	1448	2203	3.9	3.4	3.7	33	48	48
Всего	3120	6293	9413	3.7	3.7	3.7	73	61	73

Корпус клинических текстов на русском языке (**RuClinical**) состоит из историй болезней пациентов больных аллергией и легочными заболеваниями. В корпусе содержится 112 размеченных документов, включающих отчеты по выписке, радиологии, эхокардиографии и ультразвуковой диагностики, рекомендации и другие записи, связанные с историей болезни. Схема разметки содержит следующие аннотации: Disease (название заболевания), Symptom (симптом), Drug (название лекарства), Treatment (лечение), Body location

(часть тела), “Severity” (тяжесть течения болезни), Course (стадия болезни). В текстах выделены следующие типы отношений между сущностями:

- Severity - тяжесть заболевания, отношение между сущностями Severity и Disease;
- Course - описывает стадию заболевания (обострение,), отношение между сущностями Course и Disease;
- Body_location - часть тела с которым связано заболевание, отношение между сущностями Body_location и Disease;
- Symptom_bdyloc - часть тела в которым проявляются симптомы, отношение между сущностями Body_location и Symptom.

В таблице 19 представлена общая статистика отношений в корпусе. Поскольку исходный корпус предоставляется без разделения на обучающие и тестовые данные, в таблице представлена общая статистика по корпусу. Количество отношений типа Symptom_bdyloc существенно больше, чем остальных типов. В отношения типа Course сущности, в среднем наиболее удалены друг от друга, чем в остальных типах отношений. Сущности в типе Severity находятся ближе всего друг к другу, максимальное расстояние в данном типе отношений всего 4 токена.

Таблица 19 — Общая статистика корпуса RuClinical.

Тип отношения	**Кол-во отношений**	**Среднее расстояние**	**Максимальное расстояние**
Severity	202	0.6	4
Course	145	2.7	11
Body_location	144	0.4	16
Symptom_bdyloc	1110	1.9	38
Всего	1601	1.4	38

3.4 Оценка эффективности модели LSTM+CA

В данном параграфе описаны базовые модели с которыми сравнивалась модель LSTM+CA+CA, выбранные параметры моделей, предобработка корпусов, постановка экспериментов, полученные результаты.

3.4.1 Базовые модели

В качестве базовых моделей было рассмотрено две архитектуры нейронных сетей: сверточная сеть с механизмом внимания (Attention-based convolutional neural network; CNNA) [179], BERT [77]. Данные модели показывают наилучшие результаты в задаче извлечения отношений. Далее приведено подробное описание архитектур базовых моделей.

Сеть **CNNA** состоит из двух частей: первая часть формирует векторное представление контекста с помощью сети CNN, вторая часть извлекает признаки на основе слоя внимания [179]. Конкатенация полученных векторов подается в дальнейшем на вход слою с классификацией. На вход CNN слою подаются векторные представления слов, позиционные признаки и части речи. Позиционные признаки формировались аналогичным образом, как и в сети LSTM+CA. Поскольку признаки на основе частей речи не дали прироста при тестировании модели, данные признаки были исключены. Вектор весов внимания подсчитывался следующим образом. Пусть каждое предложение содержит T слов, w_{it}, где $t \in [1, T]$, представляет вектор слова в i-м предложении, $e_i j$, где $j \in [1, 2]$ представляет j-ю сущность в i-м предложении. Далее вектора представления слов w_{it} и сущностей e_{ij} конкатенировались для получения нового представления слов в предложении $h_i^j t = [w_{it}, e_{ij}]$. После этого вычисляется значение u_{it}^j степень релевантности каждого слова в предложении по отношению к j-й сущности в i-м предложении по формулам:

$$h_i^{tj} = [w_{it}, eij]$$

$$u_j^{it} = W_a[\tanh(W_w e)h_i^{tj} + b_{we})] + b_a$$

После этого вычисляется нормализованный вектор весов α_{it}^j с помощью функции softmax:

$$\alpha_j^{it} = \frac{\exp u_j^{it}}{\sum_t \exp u_j^{it}}$$

На основе полученных значений весов внимания вычисляется финальное представление предложения:

$$s_{ij} = \sum_t \alpha_j^{it} w_{it}$$

BERT [77] - это контекстно-зависимая модель представления слов, основанная на двунаправленной многослойной архитектуре нейронной сети - Трансформер (Vaswani et al., 2017) [172]. Модель BERT предобучалась на большом корпусе статей Википедии и текстах книг. Для корпусов на английском языке применялась модель BioBERT [169], для корпуса на русском применялась модель RuBERT [169]. Модель **BioBERT** (Bidirectional Encoder Representations from Transformers for Biomedical Text Mining) обучалась дополнительно на текстах аннотаций научных статей с ресурсов PubMed и PMC [169]. BioBERT показала улучшение результатов по сравнению с предыдущими современными моделями в задаче извлечения биомедицинских отношений (улучшение метрики F1 на 2,80%) [169]. В данной работе использовалась реализация модели из репозитория [2], в частности запускалась задача run_re. Модель RuBERT была обучена на русских текстах Википедии и текстах новостей. В качестве инициализации начальных весов использовалась модель MultiBERT (BERT-base, Multilingual Cased), которая была предварительно обучена на Википедии на 104 языках и имеет 12 слоев множественного внимания, и в общей сложности содержит 110 миллионов параметров.

3.4.2 Параметры моделей

Входной текст для моделей CNNA и LSTM+CA кодировался векторным представлением использовалась модель векторного представления слов BioWordVec, обученная на текстах статей PubMed и клинических записях базы данных MIMIC-III Clinical Database [180]. Длина векторов 200. Статистика покрываемости корпусов словами из модели векторного представления слов: MADE - 35%, CDR - , PHAEDRA - 89%, i2b2 - 81%. Для слов, отсутствующих в модели, генерируется вектор случайных чисел с нормальным распределением и значениями, ранжирующимися в рамках значений векторов модели векторного представления слов.

Модель LSTM+CA запускалась со следующими параметрами: количество эпох - 10, шаг обучения (learning rate) - 0.001, оптимизатор - Adam, исходная инициализация весов сети - Ксавье (Xavier uniform), размер блока данных (batch) -

[2] https://github.com/dmis-lab/biobert

128, размер скрытого слоя LSTM - 300. В модели CNNA размер сверточного слоя равнялся 100, размер ядра равнялся 5, остальные параметры идентичны модели LSTM+CA. В качестве модели BioBERT использовалась версия BioBERT v1.0 (+ PubMed 200K + PMC 270K). Обе модели BioBERT и RuBERT имеют 12 слоев, 768 скрытых нейронов в каждом слое и суммарно 110 миллионов параметров. Модели обучались на 10 эпохах с размером блока данных 32.

3.4.3 Генерация отрицательных примеров

Исходная разметка корпусов содержит только положительные примеры пар сущностей, между которыми есть отношения. Соответственно, возникает необходимость на основе выделенных в корпусе сущностей сгенерировать пары сущностей между которыми отношения отсутствуют. Однако перебор всех возможных вариантов пар сущностей в рамках одного документа порождает слишком большое количество отрицательных примеров, что приводит к большому дисбалансу при классификации и существенно снижает ее качество. В связи с этим в работах по извлечению отношений используется ряд правил, чтобы отфильтровать наименее вероятные пары сущностей.

Генерация отрицательных примеров для корпусов i2b2, CDR и PHAEDRA осуществлялась согласно правилам, предложенным Гу и др. [181]. Для сущностей из одного предложения применялись следующие правила: (i) расстояние в токенах между сущностями менее 10 токенов; (ii) если в предложении несколько сущностей, которые могут находиться в отношении с заданной сущностью, то выбирается ближайшая. Отрицательные примеры для сущностей из разных предложений генерировались согласно следующим правилам: (i) рассматриваются только те сущности, которые не участвуют ни в одном отношении в рамках одного предложения; (ii) расстояние между сущностями в предложениях должно быть менее 3 (iii) если имеется несколько сущностей, которые могут находиться в отношении с заданной сущностью, то выбирается ближайшая.

Поскольку в среднем расстояние между сущностями в корпусе MADE больше, чем в остальных корпусах, генерация отрицательных примеров для данного корпуса осуществлялась по другим правилам [145]: количество символов между сущностями меньше 1000, а количество других сущностей, которые

могут участвовать в отношениях и находятся между сущностями-кандидатами, не превышает 3.

3.4.4 Оценка моделей в рамках одного корпуса

Таблица 20 — Результаты для задачи извлечения отношений в рамках одного корпуса.

Корпус	**Модель**	**No-Related**			**Related**			**Среднее**		
		P	R	F	P	R	F	P	R	F
CDR	LSTM+CA	**.850**	.628	.722	.503	**.772**	.609	.676	.700	.666
	CNNA	.798	.718	.756	.521	.628	.570	.660	.674	.663
	BioBERT	.839	**.812**	**.825**	**.637**	.679	**.657**	**.738**	**.745**	**.741**
PHAEDRA	LSTM+CA	.971	.962	.966	.727	.782	.753	.849	.872	.860
	CNNA	.948	.973	.961	.742	.593	.659	.845	.783	.810
	BioBERT	**.978**	**.985**	**.982**	**.884**	**.827**	**.854**	**.931**	**.906**	**.918**
i2b2	LSTM+CA	.912	.852	.881	.664	.779	.716	.788	.816	.799
	CNNA	.888	.873	.880	.675	.704	.688	.782	.788	.784
	BioBERT	**.919**	**.889**	**.904**	**.726**	**.787**	**.755**	**.822**	**.838**	**.829**
MADE	LSTM+CA	.981	.978	.979	.847	.862	.854	.914	.920	.917
	CNNA	.973	.982	.978	.863	.806	.834	.918	.894	.906
	BioBERT	**.988**	**.989**	**.989**	**.923**	**.916**	**.920**	**.956**	**.953**	**.954**
RuClinical	LSTM+CA	**.979**	**.986**	**.983**	.704	**.683**	**.689**	**.842**	**.843**	**.841**
	CNNA	.737	.820	.766	.613	.421	.436	.675	.621	.601
	RuBERT	.817	.904	.855	**.765**	.614	.669	.791	.759	.762

Результаты оценки моделей в рамках одного корпуса представлены в таблице 21. В данной работе оценка моделей осуществлялась согласно рекомендациям, представленным в работе Пысало с соавторами[85]. Наибольшие результаты по средней F-мере на всех англоязычных корпусах показала модель BioBERT. Наиболее высокие результаты были получены на корпусе MADE (95.4%). LSTM+CA превзошла результаты модели CNNA на всех корпусах. Наибольший прирост модели LSTM+CA по отношению к CNNA среди англоязычных корпусов был достигнут на корпусе PHAEDRA (+5%). На корпусе

ClinicalRu модель LSTM+CA показала самые высокие результаты F-меры (84.1%), превзойдя модель BioBERT на 7.9% и модель CNNA на 24%. Таким образом, полученные результаты показывают эффективность разработанной модели для русского языка.

3.4.5 Кросс-доменная оценка моделей

Таблица 21 — Результаты кросс-доменной оценки моделей для задачи извлечения отношений по метрике средней F-меры. Результаты в рамках одного домена на диагонали. Среднее - это усредненное значение F-меры для всех кросс-доменных экспериментов. Средняя потеря - это разница между результатами в рамках одного домена и среднего значения.

Target→ Source↓	**Модель**	**CDR**	**PHAEDRA**	**i2b2**	**MADE**	**Среднее**	**Средняя потеря**
CDR	LSTM+CA	.666	**.520**	.514	**.590**	**.541**	-.125
	BioBERT	**.741**	.519	**.574**	.516	.536	-.205
PHAEDRA	LSTM+CA	**.467**	.860	**.509**	**.471**	**.482**	-.378
	BioBERT	.399	**.918**	.477	.465	.447	-.471
i2b2	LSTM+CA	.579	**.628**	.799	**.541**	**.583**	-.216
	BioBERT	**.583**	.593	**.829**	.523	.566	-.263
MADE	LSTM+CA	.525	**.468**	**.606**	.917	**.533**	-.374
	BioBERT	**.579**	.446	.556	**.954**	.527	-.427

В таблице 21 представлены результаты кросс-доменной оценки моделей в задаче извлечения отношений. В качестве базовой модели для сравнения была выбрана модель BioBERT, поскольку данная модель показала наиболее высокие результаты при оценке моделей в рамках одного корпуса. Согласно полученным результатам модели показывают существенно ниже качество в случаях, когда обучающий и тестовый набор данных из разных доменов, чем когда оценка проводится в рамках одного корпуса. Падение показателя макро F-меры варьируется от 12.5% до 47.1%. Уменьшение результатов при кросс-доменной оценке может быть обусловлено различием отношений между корпусами. В таблице 22 представлена статистика одинаковых отношений в корпусах. Отношение

считалось одинаковым, если обе сущности из отношения совпадали. Согласно статистике, максимальное пересечение наблюдается между корпусами i2b2 и MADE (2.4%). Таким образом, отношения в корпусах почти не повторяются, что приводит к снижению результатов. Стоит отметить, что при обучении моделей на корпусе PHAEDRA наблюдается наибольшая потеря в качестве при кросс-доменной оценке: -37.8% для модели LSTM+CA и - 47.1% для модели BioBERT. Такой результат может быть обусловлен меньшим количеством отношений в корпусе PHAEDRA по сравнению с остальными корпусами (см. таблицу 15), а также отсутствием пересечений отношений корпуса PHAEDRA с другими корпусами.

Таблица 22 — Процент пересечения уникальных слов корпусов. Жирным шрифтом выделены максимальные значения.

Корпус	**CDR**	**PHAEDRA**	**i2b2**	**MADE**
CDR	100	0	0.69	1
PHAEDRA	0	100	0	0
i2b2	0.69	0	100	2.4
MADE	1	0	2.4	100

Разработанная модель LSTM+CA превосходит модель BioBERT по среднему значению F-меры на всех корпусах. Наибольший прирост модели LSTM+CA по сравнению с BioBERT в среднем был достигнут при обучении модели на корпусе PHAEDRA (+3.5%), наименьший при обучении на корпусе CDR (+0.5%). В трех экспериментах модель BioBERT превзошла результаты LSTM+CA: CDR-i2b2 (-6%), i2b2-CDR (-0.4%), MADE-CDR (-5.4%). Максимальный прирост модели LSTM+CA по сравнению с BioBERT был достигнут в эксперименте CDR-MADE (+7.4%). Наименьший прирост в эксперименте CDR-PHAEDRA (+0.01%).

Полученные результаты показывают, что модель LSTM+CA более эффективна в случае, когда обучающие и тестовые данные из разных доменов, что в свою очередь доказывает большую возможность практического применения модели LSTM+CA по сравнению с BioBERT.

3.4.6 Оценка модели на комбинации корпусов

В данном наборе экспериментов модель LSTM+CA обучалась на совокупности обучающих корпусов и последовательно оценивалась на оценочной части каждого корпуса. Результаты представлены в виде гистограммы на рисунке 3.2.

Рисунок 3.2 — Результаты средней F-меры модели LSTM+CA, обученной на одном корпусе (LSTM+CA) и модели, обученной на совокупности англоязычных корпусов (LSTM+CA (all)) для задачи извлечения отношений.

Результаты показывают, что обучение на совокупности корпусов не дает прироста в результатах. Отсутствие увеличения результатов может быть обусловлено двумя факторами. Во-первых, корпуса содержат разные типы отношений (см. таблицу 22). Во-вторых, анализ представления контекстов показывает, что контексты корпусов не пересекаются друг с другом (см. параграф 3.5). Наименьшая потеря в результатах наблюдается на корпусе i2b2 (-3%), наибольшая на корпусе PHAEDRA (-23.1%). Данный результат обусловлен тем, что корпус отношений из корпуса PHAEDRA уникальны и не пересекаются с отношениями других корпусов. Потеря средней F-меры на корпусах CDR и MADE составила -7.5% и -5.4%.

3.5 Анализ представлений контекста

Ключевую роль в извлечении отношений играет контекст, в котором встречаются сущности. В зависимости от контекста две одинаковые сущности могут быть связаны между собой или нет. Например, в отрывке рецепта, выписанного пациенту: “Лоразепам 1 мг каждые 6 часов по необходимости при тошноте, омепразол 20 мг в день” тошнота является показанием к применению лекарственного препарата Лоразепам, следовательно, сущности Лоразепам и тошнота связаны между собой. В то же время, в предложении: “Прохлорперазин 10 мг каждые 6 часов при тошноте, Валацикловир 500 мг 2 раза в день, Лоразепам 1 мг при бессоннице” сущности Лоразепам и тошнота уже не связаны между собой. Для более детального анализа результатов были исследованы различные методы представления контекста для задачи извлечения отношений в рамках одного корпуса и для кросс-доменного случая. В качестве контекста рассматриваются токены между сущностями, которые участвуют в отношении.

3.5.1 Анализ представления контекста в рамках одного корпуса

В рамках одного корпуса были оценены различные методы представления контекста и выявлению наиболее эффективные для задачи извлечения отношений между биомедицинскими сущностями. В рамках исследований были рассмотрены следующие методы представления контекста:

- Мешок слов (bag of words, bow) - одна из самых первых моделей представления текста, подсчитывающая количество вхождений каждого слова из словаря, где словарь является набором уникальных слов всех текстов обучающей выборки. Модель не учитывает порядок слов в тексте, что является одним из ее главных недостатков, кроме того, итоговый вектор, представляющий текст имеет большую размерность.
- Усредненное векторное представление слов (word2vec) вычисляется суммированием векторного представления каждого слова контекста, разделенного на общее количество слов в контексте. Преимуществом данной модели является возможность учитывать семантический смысл

слов, то есть вектора слов, близких по смыслам, будут близки между собой в векторном пространстве. Кроме того, такое представление имеет фиксированную размерность для всех текстов, равной длине вектора слова. Однако, при усреднении векторов данное свойство может быть утеряно на уровне текста.

- Векторное представление предложения (sent2vec) одна из вариаций метода векторного представления слов [29]. Однако, в отличии от нее, нейронная сеть учится не только на отдельных словах, но еще и на словосочетаниях и усредненном значении векторов для предложения. Таким образом, модель может лучше отразить семантический смысл предложения, чем простое усреднение векторов слов.
- Сверточная нейронная сеть (CNN), состоящая из трех сверточных слоев с размерами фильтров 3, 4 и 5. Выходы сверточных слоев проходили через слой с функцией max pooling, конкатенировались и подавались в полносвязный слой.
- Рекуррентная нейронная сеть (RNN), в данном эксперименте использовалась сеть с долгой краткосрочной памятью LSTM с размером скрытого слоя 200.
- Векторное представление контекста, полученное из модели BioBERT. В качестве вектора использовался вектор токена [CLS] в котором сохраняется информация о всех токенах текста, поданного модели на вход.

Таблица 23 — Результаты классификатора для различных моделей представления контекста.

	MADE			**CDR**		
Method	**P**	**R**	**F**	**P**	**R**	**F**
bow	.878	.573	.693	.395	.341	.367
word2vec	.760	.800	.779	**.557**	.312	.400
sent2vec	.894	.873	.883	.437	.376	.405
CNN	.725	.825	.772	.446	.334	.382
RNN	.482	.404	.440	.297	**.516**	.377
BioBERT	**.929**	**.882**	**.905**	.473	.385	**.424**

Эксперименты были построены следующим образом: на вход классификатору последовательно подавался закодированный различными способами

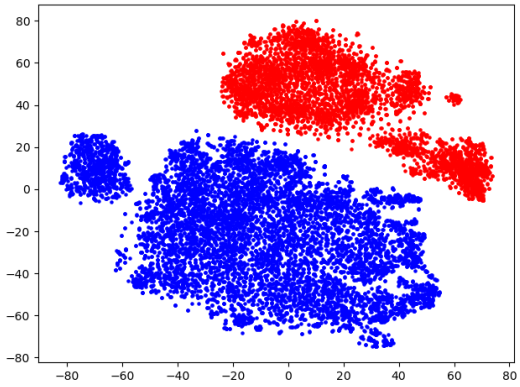
контекст. В качестве классификатора был взят метод случайный лес (RF). Результаты представлены в таблице 23. Сверточная нейронная сети обучались со следующими параметрами: количество эпох - 10, размер батча - 32, вес позитивного класса ‘related’ - 0.7, вес негативного класса ‘non-related’ - 0.3. Для обучения рекуррентной нейронной сети использовались следующие параметры: размер дропаута - 0.2, количество эпох - 20, размер батча - 64, вес позитивного класса ‘related’ - 0.75, вес негативного класса ‘non-related’ - 0.25.

Согласно полученным результатам, все модели превзошли результаты базовой модели на основе мешка слов, которая показала 69.3% и 36.7% F-меры на корпусах MADE и CDR соответственно. Наилучшим способом представления контекста для обоих корпусов является векторное представление на основе модели BioBERT. Данная модель показала 90.5% и 42.4% F-меры на корпусах и CDR соответственно. На втором месте оказалась модель sent2vec. При этом разница с моделью BioBERT для корпуса CDR составила 1.9%, а на корпусе MADE - 2.2%. остальные модели показали результаты существенно ниже. Среди подходов, основанных на нейронных сетях более высокие результаты, показала сверточная нейронная сеть, она обошла рекуррентную на 33.2% и 0.05% F-меры на корпусах MADE и CDR соответственно. По метрикам точности и полноты наиболее высокие результаты для корпуса CDR показали метод на основе усредненного векторного представления (55.7% точности) и рекуррентная нейронная сеть (51.6% полноты). На корпусе MADE наиболее высокие результаты точности и полноты достиг метод на основе BioBERT (92.9% и 88.2%, соответственно).

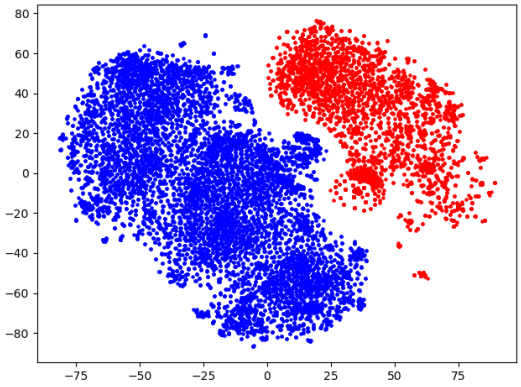
Полученные результаты позволяют сделать вывод, что в рамках одного корпуса модель BioBERT генерирует более качественное представление контекста, что позволяет модели показывать более высокие результаты.

3.5.2 Кросс-доменный анализ представления контекста

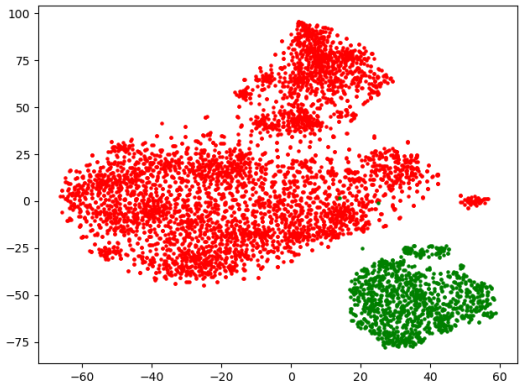
Для более детального анализа результатов была проведена оценка близости векторного представления контекста между сущностями для моделей LSTM+CA и BioBERT. Векторное представление контекста для модели LSTM+CA было получено из выходных значений слоя LSTM, который принимает на вход контекст. Для модели BioBERT векторное представление контекста



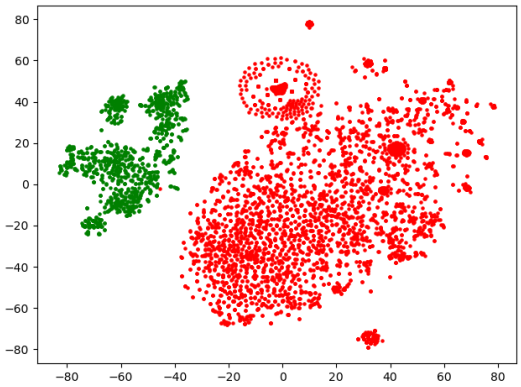
CDR-i2b2 (LSTM+CA)



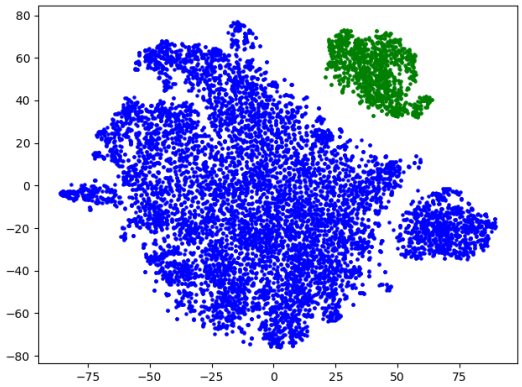
CDR-i2b2 (BioBERT)



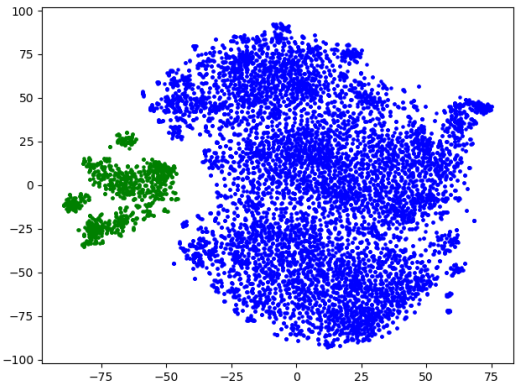
CDR- (LSTM+CA)



CDR-PHAEDRA (BioBERT)



PHAEDRA-i2b2 (LSTM+CA)



PHAEDRA-i2b2 (BioBERT)

Рисунок 3.3 — Векторное представление контекстов между сущностями для корпусов CDR (красный), PHAEDRA (зеленый) и i2b2 (синий).

было получено как усреднение скрытых состояний последних четырех слоев модели. Полученные вектора для обеих моделей были нормализованы. Для визуализации вектора были сжаты в двумерное пространство с помощью алгоритма t-SNE [182].

На рисунке 3.3 представлены результаты визуализации представлений контекстов. Результаты показывают, что контексты корпусов между собой не пересекаются, что объясняет потерю в результатах при кросс-доменном тести-

ровании моделей. При этом у модели LSTM+CA контексты из одного корпуса более уплотненные в сравнении с моделью BioBERT. Было также подсчитано расстояние Евклида между центрами кластеров. Для модели LSTM+CA были получены следующие результаты: PHAEDRA-i2b2 - 4.25, PHAEDRA-CDR - 5.96 и CDR - i2b2 - 5.02. Для модели BioBERT: PHAEDRA - i2b2 - 7.19, PHAEDRA-CDR - 6.29 и CDR - i2b2 - 6.59. Согласно подсчитанным метрикам расстояния между кластерами, LSTM+CA генерирует более близкие вектора для разных корпусов по сравнению с BioBERT.

Для оценки зависимости между полученными результатами и векторным представлением контекста был подсчитан коэффициент корреляции Спирмена между метрикой F-меры и расстоянием между контекстами корпусов. Для модели LSTM+CA коэффициент корреляции Спирмена равен -0.717, а для модели BioBERT коэффициент равен -0.517, что доказывает обратную зависимость между F-мерой и расстоянием: чем больше F-мера, тем ближе расстояние между корпусами и наоборот. При этом у модели LSTM+CA эта зависимость более сильная, поскольку коэффициент ближе к -1. Это обусловливает более высокие результаты модели LSTM+CA по сравнению с BioBERT.

3.6 Модель LSTM+CA+feat

В данной секции описана модель LSTM+CA+feat - модификация модели LSTM+CA с набором признаков, извлеченных из текста. Для разработки модели LSTM+CA+feat на первом этапе были проведены эксперименты по выявлению наиболее эффективных признаков. Полученные на первом этапе признаки были добавлены в последний слой сети LSTM+CA. Далее проводились эксперименты по оценке эффективности модели LSTM+CA+feat.

3.6.1 Описание признаков

Признаки были разделены на 4 категории, на основе: расстояния, слов, векторного представления и знаний. Признаки на основе расстояния подсчиты-

вают различные метрики расстояния между сущностями. Признаки на основе слов были получены с использованием различных свойств слов. Признаки на основе векторного представления получены из моделей, предобученных на больших корпусах биомедицинских текстов. Признаки на основе знаний извлекались из текстов различных биомедицинских ресурсов и биомедицинских тезаурусов. Далее представлен список всех признаков с подробным описанием.

Признаки на основе расстояния:

- расстояние в словах (word_dist): количество слов между сущностями;
- расстояние в символах (char_dist): количество символов между сущностями;
- расстояние в предложениях (sent_dist): количество предложений между сущностями;
- расстояние в знаках препинания (punct_dist): количество знаков препинания между сущностями;
- Позиционный признак (position): позиция сущности-кандидата (сущности типа SSLIF или DRUG) по отношению к сущности-атрибуту (все остальные сущности), где позиция сущности-атрибута равна 0.

Признаки на основе слов:

- мешок слов (bow): все слова в окне размера 10 до и после сущностей вместе с текстом самих сущностей. Были использованы только те слова, которые встречаются больше 500 раз во всем корпусе;
- мешок сущностей (boe): количество всех типов сущностей между рассматриваемыми сущностями;
- типы сущностей (type): бинарный вектор длины всех типов сущностей в корпусе с цифрой 1 на позиции типов сущностей, которые встречаются между целевыми сущностями.

Признаки на основе векторного представления:

- Векторное представление сущностей (ent_emb): векторное представление сущностей, полученное из предобученной модели векторного представления слов. Для сущностей, состоящих из нескольких слов, вектора усреднялись. Были исследованы две модели векторного представления слов: модель, обученная на текстах из Википедии, аннотациях научных статей с ресурса PubMed и PMC, а также модель BioWordVec, обученная на текстах аннотаций статей с ресурса PubMed и клинических записей с ресурса MIMIC-III;

– Векторное представление контекста (cont_emb): векторное представление контекста, полученный из предобученной модели BioSentVec. Модель BioSentVec была получена с помощью библиотеки sent2vec и состоит из векторов для предложений длины 700;
– Схожесть сущностей (sim): мера близости между векторными представлениями сущностей. Были использованы 4 меры близости: taxicab, косинусная, Евклидова и покоординатная (coordinate).

Признаки на основе знаний:

– концепты UMLS (umls): бинарный вектор семантических концептов каждой сущности, полученных из тезауруса UMLS (Unified Medical Language System);
– концепты MeSH (mesh): бинарный вектор семантических концептов каждой сущности, полученных из тезауруса MeSH (Medical Subject Headings);
– совстречаемость в текстах FDA: количество документов, описывающих клинические испытания, полученных с ресурса FDA, в которых присутствуют обе сущности;
– совстречаемость в биомедицинских текстах: статистика совстречаемости сущностей в различных документах биомедицинской тематики. Детальное описание источников текстов представлено ниже.

В экспериментах были рассмотрены три ресурса: (i) научные статьи с ресурса MEDLINE, (ii) патенты с USPTO, (iii) projects from the grant-making Agencies of USA, Canada, EU, and Australia. Система *Pharmacognitive* позволяет получить следующую статистику по перечисленным ресурсам: количество документов за каждый год или кол-во средств, выделенных на исследование, за год для терминов, указанных в запросе. Запросы были сгенерированы с использованием сущностей следующих типов: лекарство, показание к применению и побочный эффект. Запросы были расширены синонимами, которые представляет система Pharmacognitive. Были использованы следующие типы признаки для запросов:

– количество публикаций, патентов, проектов, опубликованных за указанный год (3 признака для каждого года с 1952 г. по 2018 г.);
– количество публикаций, патентов, проектов, опубликованных за предшествующий указанному год (3 признака для каждого года с 1953 г. по 2018 г.);

– общее количество публикаций, патентов, проектов, опубликованных за все время (3 признака);
– среднее значение и сумма количества проектов, опубликованных за указанный год (2 признака для каждого года с 1974 г. по 2018 г.);
– среднее значение и сумма количества проектов, опубликованных за предшествующий указанному год (2 признака для каждого года с 1974 г. по 2018 г.);
– среднее значение и сумма количества проектов, опубликованных за все время (2 признака).

Были также получены признаки на основе статистики публикаций и проектов для объединенного запроса, включающего типы сущностей: Лекарство и сущность, относящаяся к лекарству (Побочный эффект или показание к применению).

3.6.2 Оценка информативности признаков

Оценка информативности признаков проводилась на корпусе MADE. В качестве классификатора использовалась модель Случайный лес (Random Forest; RF) с количеством деревьев, равным 100 и весами 0.7 для позитивного и 0.3 для негативного классов. Исходный код модели выложен в открытый репозиторий [3]. В качестве базового подхода использовался классификатор с признаками на основе расстояния и слов.

Результаты экспериментов модели RF с различными наборами признаков представлены в таблице 25. Согласно результатам, классификатор с только дистанционными признаками достигает 76.6% F-меры. Признаки на основе расстояния слов, символов и знаков препинания дополняют друг друга, поскольку отсутствие одного из них приводит к примерно одинаковой потере результатов на обоих корпусах. Признак на основе расстояния предложений не дал улучшения результатов. Позиционный признак улучшил результаты (-0.8%). Базовая модель без набора признаков расстояния (см. строки «word» в таблице XX) снижает результаты микро F-меры на 19%, что свидетельствует о важности этих признаков для классификации отношений.

[3] https://github.com/Ilseyar/relation-extraction-ehr

Таблица 24 — Результаты F-меры для каждого типа отношений и усредненной F-меры для вскх типов отношений корпуса MADE. Признаки на основе расстояния и слов рассматривались как базовая модель (baseline).

Features	**severity**	**route**	**reason**	**do**	**du**	**fr**	**adverse**	**all**
baseline: distance & word feat-s	.933	.918	.806	.906	.905	.896	.729	.866
baseline-word_dist	.923	.922	.812	.900	.860	.909	.716	.864
baseline-char_dist	.929	.916	.810	.908	.869	.890	.731	.864
baseline-sent_dist	.933	.919	.807	.910	.880	.906	.719	.866
baseline-punc_dist	.926	.912	.798	.907	.836	.906	.735	.863
baseline-position	.931	.917	.803	.897	.865	.883	.723	.858
distance	.918	.843	.683	.859	.713	.780	.525	.766
baseline-boe	.932	.897	.775	.888	.861	.868	.715	.845
baseline-bow	.918	.906	.726	.895	.810	.843	.712	.828
baseline-type	.934	.906	.779	.899	.891	.891	.562	.839
word	.542	.777	.645	.662	.718	.846	.511	.672
baseline+emb_pubmed_pmc_wiki	.927	.898	.730	.887	.684	.900	.605	.827
baseline+emb_bio	.920	.903	.772	.893	.602	.908	.613	.833
baseline+concept_emb	.920	.897	.764	.902	**.910**	.889	.610	.841
baseline+sent_emb	.936	**.954**	**.937**	**.929**	.854	**.938**	**.869**	**.926**
sent_emb	.932	.935	.909	.915	.854	.835	.782	.884
baseline+sim	.920	.908	.796	.905	.880	.902	.737	.862
baseline+umls	.936	.915	.815	.922	.883	.891	.734	.870
baseline+mesh	**.938**	.918	.812	.910	.856	.904	.730	.868
baseline+fda	.936	.912	.808	.906	.895	.909	.730	.868
baseline+bio_text	.934	.918	.805	.906	.905	.896	.749	.866
baseline+knowledge	.936	.914	.806	.916	.889	.896	.736	.848

Признаки на основе слов также улучшили качество классификации системы извлечения отношений. Наибольшее улучшение микро-F-меры получено с помощью мешка слов (+3.8% для MADE), что можно объяснить большим размером вектора по сравнению с остальными признаками. Признак на основе мешка сущностей также увеличил результаты базовых моделей на 2.1% (см. строку «baseline-boe»). Признак на основе типов сущностей также улучшил результаты микро F-меры (+ 2.7%).

Результаты для признаков на основе векторного представления сущностей показывают, что вектора и меры сходства уменьшают результаты независимо от используемой модели векторного представления слов. Модель с признаком на

основе векторного представления предложений достигла наиболее значительного улучшения исходных результатов и показала 92.6% микро F-меры на корпусе MADE. Более того, модель, обученная только с признаком на основе векторного представления предложения превзошла базовую на 1.8%. Для оценки признаков на основе знаний лучше рассмотреть результаты для разных типов отношений отдельно. Добавление признака, основанного на UMLS, к базовой модели увеличило результаты базовой модели для типов: severity (0.3%), reason (0.9%), dose (1.6%) и adverse (0.5%). Сочетание базовых признаков и признака UMLS показало наилучшие результаты среди признаков на основе знаний. На MADE корпусе дополнение семантических типов MeSH увеличило результаты базовой модели для наибольшего числа типов отношений, включая severity (+0.3%), reason (+0.6%), dose (+0.4%), frequency (+0.8%) и adverse (+0.1%) типов. Добавление признака совстречаемости в текстах FDA позволило достичь наиболее значительного увеличения F-меры для типа frequency (1.3%) по сравнению с базовой моделью. Признак совстречаемости в биомедицинских текстах улучшил качество классификатора для типа adverse на 2% F-меры. Таким образом, все признаки на основе знаний как отдельно, так и в сочетании, повысили результаты для типов severity и adverse. Признаки на основе знаний не увеличили результаты для типов route и duration. Признак MeSH оказался наиболее эффективным.

3.6.3 Архитектура модели LSTM+CA+feat

Общая архитектура модели LSTM+CA+feat, представленная на рисунке 3.4, основана на нейронной сети LSTM+CA с набором дополнительных признаков, которые конкатенируются с финальным полносвязным слоем сети. В качестве признаков были использованы наиболее информативные признаки для задачи извлечения отношений: признаки на основе расстояния и слов. Векторное представление контекста также дает прирост при извлечении отношений, однако, эксперименты показали, что добавление данного признака в модель LSTM+CA понижает результаты модели, поэтому было решено не добавлять его в модель.

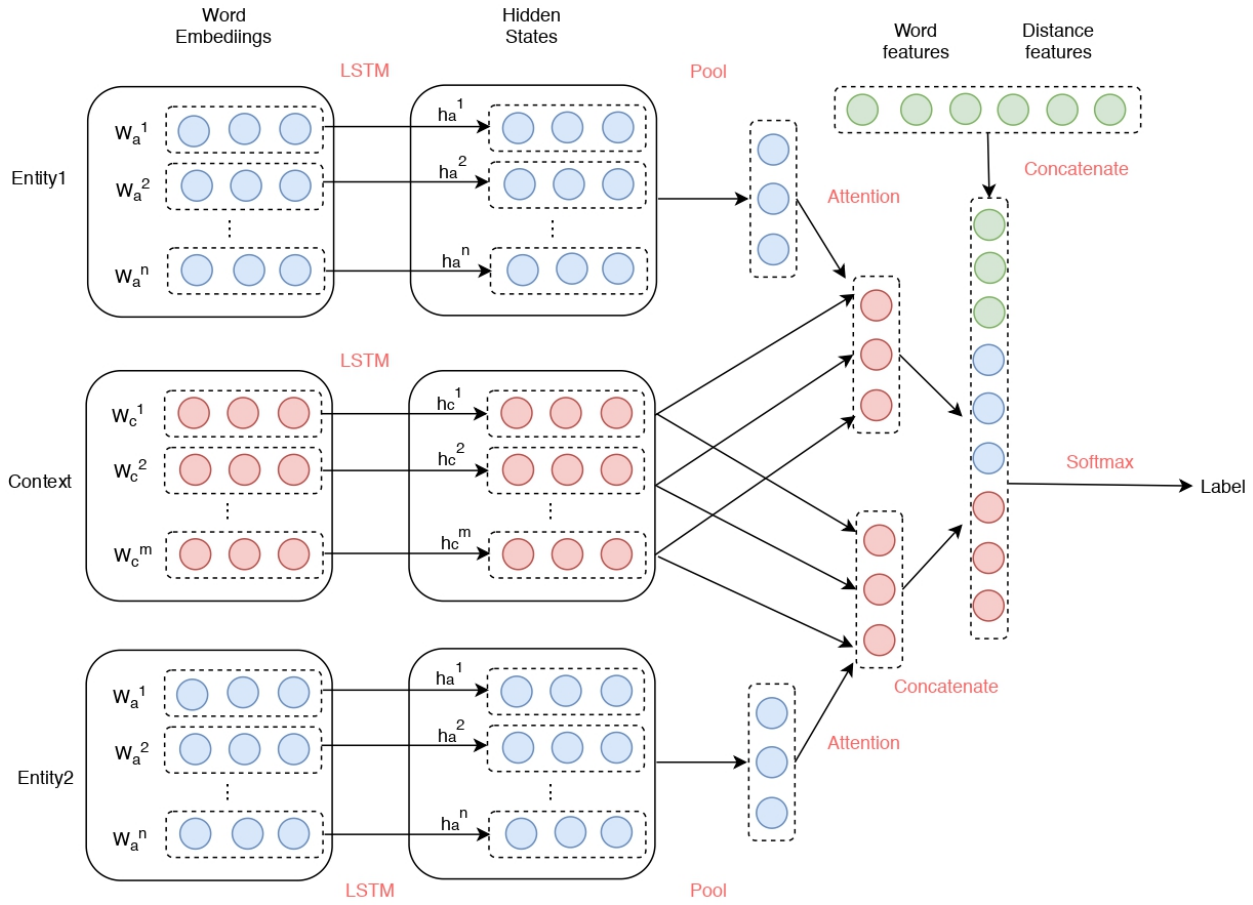


Рисунок 3.4 — общая архитектура модели LSTM+CA для извлечения отношений

3.6.4 Оценка модели LSTM+CA+feat

Таблица 25 — Результаты модели LSTM+CA+feat для задачи извлечения отношений.

Корпус	Модель	No-Related			Related			Среднее		
		P	R	F	P	R	F	P	R	F
MADE	LSTM+CA	.981	.978	.979	.847	.862	.854	.914	.920	.917
	LSTM+CA+feat	**.987**	**.987**	**.985**	**.865**	**.875**	**.881**	**.926**	**.931**	**.933**
PHAEDRA	LSTM+CA	.971	.962	.966	.727	.782	.753	.849	.872	.860
	LSTM+CA+feat	**.973**	**.967**	**.971**	**.731**	**.785**	**.761**	**.852**	**.876**	**.866**

Результаты модели LSTM+CA+feat представлены в таблице 25. Согласно полученным результатам добавление признаков в модель позволило увеличить результаты средней F-меры на корпусах MADE и PHAEDRA на 1.6% и 0.6%, соответственно. Более низкие результаты на корпусе PHAEDRA обусловлены

тем, что подбор наиболее информативных признаков осуществлялся на корпусе MADE. Остальные метрики также показали прирост после добавления признаков. Максимальный прирост был достигнут для метрики F-меры класса Related на обоих корпусах MADE (+3.5%) и PHAEDRA (+0.8%). Полученные результаты показывают эффективность добавления признаков в модель LSTM+CA.

3.7 Выводы к третьей главе

В данной главе была рассмотрена задача извлечения отношений между сущностями из текстов на русском и английском языках. Для решения поставленной задачи была разработана модель LSTM+CA, основанная на слое LSTM с механизмом внимания. Проведены обширные эксперименты для количественной оценки эффективности разработанной модели на русскоязычном корпусе клинических текстов и четырех англоязычных корпусах биомедицинской тематики. Эксперименты в рамках одного корпуса показали преимущество разработанной модели по сравнению с базовой моделью CNNA на англоязычных корпусах. На русскоязычном корпусе модель LSTM+CA показала самые высокие результаты по средней F-мере и превзошла базовые модели CNNA и BioBERT. Кроме того, модель LSTM+CA превзошла результаты базовой модели BioBERT при кросс-доменной оценке на всех англоязычных корпусах. В дополнение были выявлены наиболее информативные признаки для задачи извлечения отношений. Признаки были имплементированы в модель LSTM+CA+feat, были проведены эксперименты, показывающие улучшение результатов по сравнению с результатами модели LSTM+CA. Таким образом, проведенные эксперименты показали преимущество разработанной модели по сравнению с базовыми подходами, основанными на современных моделях нейронных сетей.

Глава 4. Архитектура программных комплексов для классификации сущностей и извлечения отношений

В данной главе представлено описание архитектур разработанных программных комплексов CAFEC (Cross-Attention Feature-based Entity Classifier) [1] для классификации сущностей и CARE (Cross-Attention Relation Extraction)[2] для извлечения отношений.

4.1 Общая архитектура программных комплексов

Программные комплексы предоставляют функционал для обучения и оценки моделей нейронных сетей с механизмом кросс-внимания LSTM+CA для задач классификации сущностей и извлечения отношений. Программные комплексы написаны на языке Python 3.6 и предоставлены в открытом доступе.

Общая архитектура программных комплексов CAFEC и CARE представлена на рисунке 4.1. Программные комплексы состоят из следующих общих модулей:

- модуль слоев нейронных сетей;
- модуль моделей;
- модуль признаков;
- модуль загрузки данных;
- модуль запуска обучения и тестирования.

Модуль слоев содержит реализацию слоев нейронных сетей: DynamicRNN и Attention. Класс DynamicRNN предоставляет реализацию рекурентных слоев: LSTM, GRU и RNN. Такая реализация позволяет легко переключаться между типа сетей и упрощает проведение экспериментов. Класс принимает на вход следующие параметры:

- input_size - размер входных векторов;
- hidden_size - размер скрытого слоя;
- num_layers - количество слоев;

[1] https://bitbucket.org/Ilseyar/entity_classification/src/master
[2] https://bitbucket.org/Ilseyar/relation-extraction/src/master/

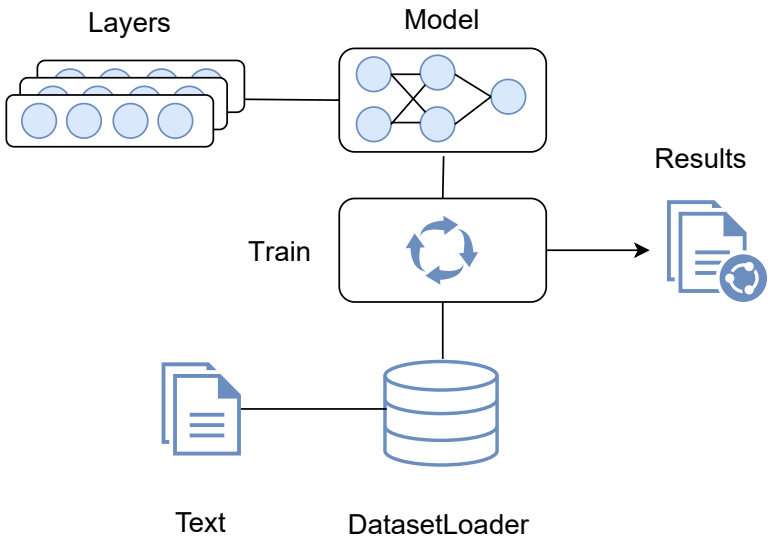


Рисунок 4.1 — Общая архитектура программных комплексов CAFEC и CARE

– bias - необходимо ли использовать коэффициенты смещения;
– batch_first - объединять ли входные и выходные значения в батчи;
– dropout - процент нейронов для отключения (дропаута);
– bidirectional - двунаправленная или однонаправленная сеть;
– rnn_type - тип рекуррентной сети.

Класс Attention реализует механизм внимания и принимает на вход следующие параметры:

– embed_dim - размер входных векторов;
– hidden_dim - размер скрытого слоя;
– n_head - количество голов внимания;
– score_function - функция вычисления внимания;
– dropout - процент нейронов для отключения (дропаута).

Модуль моделей содержит реализацию архитектур нейронных сетей. Каждый модуль, реализующий нейронную сеть, принимает на вход матрицу векторных представлений слов *'embedding_matrix'* и набор параметров, зависящий от реализации конкретной сети и описанный более подробно в секциях **??**.

Модуль загрузки данных включат в себя все необходимые функции и классы для загрузки и обработки данных. Функция *'load_word_vec'* загружает модель векторного представления слов и возвращает словарь, где для каждого слова в соответствие поставлен вектор. Функция *build_embedding_matrix'* на основе полученного словаря строит матрицу весов для слоя векторного представления слов нейронной сети. Модуль *Tokenizer* разделяет входные тексты на токены, составляет словарь корпуса, который включает в себя все уникаль-

ные токены корпуса, а также кодирует исходный текст, заменяя все токены на индексы согласно словарю. В процессе кодирования унифицируется длинна предложения путем добавления в конец более коротких предложений служебного индекса. В модуле *DatasetLoader* реализован общий конвеер загрузки данных. На вход модуль принимает файл с исходными текстами, осуществляет препроцессинг данных соответственно задаче (см. секции **??**) и возвращает структуру данных словарь с необходимыми для работы нейронной сети данными. Данный модуль наследован от класса *Dataset* библиотеки PyTorch и предоставляет дополнительный функционал для вычисления общей длины набора данных и получения одного конкретного экземпляра.

Модуль запуска обучения и тестирования реализует основные этапы выполнения программы. Модуль принимает на вход следующие параметры:

- model_name - название модели
- dataset - название набора данных
- optimizer - оптимизатор для обучения сети
- initializer - алгоритм инициализации стартовых параметров слоев нейронной сети
- learning_rate - размер шага оптимизатора при обучении модели
- dropout - процент нейронов для отключения (дропаута).
- l2reg -
- num_epoch - количество эпох для обучения сети
- batch_size - количество примеров, подаваемых единовременно на вход модели
- log_step - частота логирования данных
- logdir - директория для записи логов
- embed_dim - размер векторов в используемой модели векторного представления слов
- embed_file - путь до файла с предобученной моделью векторного представления слов
- hidden_dim - размер скрытого состояния слоя LSTM
- max_seq_len - максимальный размер входного контекста в токенах
- polarities_dim - количество классов
- device - устройство на котором проводить вычисления: графическая карта (gpu) или процессор (cpu)
- fold_num - количество фолдов

На первом этапе происходит инициализация всех необходимых параметров для обучения нейронной сети: модели, оптимизатора, метода регуляризации и т.д. Алгоритм обучения сети реализован в классе Instructor. При инициализации класса Instructor происходит загрузка и обработка данных. Далее запускается метод train, в котором реализованы циклы по эпохам и батчам. На каждой итерации происходит обновление параметров сети, подсчет функции потерь и обновление градиента. После выхода из цикла по эпохам модель делает предсказания на тестовом множестве, производится оценка метрик для полученных ответов, итоговые результаты записываются в файл.

4.2 Вспомогательные библиотеки

Для реализации программных комплексов использовались следующие библиотеки: PyTorch (версия 1.1.0), NumPy (версия 1.13.3), TensoboardX (версия 1.2), Scikit-learn (версия 0.20.0).

PyTorch - фреймворк для машинного обучения с открытым исходным кодом на языке Python, который упрощает программную реализацию моделей машинного обучения, создание прототипов для исследования и разработку моделей для производственных задач. Данная библиотека разработана на основе низкоуровневой библиотеки Torch. Pytorch разработан группой искусственного интеллекта компании Facebook. Фреймворк предоставляет функционал для тензорных вычислений, а также реализацию базовых слоев нейронных сетей.

NumPy - библиотека с открытым исходным кодом, включающая в себя функционал для реализации и выполения различных математических функций с многомерными массивами. Помимо этого библиотека включает в себя генераторы случайных чисел, процедуры линейной алгебры, преобразования Фурье и многое другое. Ядро NumPy реализовано на языке C, что позволяет осуществлять вычисления с более высокой скоростью, поскольку в процессе используется скомпилированный код.

TensorboardX - библиотека для визуализации и отслеживания процессов обучения нейронной сети. В библиотеке реализован модуль SummaryWriter, который предоставляет высокоуровневый инструментарий для создания файла событий в заданном каталоге и добавления в него отчеты о процессе обучения

нейронной сети. Класс обновляет содержимое файла асинхронно, что позволяет программе вызывать методы для добавления данных в файл непосредственно из цикла обучения, не замедляя процесс обучения.

Scikit-learn - это открытая библиотека для реализации классических алгоритмов машинного обучения. Библиотека включает различные алгоритмы классификации, регрессии и кластеризации, включая метод опорных векторов, случайные леса, градиентный бустинг и т.д. В дополнение библиотека содержит функции для оценки качества разработанных моделей обучения. В данной работе библиотека использовалась для вычисления метрик качества классификации: полнота, точность и F-мера, а также для реализации извлечения из текстов признаков.

4.3 Особенности реализации программного комплекса CAFEC

На вход программа принимает данные в следующем формате: в первой строке контекст, в котором сущность, относительно которой производится классификация заменена служебным символом T, на второй строке текст сущности и на третьей класс в виде числового значения 0 или 1. Пример входных данных представлен на рисунке 4.2.

```
So far its been very good , pains almost gone , but I $T$ , did n't have that when on 50 .
feel a bit weird
1
Due to my arthritis getting progressively worse , to the point where I am in tears with the $T$.
agony
0
```

Рисунок 4.2 — Формат входных данных для программного комплекса CAFEC.

В дополнение к основным модулям в программном комплексе CAFEC содержится модуль признаков, в котором реализовано извлечение признаков для модели LSTM+CA+feat. В данном модуле содердится пять классов, реализующих соответсвующие признаки: BrownClustersFeature, PMIFeature, PosTagFeatures, SentimentFeature, UMLSemanticTypeFeature, а также два вспомогательных класса: ExtractEntities и ExtractWindowWords, возвращающие тексты сущностей и рассматриваемых контекстов из исходных данных. Все классы наследованы от базового BaseEstimator из библиотеки Scikit-learn. В

каждом классе реализован метод *transform*, в котором вычисляются векторные значения признаков. Для работы данного модуля в программный комплекс были добавлены вспомогательные ресурсы: список токенов с соответствующими номерами кластеров Брауна, значениями PMI, концептами UMLS, а также список слов, создающих негативный контекст. В модуле загрузки данных дополнительно реализован класс FeatureDatasetLoader, который наследуется от DatasetLoader. В классе FeatureDatasetLoader формируется вектор признаков для входного текста. В данном классе также реализован признак мешок слов.

В модуле моделей имеется два класса с реализациями архитектур нейронных сетей LSTM+CA и LSTM+CA+feat. Основным отличием реализации сетей является размер последнего слоя, в LSTM+CA+feat он увеличен на длину входного вектора дополнительных признаков, а также на вход данной модели подается дополнительный вектор признаков.

4.4 Особенности реализации программного комплекса CARE

На вход программа принимает данные в следующем формате: в первой строке контекст, в котором сущности, относительно которых предсказывается наличие отношения заменены служебными символами T, на второй строке тексты сущностей, разделенные символом & и на третьей класс в виде числового значения 0 или 1. Пример входных данных представлен на рисунке 4.3.

```
1  $T$ reversed abnormal bone formation , such as woven osteoid and $T$
2  OCT & fibrosis
3  0
4  $T$ does not completely prevent the occurrence of $T$
5  OCT & hypercalcemia
6  1
```

Рисунок 4.3 — Формат входных данных для программного программного комплекса CARE.

В модуле моделей содержится один класс с реализацией сети LSTM+CA для извлечения отношений. В отличие от реализации нейронной сети LSTM+CA для классификации сущностей в этом модуле на вход доволнительно необходимо подать вектор второй сущности.

Таблица 26 — Характеристики сервера для проведения экспериментов.

Параметр	Описание
Процессор	Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60GHz
Оперативная память	251 Гб Intel
Видеокарта	2 x 24 Гб Tesla P40
Версия чипсета	NVIDIA Corporation GP102 (Pascal)
Версия драйвера видеокарты	410.48
ОС	3.10.0-693.el7.x86_64 GNU/Linux
Версия Docker	18.09.4

4.5 Оценка производительности

Для оценки производительности разработанной системы будут рассмотрены две характеристики: память и время. В качестве оценки по памяти измерялось количество занятого на видеокарте объема. Временная оценка вычислялась по следующим критериям: время, затраченное на обучение одной эпохе; общее время обучения модели; общее количество эпох, которое потребовалось для обучения модели. Время, затраченное на оценку качества модели, не учитывалось. Обучение моделей проводилось в контейнере докера, развернутом на сервере с характеристиками, представленными в таблице 26.

Оценки по памяти и по времени на каждом корпусе для программных комплексов CAFEC и CARE представлены в таблицах 27 и 28, соответственно. Временая оценка напрямую зависит от количества обучающих примеров. Самое долго обучение в программных комплексах CAFEC и CARE было на корпусе MADE (48:20 и 58:00, соответственно). Средняя скорость обучения в программном комплексе CAFEC, примерно, 500 примеров в секунду. В архитектуре нейронной сети LSTM+CA в программном комплексе CARE имеется дополнительный слой с механизмом внимания, поэтому средняя скорость обучения 400 примеров в секунду меньше, чем в CAFEC. Объем занимаемой памяти на видеокарте напрямую зависит от длинны контекста. Загрузка примеров обучения реализована по батчам, но для ускорения оценки нейронной сети в процессе обучения тестовый набор данных загружается целиком на видеокарту. В таблицах 27 и 28 приведены объемы памяти, занимаемых в процессе обучения без учета тестовых данных. Наибольший объем памяти в комплексе CAFEC

Таблица 27 — Оценка времени обучения модели LSTM+CA+feat в программном комплексе CAFEC. Количество эпох: 10, общее количество параметров: 1926602. Формат времени: mm:ss.

Корпус	**Кол-во обучающих примеров**	**Макс. длинна контекста**	**Объем памяти на видеокарте (MiB)**	**Время 1 эпоха**	**Время всего**
CADEC	5056	236	763	00:12	02:00
MADE	30852	173	727	04:50	48:20
PsyTAR	5523	264	775	00:16	02:40
Twimed-PubMed	996	150	701	00:02	00:20
TwiMed-Twitter	472	42	677	00:01	00:10
Twitter	512	37	679	00:01	00:10
RuDReC	1262	165	711	00:02	00:20

Таблица 28 — Оценка времени обучения модели LSTM+CA в программном комплексе CARE. Количество эпох: 10, общее количество параметров: 3249602. Формат времени: mm:ss.

Корпус	**Кол-во обучающих примеров**	**Макс. длинна контекста**	**Объем памяти на видеокарте (MiB)**	**Время 1 эпоха**	**Время всего**
CDR	8593	394	1059	00:34	05:40
PHAEDRA	8150	262	801	00:21	03:30
i2b2	83382	73	745	02:21	23:30
MADE	205604	981	3075	05:48	58:00
RuClinical	48180	48	526	00:26	04:20

занимает корпус PsyTAR (775 MiB). В комплексе CARE максимальный объем памяти на видеокарте занимает корпус MADE (3075 MiB).

4.6 Выводы к четвертой главе

В данной главе приведено подробное описание программных комплексов CAFEC для классификации сущностей и CARE для извлечения отношений. Рассмотрены подробные реализации модулей, показаны форматы входных данных,

команды для запуска программ, использованные в процессе разработки библиотеки. Приводится оценка скорости обучения и необходимого объема памяти моделей нейронных сетей на каждом корпусе из экспериментов, описанных в главах 2 и 3. Согласно приведенным оценкам, время, затрачиваемое на обучение для обеих задач, не превосходит 60 минут, что позволяет обучать сети для новых корпусов в довольно сжатые сроки и является преимуществом разработанных программных комплексов по сравнению с существующими языковыми моделями.

Заключение

Основные результаты работы заключаются в следующем.

1. Предложен и реализован новый метод классификации сущностей, основанный на нейронной сети с механизмом кросс-внимания и набором информативных признаков.
2. Предложен и реализован новый метод извлечения отношений между сущностями, основанный на нейронной сети с механизмом кросс-внимания и с разделением контекста и сущностей на отдельные подсети формирования контекстных векторных представлений.
3. Разработано программное обеспечение CAFEC и проведено экспериментальное исследование, обосновывающее улучшение качества предложенных методов по сравнению с существующими алгоритмами в рамках корпусов из одного домена и корпусов из разных доменов для задачи классификации сущностей.
4. Разработано программное обеспечение CARE и проведено экспериментальное исследование, обосновывающее улучшение качества предложенных методов по сравнению с существующими алгоритмами в рамках текстовых корпусов из разных доменов для задачи извлечения отношений между сущностями.

Дальнейшие перспективы развития исследований могут быть связаны с (i) проведением мультиязычных экспериментов, в которых языки обучающего и оценочного корпуса различаются; (ii) применением подходов удаленного обучения (distant supervision); (iii) разработкой модели с применением многозадачного обучения (multi-task learning).

Благодарности

В заключение хотелось бы выразить благодарность моему научному руководителю Тутубалиной Елене, которая консультировала и направляла меня в ходе исследования и оказывала колоссальную поддержку на протяжении всего процесса подготовки диссертации. Я также признательна институту Информационных Технологий и Интеллектуальных Систем КФУ, благодаря которому эта работа стала возможной, всей команде НИЛ “Хемоинформатика и Молекулярное Моделирование” КФУ и, в частности, ее руководителю Маджидову Тимуру за тёплую и дружескую рабочую атмосферу. Выражаю свою признательность Соловьеву Валерию Дмитриевичу за советы и ценные замечания при работе над данной диссертацией. Выражаю благодарность Санкт-Петербургскому отделению Математического института им. В. А. Стеклова РАН за предоставленные вычислительные мощности. И в первую очередь я благодарна своей семье и друзьям и главным образом моей маме Алимовой Ризиде Закариявне, которая поддерживала меня на протяжении всего пути. Работа выполнена при финансовой поддержке грантов РНФ №18-11-00284 и РФФИ №19-07-01115.

Список сокращений и условных обозначений

ЭМК электронные медицинские карточки пациентов.

SVM support vector machine, метод опорных векторов.

CRF conditional random fields, метод условных случайных полей.

RF random forest, метод случайных лесов.

UMLS Unified Medical Language System, система, содержащая унифицированный язык медицинских терминов.

CNN convolutional neural network, сверточная нейронная сеть.

CRNN convolutional recurrent neural network, сверточная рекуррентная нейронная сеть.

RCNN recurrent convolutional neural network, рекуррентная сверточная нейронная сеть.

CNNA attention-based convolutional neural network, сверточная нейронная сеть с вниманием.

LSTM long short term memory, сеть с долгой краткосрочной памятью.

biLSTM bidirectional long short term memory, двунаправленная сеть с долгой краткосрочной памятью.

PMI pointwise mutual information, поточечная взаимная информация.

P precision, точность, вычисляется как отношение истинно-положительных релевантных документов к общему количеству определенных системой положительных документов.

R recall, полнота, вычисляется как отношение истинно-положительных документов к общему количеству известных положительных документов.

F-мера метрика качества классификации, вычисляется как среднее гармоническое между точностью и полнотой.

LSTM+CA cross attention long short term memory, сеть с долгой краткосрочной памятью и кросс-вниманием.

LSTM+CA+feat cross attention long short term memory with features, сеть с долгой краткосрочной памятью, кросс-вниманием и признаками.

ADR adverse drug reaction, побочный эффект.

softmax обобщение логистической функции для многомерного случая, вычисляется по формуле:

$$softmax(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}},$$

где $x_1, ...x_n$ - заданный вектор.

CNN-BiLSTM convolutional neural network with bidirectional long short term memory, сверточная нейронная сеть с двунаправленной сетью с долгой краткосрочной памятью.

BERT Bidirectional Encoder Representations from Transformers, двунаправленное представление кодировщика трансформера.

BioBERT Bidirectional Encoder Representations from Transformers for Biomedical Text Mining двунаправленное представление кодировщика трансформера для биомедицинских текстов.

TD_LSTM target-dependent long short term memory, зависимая от цели сеть с долгой краткосрочной памятью.

MemNet Deep Memory Network, сеть с глубокой памятью.

RAM Recurrent Attention Memory, сеть с рекуррентным механизмом внимания и памятью.

GRU Gated Recurrent Unit, управляемый рекуррентный блок.

Список литературы

1. Utilizing social media data for pharmacovigilance: a review / A. Sarker [и др.] // Journal of biomedical informatics. — 2015. — т. 54. — с. 202—212.
2. Detecting adverse events for patient safety research: a review of current methodologies / H. J. Murff [и др.] // Journal of biomedical informatics. — 2003. — т. 36, № 1/2. — с. 131—143.
3. Adverse drug reaction identification and extraction in social media: a scoping review / J. Lardon [и др.] // Journal of medical Internet research. — 2015. — т. 17, № 7.
4. Text mining for adverse drug events: the promise, challenges, and state of the art / R. Harpaz [и др.] // Drug safety. — 2014. — т. 37, № 10. — с. 777—790.
5. *ONYE, S. C.* Review of Biomedical Relation Extraction / S. C. ONYE, A. AKKELEŞ, N. DIMILILER // European International Journal of Science and Technology. — 2017. — т. 6, № 1.
6. *Bui, Q.* Relation extraction methods for biomedical literature. т. 27 / Q. Bui. — 2011.
7. *Alshuwaier, F.* A comparative study of the current technologies and approaches of relation extraction in biomedical literature using text mining / F. Alshuwaier, A. Areshey, J. Poon // Engineering Technologies and Applied Sciences (ICETAS), 2017 4th IEEE International Conference on. — IEEE. 2017. — с. 1—13.
8. Overview of the First Natural Language Processing Challenge for Extracting Medication, Indication, and Adverse Drug Events from Electronic Health Record Notes (MADE 1.0) / A. Jagannatha [и др.] // Drug safety. — 2018. — с. 1—13.
9. *Ramponi, A.* Cross-Domain Evaluation of Edge Detection for Biomedical Event Extraction / A. Ramponi, B. Plank, R. Lombardo // Proceedings of The 12th Language Resources and Evaluation Conference. — 2020. — с. 1982—1989.

10. *Алимова, И. С.* Сравнительный анализ нейронных сетей в задаче классификации побочных эффектов на уровне сущностей в англоязычных текстах / И. С. Алимова, Е. В. Тутубалина // Труды Института системного программирования РАН. — 2018. — т. 30, № 5.

11. *Alimova, I.* Automated detection of adverse drug reactions from social media posts with machine learning / I. Alimova, E. Tutubalina // International Conference on Analysis of Images, Social Networks and Texts. — Springer. 2017. — P. 3—15.

12. A Machine Learning Approach to Classification of Drug Reviews in Russian / I. Alimova [и др.] // Ivannikov ISPRAS Open Conference (ISPRAS), 2017. — IEEE. 2017. — с. 64—69.

13. *Alimova, I.* A Comparative Study on Feature Selection in Relation Extraction from Electronic Health Records. / I. Alimova, E. Tutubalina // DAMDID/RCDL. — 2019. — с. 34—45.

14. *Alimova, I.* Detecting Adverse Drug Reactions from Biomedical Texts With Neural Networks / I. Alimova, E. Tutubalina // Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop. — 2019. — с. 415—421.

15. *Tutubalina, E.* Biomedical Entities Impact on Rating Prediction for Psychiatric Drugs / E. Tutubalina, I. Alimova, V. Solovyev // International Conference on Analysis of Images, Social Networks and Texts. — Springer. 2019. — с. 97—104.

16. *Alimova, I.* Multiple features for clinical relation extraction: A machine learning approach / I. Alimova, E. Tutubalina // Journal of Biomedical Informatics. — 2020. — т. 103. — с. 103382.

17. *Alimova, I.* Interactive Attention Network for Adverse Drug Reaction Classification / I. Alimova, V. Solovyev // Artificial Intelligence and Natural Language. — Springer. 2018. — с. 36—48.

18. Using semantic analysis of texts for the identification of drugs with similar therapeutic effects / E. Tutubalina [и др.] // Russian Chemical Bulletin. — 2017. — т. 66, № 11. — с. 2180—2189.

19. *Miftahutdinov, Z.* KFU NLP Team at SMM4H 2019 Tasks: Want to Extract Adverse Drugs Reactions from Tweets? BERT to The Rescue / Z. Miftahutdinov, I. Alimova, E. Tutubalina // Proceedings of the Fourth Social Media Mining for Health Applications (# SMM4H) Workshop & Shared Task. — 2019. — с. 52—57.

20. The Russian Drug Reaction Corpus and neural models for drug reactions and effectiveness detection in user reviews / E. Tutubalina [и др.] // Bioinformatics. — 2020. — июль. — URL: https://doi.org/10.1093/bioinformatics/btaa675.

21. *Nugmanov, R.* Detecting adverse drug reactions from user reviews with machine learning / R. Nugmanov, I. Alimova, E. Tutubalina // European Journal of Clinical Investigation. т. 48. — Wiley. 2018. — с. 217—218.

22. *Nugmanov, R.* Adverse drug reactions identification in social media posts and electronic health records with neural networks / R. Nugmanov, I. Alimova, E. Tutubalina // European Journal of Clinical Investigation. т. 49. — Wiley. 2019. — с. 116—117.

23. *Genkin, A.* Large-scale Bayesian logistic regression for text categorization / A. Genkin, D. D. Lewis, D. Madigan // Technometrics. — 2007. — т. 49, № 3. — с. 291—304.

24. Some effective techniques for naive bayes text classification / S.-B. Kim [и др.] // IEEE transactions on knowledge and data engineering. — 2006. — т. 18, № 11. — с. 1457—1466.

25. Text classification using string kernels / H. Lodhi [и др.] // Journal of Machine Learning Research. — 2002. — т. 2, Feb. — с. 419—444.

26. *Magerman, D. M.* Statistical Decision-Tree Models for Parsing / D. M. Magerman // 33rd Annual Meeting of the Association for Computational Linguistics. — 1995. — с. 276—283.

27. *Quinlan, J. R.* Induction of decision trees / J. R. Quinlan // Machine learning. — 1986. — т. 1, № 1. — с. 81—106.

28. *Ho, T. K.* Random decision forests / T. K. Ho // Proceedings of 3rd international conference on document analysis and recognition. т. 1. — IEEE. 1995. — с. 278—282.

29. Hdltex: Hierarchical deep learning for text classification / K. Kowsari [и др.] // 2017 16th IEEE international conference on machine learning and applications (ICMLA). — IEEE. 2017. — с. 364—371.

30. *Sutskever, I.* Generating text with recurrent neural networks / I. Sutskever, J. Martens, G. E. Hinton // ICML. — 2011.

31. *Mandic, D.* Recurrent neural networks for prediction: learning algorithms, architectures and stability / D. Mandic, J. Chambers. — Wiley, 2001.

32. *Hochreiter, S.* Long short-term memory / S. Hochreiter, J. Schmidhuber // Neural computation. — 1997. — т. 9, № 8. — с. 1735—1780.

33. On the properties of neural machine translation: Encoder-decoder approaches / K. Cho [и др.] // arXiv preprint arXiv:1409.1259. — 2014.

34. Reading text in the wild with convolutional neural networks / M. Jaderberg [и др.] // International journal of computer vision. — 2016. — т. 116, № 1. — с. 1—20.

35. Identifying potential adverse effects using the web: A new approach to medical hypothesis generation / A. Benton [и др.] // Journal of biomedical informatics. — 2011. — т. 44, № 6. — с. 989—996.

36. Social media mining for drug safety signal detection / C. C. Yang [и др.] // Proceedings of the 2012 international workshop on Smart health and wellbeing. — ACM. 2012. — с. 33—40.

37. *Liu, X.* AZDrugMiner: an information extraction system for mining patient-reported adverse drug events in online patient forums / X. Liu, H. Chen // International conference on smart health. — Springer. 2013. — с. 134—150.

38. A pipeline to extract drug-adverse event pairs from multiple data sources / S. Yeleswarapu [и др.] // BMC medical informatics and decision making. — 2014. — т. 14, № 1. — с. 13.

39. Digital drug safety surveillance: monitoring pharmaceutical products in twitter / C. C. Freifeld [и др.] // Drug safety. — 2014. — т. 37, № 5. — с. 343—350.

40. Pharmacovigilance on twitter? Mining tweets for adverse drug reactions / K. O'Connor [и др.] // AMIA annual symposium proceedings. т. 2014. — American Medical Informatics Association. 2014. — с. 924.

41. *Nikfarjam, A.* Pattern mining for extraction of mentions of adverse drug reactions from user comments / A. Nikfarjam, G. H. Gonzalez // AMIA Annual Symposium Proceedings. т. 2011. — American Medical Informatics Association. 2011. — с. 1019.

42. Sentiment classification of drug reviews using a rule-based linguistic approach / J.-C. Na [и др.] // International conference on asian digital libraries. — Springer. 2012. — с. 189—198.

43. Analysis of polarity information in medical text / Y. Niu [и др.] // AMIA annual symposium proceedings. т. 2005. — American Medical Informatics Association. 2005. — с. 570.

44. *Chee, B. W.* Predicting adverse drug events from personal health messages / B. W. Chee, R. Berlin, B. Schatz // AMIA Annual Symposium Proceedings. т. 2011. — American Medical Informatics Association. 2011. — с. 217.

45. *Bian, J.* Towards large-scale twitter mining for drug-related adverse events / J. Bian, U. Topaloglu, F. Yu // Proceedings of the 2012 international workshop on Smart health and wellbeing. — ACM. 2012. — с. 25—32.

46. *Yang, M.* Identification of Consumer Adverse Drug Reaction Messages on Social Media. / M. Yang, X. Wang, M. Y. Kiang // PACIS. — 2013. — с. 193.

47. *Sarker, A.* Portable automatic text classification for adverse drug reaction detection via multi-corpus training / A. Sarker, G. Gonzalez // Journal of biomedical informatics. — 2015. — т. 53. — с. 196—207.

48. Mining Twitter for adverse drug reaction mentions: a corpus and classification benchmark / R. Ginn [и др.] // Proceedings of the fourth workshop on building and evaluating resources for health and biomedical text processing. — Citeseer. 2014.

49. Mining adverse drug reaction signals from social media: going beyond extraction / A. Patki [и др.] // Proceedings of BioLinkSig. — 2014. — т. 2014. — с. 1—8.

50. Extraction of adverse drug effects from clinical records. / E. Aramaki [и др.] // MedInfo. — 2010. — с. 739—743.

51. Detecting signals in noisy data-can ensemble classifiers help identify adverse drug reaction in tweets / M. Rastegar-Mojarad [и др.] // Proceedings of the Social Media Mining Shared Task Workshop at the Pacific Symposium on Biocomputing. — 2016.

52. Adverse Drug Reaction Classification With Deep Neural Networks / T. Huynh [и др.] // Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. — 2016. — с. 877—887.

53. *Miranda*, *D. S.* Automated detection of adverse drug reactions in the biomedical literature using convolutional neural networks and biomedical word embeddings / D. S. Miranda // arXiv preprint arXiv:1804.09148. — 2018.

54. *Zhang*, *M.* Adverse Drug Event Detection Using a Weakly Supervised Convolutional Neural Network and Recurrent Neural Network Model / M. Zhang, G. Geng // Information. — 2019. — т. 10, № 9. — с. 276.

55. Detecting adverse drug reactions from social media based on multi-channel convolutional neural networks / C. Shen [и др.] // Neural Computing and Applications. — 2019. — т. 31, № 9. — с. 4799—4808.

56. *Jain*, *S.* Detecting Twitter posts with Adverse Drug Reactions using Convolutional Neural Networks. / S. Jain, X. Peng, B. C. Wallace // SMM4H@ AMIA. — 2017. — с. 72—75.

57. *Stanovsky*, *G.* Recognizing mentions of adverse drug reaction in social media using knowledge-infused recurrent models / G. Stanovsky, D. Gruhl, P. Mendes // Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers. — 2017. — с. 142—151.

58. *Tutubalina*, *E.* Combination of deep recurrent neural networks and conditional random fields for extracting adverse drug reactions from user reviews / E. Tutubalina, S. Nikolenko // Journal of healthcare engineering. — 2017. — т. 2017.

59. Bidirectional LSTM-CRF for adverse drug event tagging in electronic health records / S. Wunnava [и др.] // International Workshop on Medication and Adverse Drug Event Detection. — 2018. — с. 48—56.

60. Recognizing continuous and discontinuous adverse drug reaction mentions from social media using LSTM-CRF / B. Tang [и др.] // Wireless Communications and Mobile Computing. — 2018. — т. 2018.

61. Improving RNN with attention and embedding for adverse drug reactions / C. Pandey [и др.] // Proceedings of the 2017 International Conference on Digital Health. — ACM. 2017. — с. 67—71.

62. *Ramamoorthy, S.* An attentive sequence model for adverse drug event extraction from biomedical text / S. Ramamoorthy, S. Murugan // arXiv preprint arXiv:1801.00625. — 2018.

63. *Zhang, Z.* An ensemble method for binary classification of adverse drug reactions from social media / Z. Zhang, J. Nie, X. Zhang // Proceedings of the Social Media Mining Shared Task Workshop at the Pacific Symposium on Biocomputing. — 2016.

64. *Ofoghi, B.* Read-biomed-ss: Adverse drug reaction classification of microblogs using emotional and conceptual enrichment / B. Ofoghi, S. Siddiqui, K. Verspoor // Proceedings of the Social Media Mining Shared Task Workshop at the Pacific Symposium on Biocomputing. — 2016.

65. NRC-Canada at SMM4H Shared Task: Classifying Tweets Mentioning Adverse Drug Reactions and Medication Intake / S. Kiritchenko [и др.] // arXiv preprint arXiv:1805.04558. — 2018.

66. *Friedrichs, J.* InfyNLP at SMM4H Task 2: Stacked Ensemble of Shallow Convolutional Neural Networks for Identifying Personal Medication Intake from Twitter / J. Friedrichs, D. Mahata, S. Gupta // arXiv preprint arXiv:1803.07718. — 2018.

67. Detecting Tweets Mentioning Drug Name and Adverse Drug Reaction with Hierarchical Tweet Representation and Multi-Head Self-Attention / C. Wu [и др.] // Proceedings of the 2018 EMNLP Workshop SMM4H: The 3rd Social Media Mining for Health Applications Workshop & Shared Task. — 2018. — с. 34—37.

68. *Minard, A.-L.* IRISA at SMM4H 2018: Neural Network and Bagging for Tweet Classification / A.-L. Minard, C. Raymond, V. Claveau // Social Media Mining for Health Applications (SMM4H), Workshop of EMNLP. — 2018.

69. *Plachouras, V.* Quantifying self-reported adverse drug events on Twitter: signal and topic analysis / V. Plachouras, J. L. Leidner, A. G. Garrow // Proceedings of the 7th 2016 International Conference on Social Media & Society. — ACM. 2016. — с. 6.

70. *Sarker, A.* Social media mining shared task workshop / A. Sarker, A. Nikfarjam, G. Gonzalez // Biocomputing 2016: Proceedings of the Pacific Symposium. — World Scientific. 2016. — с. 581—592.

71. *Sarker, A.* Overview of the second social media mining for health (smm4h) shared tasks at amia 2017 / A. Sarker, G. Gonzalez-Hernandez // Training. — 2017. — т. 1, № 10, 822. — с. 1239.

72. Overview of the third social media mining for health (smm4h) shared tasks at emnlp 2018 / D. Weissenbacher [и др.] // Proceedings of the 2018 EMNLP Workshop SMM4H: The 3rd Social Media Mining for Health Applications Workshop & Shared Task. — 2018. — с. 13—16.

73. Overview of the fourth social media mining for health (SMM4H) shared tasks at ACL 2019 / D. Weissenbacher [и др.] // Proceedings of the Fourth Social Media Mining for Health Applications (# SMM4H) Workshop & Shared Task. — 2019. — с. 21—30.

74. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports / H. Gurulingappa [и др.] // Journal of biomedical informatics. — 2012. — т. 45, № 5. — с. 885—892.

75. Feature engineering for recognizing adverse drug reactions from twitter posts / H.-J. Dai [и др.] // Information. — 2016. — т. 7, № 2. — с. 27.

76. UZH@ SMM4H: System Descriptions / T. Ellendorff [и др.] // Proceedings of the 2018 EMNLP Workshop SMM4H: The 3rd Social Media Mining for Health Applications Workshop & Shared Task. — 2018. — с. 56—60.

77. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding / J. Devlin [и др.] // Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). — 2019. — с. 4171—4186.

78. A rich feature vector for protein-protein interaction extraction from multiple corpora / M. Miwa [и др.] // Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1. — Association for Computational Linguistics. 2009. — с. 121—130.

79. All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning / A. Airola [и др.] // BMC bioinformatics. — 2008. — т. 9, № 11. — S2.

80. *Huang, M.* A hybrid method for relation extraction from biomedical literature / M. Huang, X. Zhu, M. Li // International journal of medical informatics. — 2006. — т. 75, № 6. — с. 443—455.

81. *Segura-Bedmar, I.* A linguistic rule-based approach to extract drug-drug interactions from pharmacological documents / I. Segura-Bedmar, P. Martınez, C. de Pablo-Sánchez // BMC bioinformatics. т. 12. — BioMed Central. 2011. — S1.

82. *Kilicoglu, H.* Adapting a general semantic interpretation approach to biological event extraction / H. Kilicoglu, S. Bergler // Proceedings of the BioNLP Shared Task 2011 Workshop. — Association for Computational Linguistics. 2011. — с. 173—182.

83. *Baumgartner, W. A.* An open-source framework for large-scale, flexible evaluation of biomedical text mining systems / W. A. Baumgartner, K. B. Cohen, L. Hunter // Journal of biomedical discovery and collaboration. — 2008. — т. 3, № 1. — с. 1.

84. An overview of BioCreative II. 5 / F. Leitner [и др.] // IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB). — 2010. — т. 7, № 3. — с. 385—399.

85. Comparative analysis of five protein-protein interaction corpora / S. Pyysalo [и др.] // BMC bioinformatics. т. 9. — BioMed Central. 2008. — S6.

86. Comparative experiments on learning information extractors for proteins and their interactions / R. Bunescu [и др.] // Artificial intelligence in medicine. — 2005. — т. 33, № 2. — с. 139—155.

87. Mining and evaluation of molecular relationships in literature / C. Senger [и др.] // Bioinformatics. — 2012. — т. 28, № 5. — с. 709—714.

88. *He, M.* PPI finder: a mining tool for human protein-protein interactions / M. He, Y. Wang, W. Li // PloS one. — 2009. — т. 4, № 2. — e4554.

89. *Torvik, V. I.* A quantitative model for linking two disparate sets of articles in MEDLINE / V. I. Torvik, N. R. Smalheiser // Bioinformatics. — 2007. — т. 23, № 13. — с. 1658—1665.

90. Integrating protein-protein interactions and text mining for protein function prediction / S. Jaeger [и др.] // BMC bioinformatics. т. 9. — BioMed Central. 2008. — S2.

91. Discovering patterns to extract protein–protein interactions from the literature: Part II / Y. Hao [и др.] // Bioinformatics. — 2005. — т. 21, № 15. — с. 3294—3300.

92. Relation extraction methods for biomedical literature / Q. Bui [и др.]. — Citeseer, 2012.

93. Inference of transcriptional regulatory network by bootstrapping patterns / H.-C. Wang [и др.] // Bioinformatics. — 2011. — т. 27, № 10. — с. 1422—1428.

94. Efficient extraction of protein-protein interactions from full-text articles / J. Hakenberg [и др.] // IEEE/ACM Transactions on Computational Biology and Bioinformatics. — 2010. — т. 7, № 3. — с. 481—494.

95. *Nguyen, Q. L.* Simple tricks for improving pattern-based information extraction from the biomedical literature / Q. L. Nguyen, D. Tikk, U. Leser // Journal of biomedical semantics. — 2010. — т. 1, № 1. — с. 9.

96. Mining protein-protein interactions from GeneRIFs with OpenDMAP / A. D. Fox [и др.] // Linking Literature, Information, and Knowledge for Biology. — Springer, 2010. — с. 43—52.

97. Measuring prediction capacity of individual verbs for the identification of protein interactions / D. Rebholz-Schuhmann [и др.] // Journal of biomedical informatics. — 2010. — т. 43, № 2. — с. 200—207.

98. *Choi, Y. S.* Tree pattern expression for extracting information from syntactically parsed text corpora / Y. S. Choi // Data Mining and Knowledge Discovery. — 2011. — т. 22, № 1/2. — с. 211—231.

99. Medication information extraction with linguistic pattern matching and semantic rules / I. Spasić [и др.] // Journal of the American Medical Informatics Association. — 2010. — т. 17, № 5. — с. 532—535.

100. Finding the evidence for protein-protein interactions from PubMed abstracts / H. Jang [и др.] // Bioinformatics. — 2006. — т. 22, № 14. — e220—e226.

101. Automated extraction of information on protein–protein interactions from the biological literature / T. Ono [и др.] // Bioinformatics. — 2001. — т. 17, № 2. — с. 155—161.

102. *Koike, A.* Automatic extraction of gene/protein biological functions from biomedical text / A. Koike, Y. Niwa, T. Takagi // Bioinformatics. — 2004. — т. 21, № 7. — с. 1227—1236.

103. Learning to extract relations for protein annotation / J.-H. Kim [и др.] // Bioinformatics. — 2007. — т. 23, № 13. — с. i256—i263.

104. Mining of relations between proteins over biomedical scientific literature using a deep-linguistic approach / F. Rinaldi [и др.] // Artificial intelligence in medicine. — 2007. — т. 39, № 2. — с. 127—136.

105. *Fundel, K.* RelEx—Relation extraction using dependency parse trees / K. Fundel, R. Küffner, R. Zimmer // Bioinformatics. — 2006. — т. 23, № 3. — с. 365—371.

106. An environment for relation mining over richly annotated corpora: the case of GENIA / F. Rinaldi [и др.] // BMC bioinformatics. т. 7. — BioMed Central. 2006. — S3.

107. UTH-CCB@ BioCreative V CDR task: identifying chemical-induced disease relations in biomedical text / J. Xu [и др.] // Proceedings of the Fifth BioCreative Challenge Evaluation Workshop. — 2015. — с. 254—259.

108. RELigator: chemical-disease relation extraction using prior knowledge and textual information / E. Pons [и др.] // Proceedings of the Fifth BioCreative Challenge Evaluation Workshop. — 2015. — с. 247—253.

109. The UET-CAM system in the BioCreAtIvE V CDR task / H.-Q. Le [и др.] // Fifth BioCreative challenge evaluation workshop. — 2015. — с. 208—213.

110. A comprehensive benchmark of kernel methods to extract protein–protein interactions from literature / D. Tikk [и др.] // PLoS computational biology. — 2010. — т. 6, № 7. — e1000837.

111. Protein–protein interaction extraction by leveraging multiple kernels and parsers / M. Miwa [и др.] // International journal of medical informatics. — 2009. — т. 78, № 12. — e39—e46.

112. *Kim, S.* Kernel approaches for genic interaction extraction / S. Kim, J. Yoon, J. Yang // Bioinformatics. — 2007. — т. 24, № 1. — с. 118—126.

113. *Segura-Bedmar, I.* Using a shallow linguistic kernel for drug–drug interaction extraction / I. Segura-Bedmar, P. Martinez, C. de Pablo-Sánchez // Journal of biomedical informatics. — 2011. — т. 44, № 5. — с. 789—804.

114. Walk-weighted subsequence kernels for protein-protein interaction extraction / S. Kim [и др.] // BMC bioinformatics. — 2010. — т. 11, № 1. — с. 107.

115. Kernel-based learning for biomedical relation extraction / J. Li [и др.] // Journal of the American Society for Information Science and Technology. — 2008. — т. 59, № 5. — с. 756—769.

116. *Giuliano, C.* Exploiting shallow linguistic information for relation extraction from biomedical literature / C. Giuliano, A. Lavelli, L. Romano // 11th Conference of the European Chapter of the Association for Computational Linguistics. — 2006.

117. Linguistic feature analysis for protein interaction extraction / T. Fayruzov [и др.] // BMC bioinformatics. — 2009. — т. 10, № 1. — с. 374.

118. Extracting protein-protein interactions from text using rich feature vectors and feature selection / S. Van Landeghem [и др.] // 3rd International symposium on Semantic Mining in Biomedicine (SMBM 2008). — Turku Centre for Computer Sciences (TUCS). 2008. — с. 77—84.

119. *Sætre, R.* Syntactic features for protein-protein interaction extraction. / R. Sætre, K. Sagae, J. Tsujii // LBM (Short Papers). — 2007. — т. 319.

120. *Kim, M.-Y.* Detection of gene interactions based on syntactic relations / M.-Y. Kim // BioMed Research International. — 2008. — т. 2008.

121. PIE: an online prediction system for protein–protein interactions from text / S. Kim [и др.] // Nucleic acids research. — 2008. — т. 36, suppl_2. — W411—W415.

122. BioEve: bio-molecular event extraction from text using semantic classification and dependency parsing / S. T. Ahmed [и др.] // Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task. — Association for Computational Linguistics. 2009. — с. 99—102.

123. *Niu, Y.* Evaluation of linguistic features useful in extraction of interactions from PubMed; application to annotating known, high-throughput and predicted interactions in I2D / Y. Niu, D. Otasek, I. Jurisica // Bioinformatics. — 2009. — т. 26, № 1. — с. 111—119.

124. Overview of the BioCreative V chemical disease relation (CDR) task / C.-H. Wei [и др.] // Proceedings of the fifth BioCreative challenge evaluation workshop. т. 14. — 2015.

125. Distant supervision for relation extraction via piecewise convolutional neural networks / D. Zeng [и др.] // Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. — 2015. — с. 1753—1762.

126. Neural relation extraction with selective attention over instances / Y. Lin [и др.] // Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). — 2016. — с. 2124—2133.

127. Relation Classification via Convolutional Deep Neural Network / D. Zeng [и др.] // Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. — 2014. — с. 2335—2344.

128. Bidirectional long short-term memory networks for relation classification / S. Zhang [и др.] // Proceedings of the 29th Pacific Asia conference on language, information and computation. — 2015. — с. 73—78.

129. *Ye, Z.-X.* Multi-Level Matching and Aggregation Network for Few-Shot Relation Classification / Z.-X. Ye, Z.-H. Ling // CoRR. — 2019. — т. abs/1906.06678. — arXiv: 1906.06678. — URL: http://arxiv.org/abs/1906.06678.

130. Mining clinical relationships from patient narratives / A. Roberts [и др.] // BMC bioinformatics. — 2008. — т. 9, № 11. — s3.

131. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text / Ö. Uzuner [и др.] // Journal of the American Medical Informatics Association. — 2011. — т. 18, № 5. — с. 552—556.

132. *Roberts, K.* Extraction of medical concepts, assertions, and relations from discharge summaries for the fourth i2b2/VA shared task / K. Roberts, B. Rink, S. Harabagiu // Proceedings of the 2010 i2b2/VA Workshop on Challenges in Natural Language Processing for Clinical Data. Boston, MA, USA: i2b2. — 2010.

133. NRC at i2b2: one challenge, three practical tasks, nine statistical systems, hundreds of clinical records, millions of useful features / B. de Bruijn [и др.] // Proceedings of the 2010 i2b2/VA Workshop on Challenges in Natural Language Processing for Clinical Data. Boston, MA, USA: i2b2. — 2010.

134. CARAMBA: concept, assertion, and relation annotation using machine-learning based approaches / C. Grouin [и др.] // i2b2 Medication Extraction Challenge Workshop. — 2010.

135. I2b2 challenges in clinical natural language processing 2010 / J. Patrick [и др.] // Proceedings of the 2010 i2b2/VA Workshop on Challenges in Natural Language Processing for Clinical Data. Boston, MA, USA: i2b2.

136. *Jonnalagadda, S.* Can distributional statistics aid clinical concept extraction / S. Jonnalagadda, G. Gonzalez // Proceedings of the 2010 i2b2/VA workshop on challenges in natural language processing for clinical data. Boston, MA, USA: i2b2. — 2010.

137. Salt lake city VA's challenge submissions / G. Divita, O. Treitler, Y. Kim [и др.] // Proceedings of the 2010 i2b2/VA Workshop on Challenges in Natural Language Processing for Clinical Data. Boston, MA, USA: i2b2. — 2010.

138. *Solt, I.* Concept, Assertion and Relation Extraction at the 2010 i2b2 Relation Extraction Challenge using parsing information and dictionaries / I. Solt, F. P. Szidarovszky, D. Tikk // Proc. of i2b2/VA Shared-Task. Washington, DC. — 2010.

139. NLM's system description for the fourth i2b2/VA challenge / D. Demner-Fushman, E. Apostolova, R. Islamaj Dogan [и др.] // Proceedings of the 2010 i2b2/VA Workshop on Challenges in Natural Language Processing for Clinical Data. Boston, MA, USA: i2b2. — 2010.

140. *D'Souza, J.* Knowledge-rich temporal relation identification and classification in clinical notes / J. D'Souza, V. Ng // Database. — 2014. — т. 2014.

141. Relation extraction from clinical texts using domain invariant convolutional neural network / S. Sahu [и др.] // Proceedings of the 15th Workshop on Biomedical Natural Language Processing. — 2016. — с. 206—215.

142. Clinical relation extraction with deep learning / X. Lv [и др.] // IJHIT. — 2016. — т. 9, № 7. — с. 237—248.

143. Detecting adverse drug events with rapidly trained classification models / A. B. Chapman [и др.] // Drug safety. — 2019. — с. 1—10.

144. *Dandala, B.* Adverse drug events detection in clinical notes by jointly modeling entities and relations using neural networks / B. Dandala, V. Joopudi, M. Devarakonda // Drug safety. — 2019. — с. 1—12.

145. *Xu, D.* UArizona at the MADE1. 0 NLP Challenge / D. Xu, V. Yadav, S. Bethard // Proceedings of machine learning research. — 2018. — т. 90. — с. 57.

146. *Magge, A.* Clinical NER and relation extraction using bi-char-LSTMs and random forest classifiers / A. Magge, M. Scotch, G. Gonzalez-Hernandez // International Workshop on Medication and Adverse Drug Event Detection. — 2018. — с. 25—30.

147. *Munkhdalai, T.* Clinical relation extraction toward drug safety surveillance using electronic health record narratives: classical learning versus deep learning / T. Munkhdalai, F. Liu, H. Yu // JMIR public health and surveillance. — 2018. — т. 4, № 2.

148. *Shelmanov, A.* Information extraction from clinical texts in Russian / A. Shelmanov, I. Smirnov, E. Vishneva // Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference "Dialogue. т. 14. — 2015. — с. 537—549.

149. Клиническая фармакология и фармакотерапия / В. Г. Кукес [и др.]. — Общество с ограниченной ответственностью Издательская группа ГЭОТАР-Медиа, 2003.

150. Interactive attention networks for aspect-level sentiment classification / D. Ma [и др.] // arXiv preprint arXiv:1709.00893. — 2017.

151. Recommending Themes for Ad Creative Design via Visual-Linguistic Representations / Y. Zhou [и др.] // Proceedings of The Web Conference 2020. — 2020. — с. 2521—2527.

152. Attention-Fused Deep Matching Network for Natural Language Inference. / C. Duan [и др.] // IJCAI. — 2018. — с. 4033—4040.

153. *Wang, Y.* Image caption with synchronous cross-attention / Y. Wang, J. Liu, X. Wang // Proceedings of the on Thematic Workshops of ACM Multimedia 2017. — 2017. — с. 433—441.

154. *Kiritchenko, S.* Sentiment analysis of short informal texts / S. Kiritchenko, X. Zhu, S. M. Mohammad // Journal of Artificial Intelligence Research. — 2014. — т. 50. — с. 723—762.

155. *Loper, E.* NLTK: the natural language toolkit / E. Loper, S. Bird // arXiv preprint cs/0205028. — 2002.

156. Texterra: A framework for text analysis / D. Y. Turdakov [и др.] // Programming and Computer Software. — 2014. — т. 40, № 5. — с. 288—295.

157. *Baccianella, S.* Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. / S. Baccianella, A. Esuli, F. Sebastiani // Lrec. т. 10. — 2010. — с. 2200—2204.

158. *Wilson, T.* Recognizing contextual polarity in phrase-level sentiment analysis / T. Wilson, J. Wiebe, P. Hoffmann // Proceedings of the conference on human language technology and empirical methods in natural language processing. — Association for Computational Linguistics. 2005. — с. 347—354.

159. *Hu, M.* Mining and summarizing customer reviews / M. Hu, B. Liu // Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. — ACM. 2004. — с. 168—177.

160. *Loukachevitch, N. V.* Creating a General Russian Sentiment Lexicon. / N. V. Loukachevitch, A. Levchik // LREC. — 2016.

161. *Z, M.* Identifying disease-related expressions in reviews using conditional random fields / M. Z, T. EV, T. AE // Computational Linguistics and Intellectual Technologies. — 2017. — с. 155—166.

162. Cadec: A corpus of adverse drug event annotations / S. Karimi [и др.] // Journal of biomedical informatics. — 2015. — т. 55. — с. 73—81.

163. Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features / A. Nikfarjam [и др.] // Journal of the American Medical Informatics Association. — 2015. — т. 22, № 3. — с. 671—681.

164. NLP Challenges for Detecting Medication and Adverse Drug Events from Electronic Health Records (MADE1.0) / F. Liu [и др.]. — 2018. — URL: https://bio-nlp.org/index.php/projects/39-nlp-challenges.

165. *Alvaro, N.* TwiMed: Twitter and PubMed comparable corpus of drugs, diseases, symptoms, and their relations / N. Alvaro, Y. Miyao, N. Collier // JMIR public health and surveillance. — 2017. — т. 3, № 2.

166. A systematic approach for developing a corpus of patient reported adverse drug events: A case study for SSRI and SNRI medications / M. Zolnoori [и др.] // Journal of biomedical informatics. — 2019. — т. 90. — с. 103091.

167. *Kim, Y.* Convolutional neural networks for sentence classification / Y. Kim // arXiv preprint arXiv:1408.5882. — 2014.

168. A C-LSTM neural network for text classification / C. Zhou [и др.] // arXiv preprint arXiv:1511.08630. — 2015.

169. Biobert: pre-trained biomedical language representation model for biomedical text mining / J. Lee [и др.] // arXiv preprint arXiv:1901.08746. — 2019.

170. *Dos Santos, C.* Deep convolutional neural networks for sentiment analysis of short texts / C. Dos Santos, M. Gatti // Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. — 2014. — с. 69—78.

171. *Gurulingappa, H.* Extraction of potential adverse drug events from medical case reports / H. Gurulingappa, A. Mateen-Rajpu, L. Toldo // Journal of biomedical semantics. — 2012. — т. 3, № 1. — с. 15.

172. Attention is all you need / A. Vaswani [и др.] // Advances in neural information processing systems. — 2017. — с. 5998—6008.

173. Identification of Adverse Drug Reaction Mentions in Tweets–SMM4H Shared Task 2019 / S. Rawal [и др.] // Proceedings of the Fourth Social Media Mining for Health Applications (# SMM4H) Workshop & Shared Task. — 2019. — с. 136—137.

174. Deeppavlov: Open-source library for dialogue systems / M. Burtsev [и др.] // Proceedings of ACL 2018, System Demonstrations. — 2018. — с. 122—127.

175. Effective LSTMs for target-dependent sentiment classification / D. Tang [и др.] // arXiv preprint arXiv:1512.01100. — 2015.

176. *Tang, D.* Aspect level sentiment classification with deep memory network / D. Tang, B. Qin, T. Liu // arXiv preprint arXiv:1605.08900. — 2016.

177. Recurrent attention network on memory for aspect sentiment analysis / P. Chen [и др.] // Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. — 2017. — с. 452—461.

178. Annotation and detection of drug effects in text for pharmacovigilance / P. Thompson [и др.] // Journal of cheminformatics. — 2018. — т. 10, № 1. — с. 37.

179. *Shen, Y.* Attention-based convolutional neural network for semantic relation extraction / Y. Shen, X.-J. Huang // Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. — 2016. — с. 2526—2536.

180. BioWordVec, improving biomedical word embeddings with subword information and MeSH / Y. Zhang [и др.] // Scientific data. — 2019. — т. 6, № 1. — с. 1—9.

181. *Gu, J.* Chemical-induced disease relation extraction with various linguistic features / J. Gu, L. Qian, G. Zhou // Database. — 2016. — т. 2016.

182. *Maaten, L. v. d.* Visualizing data using t-SNE / L. v. d. Maaten, G. Hinton // Journal of machine learning research. — 2008. — т. 9, Nov. — с. 2579—2605.

Список рисунков

Список таблиц

Приложение А

Результаты оценки моделей на основе LSTM для классификации сущностей по типам

		non-ADR			ADR			Макро		
Корпус	**Модель**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Twitter	LSTM+CA	.654	.627	.634	.951	.957	.954	.802	.792	.794
	RAM	.779	.653	.705	.955	.973	.964	.867	.813	.834
	MemNet	.559	.667	.590	.954	.918	.935	.757	.792	.763
	TD_LSTM	.606	.547	.570	.940	.952	.946	.773	.749	.758
	LSTM	.388	.427	.392	.920	.889	.903	.618	.621	.613
CADEC	LSTM+CA	.699	.637	.662	.966	.972	.969	.832	.805	.815
	RAM	.696	.406	.506	.946	.981	.963	.821	.694	.734
	MemNet	.575	.570	.559	.960	.955	.957	.767	.762	.758
	TD_LSTM	.630	.557	.582	.958	.967	.962	.794	.762	.772
	LSTM	.664	.554	.602	.958	.973	.966	.811	.764	.784
MADE	LSTM+CA	.982	.991	.986	.740	.524	.585	.861	.758	.786
	RAM	.980	.989	.985	.615	.486	.538	.798	.737	.761
	MemNet	.979	.991	.985	.684	.447	.535	.832	.719	.760
	TD_LSTM	.980	.988	.984	.606	.470	.515	.793	.729	.750
	LSTM	.981	.989	.985	.636	.510	.557	.809	.749	.771
Twimed-Twitter	LSTM+CA	.802	.825	.813	.836	.813	.824	.819	.819	.819
	RAM	.799	.736	.764	.773	.823	.796	.786	.779	.780
	MemNet	.772	.821	.789	.823	.791	.801	.798	.806	.795
	TD_LSTM	.731	.711	.717	.741	.751	.742	.736	.731	.730
	LSTM	.669	.757	.709	.743	.649	.691	.706	.703	.700
Twimed-Twitter	LSTM+CA	.936	.977	.956	.878	.738	.792	.907	.858	.874
	RAM	.917	.916	.916	.675	.669	.662	.796	.792	.789
	MemNet	.929	.912	.917	.736	.748	.705	.833	.830	.811
	TD_LSTM	.495	.493	.487	.932	.930	.931	.714	.712	.709
	LSTM	.929	.949	.939	.786	.707	.740	.858	.828	.839
PsyTAR	LSTM+CA	.848	.712	.752	.841	.915	.873	.844	.814	.812
	RAM	.808	.736	.765	.836	.875	.853	.822	.805	.809
	MemNet	.507	.531	.519	.737	.797	.757	.622	.664	.638
	TD_LSTM	.847	.872	.859	.907	.889	.898	.877	.888	.878
	LSTM	.825	.846	.834	.898	.880	.888	.861	.863	.861
RuDReC	LSTM+CA	.877	.889	.882	.696	.655	.665	.786	.772	.774
	RAM	.789	.942	.857	.575	.308	.387	.682	.625	.622
	MemNet	.805	.927	.860	.719	.392	.482	.762	.66	.671
	TD_LSTM	.843	.914	.875	.709	.537	.595	.776	.726	.735
	LSTM	.844	.834	.837	.579	.585	.570	.712	.709	.703