Министерство образования и науки Российской Федерации

Московский физико-технический институт (государственный университет)

Физтех-школа прикладной математики и информатики

Кафедра интеллектуальной обработки документов

Выпускная квалификационная работа бакалавра

Генерация сложных примеров для нормализации денежных сумм

Автор:

Студент Б05-922 группы

Малхасян Арсен Жирайрович

Научный руководитель:

Афанасьев Денис Евгеньевич

Москва 2023

Аннотация

енерация сложных примеров для нормализации денежных сумм

Малхасян Арсен Жирайрович

Данная дипломная работа посвящена проблеме классификации и нормализации денежных аннотаций с использованием методов глубокого обучения. Целью работы является разработка эффективной модели, способной автоматически распознавать и классифицировать суммы денег из текстовых данных.

В начале работы проведен анализ предметной области и изучены существующие подходы к решению задачи классификации денежных аннотаций. Для решения поставленной задачи были применены методы глубокого обучения, включая BiLSTM, self-attention и CRF.

В ходе экспериментов было проведено сравнение различных моделей и архитектур с использованием реальных и синтетических данных. Переход от простой модели к более сложной архитектуре привел к значительному улучшению результатов. Кроме того, генерация синтетических данных оказала существенное влияние на метрики модели, приводя к улучшению производительности и точности.

В результате проведенных исследований удалось достичь определенной точности и надежности в классификации и нормализации денежных аннотаций.

Данная работа вносит вклад в область обработки естественного языка и глубокого обучения, а также может быть полезной для разработки систем, связанных с автоматической обработкой финансовых данных и денежных аннотаций.

Оглавление

Глава 1

Введение

В современном информационном обществе, где огромные объемы данных охватывают множество сфер деятельности, нормализация и структурирование информации становятся неотъемлемыми задачами, поскольку эффективное управление и анализ данных становится критически важным для принятия обоснованных решений и достижения успеха в различных отраслях. Без нормализации данных, особенно когда речь идет о денежных суммах, возникают проблемы с точностью и сопоставимостью, что может привести к ошибкам в финансовых отчетах, недостоверным статистическим данным и потере доверия со стороны заинтересованных сторон.

Сегодня мы сталкиваемся с беспрецедентным ростом объема информации, которую мы генерируем и потребляем. Во время проведения транзакций, выполнения операций и взаимодействия в цифровой среде, денежные суммы представляют собой неотъемлемую часть информации, которую нужно нормализовать, чтобы обеспечить ее стандартизацию и обработку в соответствии с требованиями и правилами. Поэтому дипломный проект сфокусирован на разработке и генерации сложных примеров для нормализации денежных сумм, что поможет в создании надежных и эффективных алгоритмов, способных обрабатывать денежные данные с высокой точностью и надежностью.

1.1 Предметная область и актуальность задачи

Проект связан с обработкой текстов и аннотаций, содержащих денежные суммы. Анализ и извлечение информации из таких текстовых данных представляют собой актуальную задачу, требующую разработки эффективных и точных методов. В современном мире, где финансовые операции, экономические отчеты и коммерческие

сделки играют важную роль, надежное и автоматизированное определение денежных сумм является необходимостью. Точная нормализация и структурирование этих сумм являются ключевыми шагами для обеспечения точности и надежности анализа таких текстовых данных.

1.2 Цель исследования и задачи

Основной целью является обнаруживать и разработать инновационные подходы, которые могут применяться в практических сценариях для эффективной обработки и анализа данных о денежных суммах с использованием методов анализа и классификации текстовых данных с дополнением новых сложных данных с помощью алгоритимической генереации, с целью выделения и категоризации различных классов денежных сумм и валют[1].

1.3 Работы, которые нам предстоит делать для решение задачи

Подготовка данных: Необходимо собрать набор текстов и соответствующие им аннотации, содержащие денежные суммы, валюты[1]. Данные должны быть предварительно обработаны, чтобы учесть особенности форматирования, различные валютные символы и возможные варианты записи.

Анализ и классификация: Необходимо провести детальный анализ текстов и аннотаций для выделения различных классов денежных сумм, валют и сумм, записанных в скобках.

Использование различных подходов: В данной работе будут использованы различные методы глубокого обучения, нейронные сети - обработки естественного языка (NLP) для достижения более точных и надежных результатов, включая модели с распознаванием именованных сущностей (NER). Эти методы позволяют более точно и надежно распознавать и извлекать сущности из текстовых данных. Методы глубокого обучения, основанные на нейронных сетях, показали высокую эффективность в обработке текстовых данных и классификации.

Использование нейронных сетей позволяет моделировать сложные зависимости

[1]5 классов - major amount, minor amount, major currency, minor currency, amount bracketed

между текстом и его семантическим содержанием, что особенно полезно при анализе и классификации денежных сумм, где существуют различные варианты форматирования и записи.

Сравнение подходов: Проанализировать и сравнить разработанные подходы, исследованием преимущества и недостатки предложенного алгоритма по сравнению с другими подходами и методами. Сделать выводы о его применимости, эффективности и потенциальной области применения.

В рамках данного проекта мы планируем провести исследование, нацеленное на изучение различных подходов к работе искусственного интеллекта и методов генерации сложных примеров для нормализации денежных сумм.

Используя принципы и методы искусственного интеллекта, мы намерены исследовать различные модели глубокого обучения и алгоритмы обработки естественного языка, с целью определения наиболее эффективных подходов к генерации сложных примеров для нормализации денежных сумм. Мы уделим особое внимание разработке моделей, способных автоматически распознавать и корректировать различные форматы записи, валютные символы и другие особенности, связанные с денежными суммами.

Глава 2

Обзор литературы

2.1 Нейронные сети

Нейронные сети (Neural Networks) являются мощным инструментом машинного обучения, вдохновленным биологической нервной системой. Они состоят из множества связанных и взаимодействующих нейронов, которые обрабатывают и передают информацию. В этом разделе будет представлен обзор некоторых ключевых концепций и архитектур нейронных сетей.

2.1.1 Архитектура нейронной сети

Нейронные сети состоят из нескольких слоев, включая входной, скрытые и выходной слои. Каждый слой содержит набор нейронов, которые вычисляют взвешенные суммы входных сигналов и применяют нелинейную функцию активации для создания нелинейных преобразований. Различные архитектуры нейронных сетей имеют различное количество и типы слоев, что позволяет моделировать сложные функции и обрабатывать данные различных типов.

2.1.2 Сверточные нейронные сети (CNN)

Сверточные нейронные сети (Convolutional Neural Networks, CNN) — это класс нейронных сетей, широко применяемый в анализе изображений, а также в обработке текста и звука. Они основаны на использовании операций свертки и пулинга для извлечения локальных шаблонов и признаков из входных данных.

Архитектура CNN обычно состоит из трех основных типов слоев: сверточных слоев, слоев объединения (пулинга) и полносвязных слоев.

- **Сверточные слои:** Сверточные слои выполняют операцию свертки по входным данным с помощью фильтров. Каждый фильтр представляет собой матрицу весов, называемую ядром свертки или фильтром. Операция свертки вычисляет сумму произведений элементов ядра свертки с соответствующими элементами входных данных. Формула для операции свертки имеет вид:

$$C(i,j) = \sum_m \sum_n I(i+m, j+n) \cdot K(m,n), \tag{2.1}$$

где $C(i,j)$ - выходной пиксель, $I(i,j)$ - входной пиксель, $K(m,n)$ - элемент ядра свертки.

- **Слои объединения (пулинга):** Слои объединения выполняют уменьшение размерности карт признаков, выбирая наиболее значимые значения. Они помогают снизить количество параметров и вычислений в сети, а также делают представление инвариантным к масштабу и небольшим переворотам. Один из наиболее распространенных методов пулинга - операция максимального пулинга, которая выбирает максимальное значение в определенной области.

- **Полносвязные слои:** Полносвязные слои находятся в конце архитектуры CNN и служат для классификации или регрессии. Они связывают все признаки предыдущих слоев и генерируют соответствующий выход.

2.1.3 Рекуррентные нейронные сети (RNN) и LSTM

Рекуррентные нейронные сети (Recurrent Neural Networks, RNN) являются классом нейронных сетей, предназначенных для обработки последовательных данных, сохраняя информацию о предыдущих состояниях. Однако стандартные RNN могут столкнуться с проблемой исчезающего или взрывного градиента, когда информация о далеких зависимостях теряется или становится неустойчивой.

Для решения этой проблемы была предложена архитектура Long Short-Term Memory (LSTM). LSTM является особым типом RNN, который способен сохранять и использовать долгосрочные зависимости в данных. Он достигает этого за счет использования специальных внутренних структур, таких как вентили (gates), которые контролируют поток информации внутри сети.

LSTM состоит из нескольких вентилей, включая вентиль забывания (forget gate), входной вентиль (input gate) и выходной вентиль (output gate). Вентили позволяют сети контролировать поток информации, регулируя, какая информация должна быть сохранена, забыта или передана дальше. Каждый вентиль имеет свои весовые коэффициенты, которые автоматически настраиваются в процессе обучения.

Формулы для LSTM включают следующие шаги:

1. Вычисление вектора входного вентиля:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i), \tag{2.2}$$

2. Вычисление вектора вентиля забывания:

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f), \tag{2.3}$$

3. Вычисление вектора выходного вентиля:

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o), \tag{2.4}$$

4. Вычисление вектора обновления состояния:

$$u_t = \tanh(W_{xu}x_t + W_{hu}h_{t-1} + b_u), \tag{2.5}$$

5. Обновление внутреннего состояния:

$$c_t = f_t \odot c_{t-1} + i_t \odot u_t, \tag{2.6}$$

6. Вычисление выхода LSTM:

$$h_t = o_t \odot \tanh(c_t), \tag{2.7}$$

где x_t - входной вектор в момент времени t, h_t - скрытый состояние сети в момент времени t, c_t - внутреннее состояние сети в момент времени t, W и b - матрицы весов и векторы смещения, σ - функция активации, такая как сигмоида, а $\odot$ обозначает поэлементное умножение.

Модель LSTM позволяет сети запоминать информацию на протяжении длительного времени, а также контролировать поток информации, делая ее более устойчивой к проблеме исчезающего или взрывного градиента.

2.2 Векторные представления (эмбеддинги)

Векторные представления, также известные как эмбеддинги, являются способом представления слов и текстов в виде числовых векторов. Они играют важную роль в обработке естественного языка, позволяя моделям машинного обучения работать с текстовыми данными.

- **Эмбеддинги слов** представляют каждое слово в тексте в виде плотного вектора фиксированной размерности, где смысловая близость слов отображается близостью векторов. Эмбеддинги слов позволяют моделям учитывать семантические и синтаксические отношения между словами.
- **Эмбеддинги символов** представляют отдельные символы или подстроки слов в виде векторов. Они полезны в случаях, когда имеется ограниченный словарный запас или когда важны морфологические особенности слов.
- **Эмбеддинги предобученные на больших корпусах текста**, такие как Word2Vec, GloVe или FastText, обладают способностью захватывать семантические и синтаксические свойства слов.

Использование эмбеддингов позволяет моделям машинного обучения работать с текстовыми данными, улавливать семантические связи между словами и повышать качество их предсказаний.

2.3 Обработка естественного языка (NLP)

2.3.1 Основные концепции NLP

- NLP и его роль: Обзор дисциплины NLP, ее цели и применение в различных областях, включая машинный перевод, анализ тональности, генерация текста и многое другое.
- Языковые модели: Введение в языковые модели, которые используются для моделирования естественного языка и прогнозирования следующих слов в тексте с помощью вероятностных методов.
- Задачи NLP: Обзор основных задач в NLP, таких как сегментация текста, токенизация, частеречная разметка, синтаксический анализ, семантический анализ, извлечение информации и многое другое.

– Основные инструменты и библиотеки: Обзор популярных инструментов и библиотек, используемых для реализации NLP-решений, таких как NLTK, SpaCy, Gensim и TensorFlow.

2.3.2 Извлечение признаков из текста

– Представление текста: Обзор различных методов представления текста в виде числовых векторов, таких как мешок слов (bag-of-words), TF-IDF и эмбеддинги слов.

– Модели представления документов: Изучение методов представления документов, таких как векторное пространственное моделирование (VSM), латентное размещение Дирихле (LDA) и Doc2Vec.

– Применение методов извлечения признаков: Обзор применения методов извлечения признаков в задачах NLP, таких как классификация текста, кластеризация, анализ тональности и многое другое.

2.3.3 Именованное сущностьное распознавание (NER)

Именованное сущностьное распознавание (NER) является важной задачей в области обработки естественного языка и относится к процессу определения и классификации именованных сущностей, таких как имена людей, названия организаций, географические места, даты, времена и другие в текстовых данных. Это помогает извлекать и структурировать информацию из текста и играет важную роль во многих приложениях NLP, таких как информационный поиск, извлечение информации, автоматическая обработка документов и многое другое.

Определение именованных сущностей

Именованные сущности могут быть различных типов, таких как люди, организации, локации, даты, числа и другие сущности, которые имеют уникальный идентификатор и могут быть именованы.

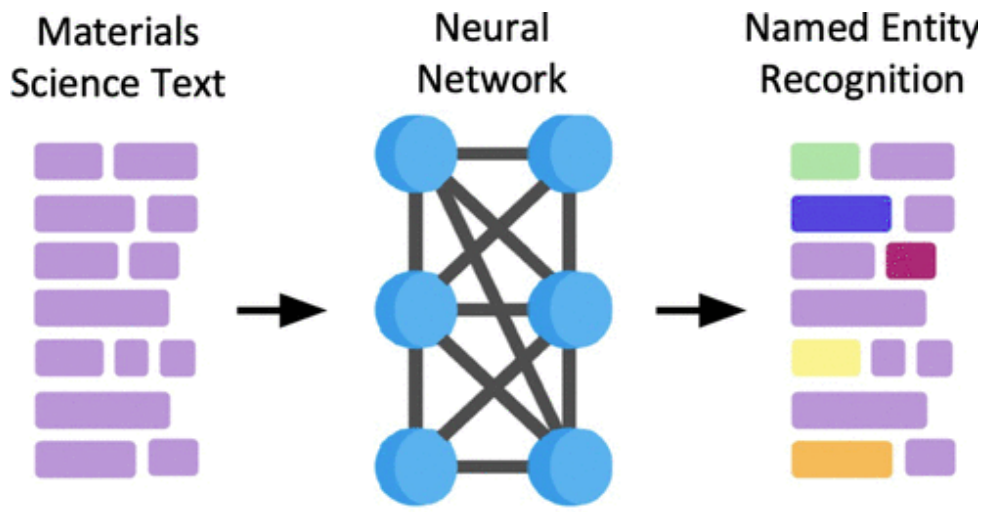


Рис. 2.1: распознавание именованных сущностей

Алгоритмы, методы и применение NER

Мы будем рассмотривать различные подходы и алгоритмы, используемые для именованного сущностьного распознавания. Будут рассмотрены методы, условные случайные поля (CRF), рекуррентные нейронные сети (RNN), сверточные нейронные сети (CNN) и attention-ы.

Мы также рассмотрим некоторые инструменты и библиотеки, которые могут быть использованы для реализации именованного сущностьного распознавания, например, библиотеки SpaCy, NLTK(для токенизации), Stanford NER и другие.

Важно отметить, что именованное сущностьное распознавание является сложной задачей в области NLP, так как оно требует не только распознавания сущностей, но и их классификации в соответствующие категории. Для достижения высокой точности в задаче NER может потребоваться комбинирование различных методов и подходов, а также обучение на больших размеченных корпусах данных.

Слой внимания (Attention Layer)

Слой внимания используется для моделирования взаимодействия между элементами входных данных и определения их важности в контексте задачи NER. Один из наиболее распространенных типов слоев внимания - это механизм внимания dot product.

Пусть у нас есть входные данные: последовательность слов $X = (x_1, x_2, ..., x_n)$ и контекстуализированное представление этих слов $H = (h_1, h_2, ..., h_n)$, полученное от предыдущих слоев модели. Для каждого элемента h_i входных данных мы можем вычислить весовой коэффициент α_i, который отражает его важность в контексте задачи NER. Формула для вычисления весового коэффициента может быть записана следующим образом:

$$\alpha_i = \sum_{j=1}^{n} \frac{exp(e_j)}{exp(e_i)} \tag{2.8}$$

где e_i - это энергия, вычисляемая на основе сходства между элементом h_i и текущим контекстом модели. Один из простых способов вычислить энергию - это использовать скалярное произведение:

$$e_i = h_i^T \cdot W \cdot c \tag{2.9}$$

где W - матрица обучаемых параметров, c - контекст модели.

После вычисления весовых коэффициентов α_i, мы можем получить контекстуализированное представление C, взвешенно усреднив элементы H:

$$C = \sum_{i=1}^{n} \alpha_i \cdot h_i \tag{2.10}$$

Контекстуализированное представление C может быть использовано для дальнейшего распознавания именованных сущностей.

Условные случайные поля (CRF)

Условные случайные поля (CRF) являются графическими моделями, которые моделируют зависимости между метками именованных сущностей в последовательностях данных. Они основаны на предположении о марковском свойстве, что метка текущей позиции зависит только от предыдущей метки в последовательности.

Пусть у нас есть последовательность слов $X = (x_1, x_2, ..., x_n)$ и соответствующие метки именованных сущностей $Y = (y_1, y_2, ..., y_n)$. Цель CRF состоит в моделировании условной вероятности $P(Y|X)$.

Для моделирования зависимостей между метками, CRF определяет энергию для каждой возможной конфигурации меток в последовательности. Энергия

вычисляется как сумма весов, связанных с каждой меткой и ее контекстом. Формула для вычисления энергии выглядит следующим образом:

$$E(X,Y) = \sum_{i=1}^{n} \sum_{k=1}^{K} w_k \cdot f_k(y_{i-1}, y_i, X, i) \tag{2.11}$$

где K - количество различных типов меток именованных сущностей, w_k - весовой параметр для каждого типа метки, f_k - функция, которая вычисляет релевантные признаки для каждой пары меток и контекста.

Для вычисления условной вероятности $P(Y|X)$, CRF использует мягкое максимальное правдоподобие (soft maximum likelihood). Формула для вычисления вероятности определенной конфигурации меток выглядит следующим образом:

$$P(Y|X) = \frac{exp(-E(X,Y))}{\sum_{Y'} exp(-E(X,Y'))} \tag{2.12}$$

где сумма в знаменателе вычисляется по всем возможным конфигурациям меток в последовательности.

CRF может быть использован вместе с другими моделями, такими как рекуррентные нейронные сети или трансформеры, чтобы улучшить производительность в задаче NER.

Глава 3

Постановка задачи и требования

3.1 Описание задачи

В рамках данной дипломной работы требуется решить следующую задачу:

- Для каждого текста из имеющегося набора текстовых документов, предоставленного в качестве входных данных, существует набор аннотаций.
- Каждая аннотация представляет собой интервал, определенный внутри соответствующего текста.
- Задачей является классификация каждой поданнотации на пять заданных классов: *major amount* (основная сумма), *major currency* (основная валюта), *minor amount* (второстепенная сумма), *minor currency* (второстепенная валюта) и *amount bracketed* (сумма в скобках).
- Важно отметить, что для данной задачи предоставлена разметка в формате XML, которая содержит информацию о текстах и аннотациях.

3.2 Требования к решению

Для успешного выполнения задачи необходимо учесть следующие требования:

1. **Эффективность вычислений:** Решение должно быть эффективным в использовании вычислительных ресурсов. Операции обработки текстов и аннотаций должны выполняться достаточно быстро, особенно при работе с большим объемом данных.

2. **Поддержка XML-формата:** Разработанная система должна быть способна корректно обрабатывать и анализировать XML-файлы с разметкой. Информация о текстах и аннотациях должна быть извлечена из XML-файлов для использования в процессе классификации.

3. **Корректная токенизация текста:** Должно обеспечиваться корректная токенизацая текста. Токенизация должна учитывать особенности языка и устанавливать границы между отдельными словами, символами и пунктуацией. Корректная токенизация является важным этапом обработки текстовых данных для достижения правильного понимания контекста и анализа.

4. **Методы обработки естественного языка (NLP):** Для достижения поставленной цели необходимо применить соответствующие методы обработки естественного языка (NLP). Это включает анализ текстовых данных, выделение ключевых особенностей и контекста, а также морфологический и синтаксический анализ текстов.

5. **Проектирование и обучение модели машинного обучения:** Требуется разработка и обучение модели машинного обучения, которая будет использоваться для классификации поданнотаций. Модель должна быть оптимально настроена и обучена на размеченных данных для достижения высокой точности и надежности.

6. **Выходные результаты:** Результатом работы разработанного алгоритма должна быть классификация каждой поданнотации на один из пяти заданных классов[1].. Кроме того, должны быть предоставлены соответствующие метки, соответствующие каждому классу, для каждой поданнотации.

[1]но могут быть еще подинтервалы(поданнотатации), которые не относятся не одному из классов. Например: знаки пунктуации

Глава 4

Экспериментальные исследования и анализ

В данной главе рассматривается методика исследования и разработки решения задачи анализа, классификации данных, направленной на выделение различных классов денежных сумм, валют и сумм, записанных в скобках и генерации сложных примеров.
Решение задачи состоит из нескольких этапов, описанных ниже:

4.1 Парсинг набора данных

Для начала необходимо произвести парсинг набора данных, представленного в формате XML. Данные включают текстовые фрагменты и соответствующие аннотации. Аннотации содержат информацию о денежных суммах, которые могут быть представлены двумя типами: "объекты с полными аннотациями"и "объекты с разрывными аннотациями".

- **Объекты с полными аннотациями:** описывают непрерывные интервалы в тексте, в которых находятся денежные суммы.

- **Объекты с разрывными аннотациями:** представляют собой денежные суммы, разбитые на несколько отдельных интервалов. Это может быть полезно, когда денежная сумма записана в нескольких фрагментах текста, разделенных другими словами или символами.

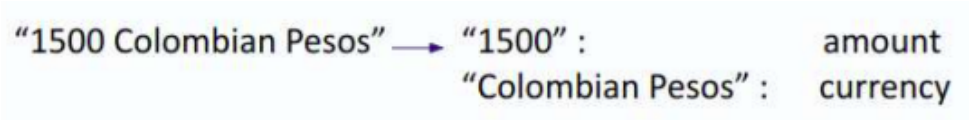


Рис. 4.1: пример объекта с полным аннотациям

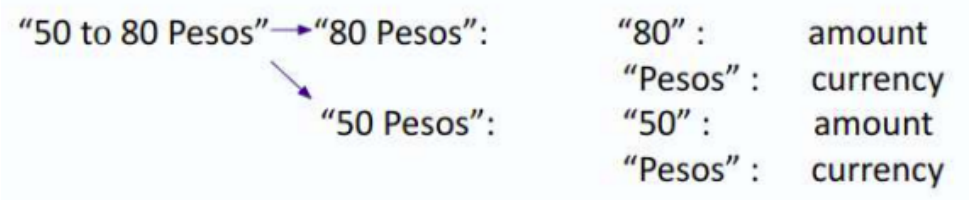


Рис. 4.2: пример объекта с разрывным аннотациям

Данный раздел включает два подпункта: обработку XML файла и токенизацию.

4.1.1 Обработка XML файла

В этом подпункте мы выполняем предварительную обработку XML файла, содержащего текст и аннотации. Обработка включает выделение соответвушие аннотации(интервалы) из текста размечание соответствующие поданнотации. Этот шаг позволяет получить структурированный набор данных, который будет использован для дальнейшего анализа.

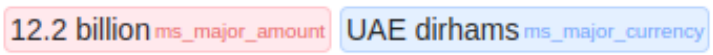
4.1.2 Токенизация

В этом подпункте мы проводим токенизацию текстовых данных. Токенизация разделяет текст на отдельные слова или токены, которые будут использоваться для анализа и классификации. Для токенизации текстовых данных в рамках данного исследования был использован токенайзер из библиотеки **razdel**. **Razdel** представляет собой библиотеку для разделения текста на токены, основанную на правилах.(имеет широкий спектр возможностей для токенизации текста на русском языке).

Однако, чтобы обеспечить более точную и полную токенизацию, были применены дополнительные методы. Регулярные выражения были использованы для обработки отдельных редко встречающихся символов, таких как денежные валюты, тире, нижнее подчеркивание и скобки: **-_MT€$£¥₫RF**

Эти символы имеют особое значение при анализе текстов и были учтены для более точной обработки. Таким образом, комбинация использования токенайзера **razdel** и дополнительных регулярных выражений обеспечила более точную и полную токенизацию текстовых данных, включая обработку редко встречающихся символов. Это было важным шагом для последующего анализа и классификации денежных сумм и валют в текстах.

пример:



12.2 billion UAE dirhams $\rightarrow$ [Substring$(0, 4, '12.2')$, Substring$(5, 12, 'billion')$, Substring$(13, 16, 'UAE')$, Substring$(17, 24, 'dirhams')$]

4.2 NER-BiLSTM нейронная сеть

Уже мы можем перейди к разработке нейронной сети, которая будет основана на подходе NER (Named Entity Recognition) и будет использовать модель BiLSTM. Наша нейронная сеть будет специально адаптирована для решения задачи выделения и классификации денежных сумм и валют в текстовых данных.

Мы создадим модель, которая будет обучаться на размеченных данных, чтобы научиться распознавать именованные сущности, включая денежные суммы, валюты и суммы, записанные в скобках. Для этого мы воспользуемся архитектурой BiLSTM (Bidirectional Long Short-Term Memory), которая позволяет учитывать контекстуальную информацию как слева, так и справа от текущего элемента, обеспечивая более точное выделение и классификацию.

Разрабатываемая нами нейронная сеть будет иметь слои BiLSTM, которые будут обучаться на текстовых данных, и слой классификации, который будет отвечать за определение классов денежных сумм и валют. Мы также проведем оптимизацию модели и настроим гиперпараметры для достижения наилучших результатов.

Архитектура CharLSTM модель:

1. Embedding символов: Входные символы преобразуются в векторные представления при помощи слоя эмбеддинга.

2. Dropout: Применяется dropout для улучшения обобщающей способности модели и снижения переобучения.

3. Линейный слой и функция активации: Векторы символов проходят через линейный слой, за которым следует функция активации (например, ReLU).

4. BI-LSTM: Используется двунаправленная LSTM (BI-LSTM) для извлечения контекстуальных признаков из последовательности символов.

5. Линейный слой: Выходы BI-LSTM подаются на линейный слой для классификации и определения классов денежных сумм и валют.

Каждый шаг в архитектуре NER-BiLSTM выполняется последовательно, и каждый слой играет свою роль в обработке и классификации текстовых данных.

Промежуточные результаты и метрики:

После обучения модели CharLSTM и проведения экспериментов на тестовых данных, были получены промежуточные результаты и метрики, которые свидетельствуют о качестве классификации и определения денежных сумм и валют.

Для оценки эффективности архитектуры CharLSTM решении задачи классификации и определения классов денежных сумм и валют, были использованы следующие метрики:

- Полнота (Recall): Измеряет способность модели обнаруживать все денежные суммы и валюты в тексте.
- Точность восстановления (Precision): Оценивает точность классификации найденных денежных сумм и валют.
- F-мера (F1-score): Комбинирует показатели полноты и точности для общей оценки модели.

	precision	recall	f1-score	support
outer	1.00	1.00	1.00	85916
ms_major_amount	1.00	1.00	1.00	5407
ms_major_currency	1.00	1.00	1.00	5019
ms_minor_amount	0.79	0.79	0.79	29
ms_minor_currency	0.91	0.74	0.82	27
ms_major_amount_bracketed	1.00	1.00	1.00	15
outer_like_brackets	0.97	0.81	0.88	47
accuracy			1.00	96460
macro avg	0.95	0.91	0.93	96460
weighted avg	1.00	1.00	1.00	96460

Рис. 4.3: метрики качество модели

4.3 Добавление синтетических данных:

Этот этап включает несколько шагов, которые направлены на генерацию сложных примеров. Он основан на анализе ошибок модели и использовании различных подходов к генерации синтетических данных. Ключевые этапы включают:

1. Анализ ошибок модели: Изучение ошибок, допущенных моделью при классификации и нормализации денежных сумм. Это позволяет выявить основные проблемные случаи и сложности, с которыми модель сталкивается.

```
ms_major_currency_syntetics = ['cents of euro','testone', '¢', 'sen', 'manats', 'yen',
                               'Brunei dollars', 'mills', 'giulio', '']
ms_major_amount_syntetics = ['lakh', 'a', 'A', '⅛', 'crore', 'Four', 'trillion', 'a single',
                             'Lacs', 'billion' 'lacs','sextillion', 'Ten', '']
ms_like_brackets_syntetics = ['(', '(', '(', '(', '(', '(', '’', 'and', ':', ';', 'of', '']
ms_minor_currency_syntetics = ['kopeks', 'kopecks', 'd', 'shillings', 'kobo', 'quadrans','']
```

Рис. 4.4: Данные, на которых модель ошиблась

2. Генерация новых синтетических данных:

- На основе ошибок сетки: Использование ошибочных примеров, на которых модель допускает ошибки, для создания новых синтетических данных. Это помогает модели лучше обучиться на сложных случаях.
- Использование символьного представления чисел (num2words в python): Применение символьного представления любого числа (например, ’123’ - ’сто двадцать три’) для создания новых примеров с различными денеж-

ными суммами.

– Учет порядка sub-аннотаций: Рассмотрение порядка поданнотаций (например, 'USD 10 million') для генерации синтетических данных с различными комбинациями и порядком денежных сумм и валют.

3. Добавление синтетических данных в обучающий набор: Сгенерированные синтетические данные добавляются в обучающий набор, чтобы обогатить его разнообразием примеров и улучшить обобщающую способность модели.

Для определения оптимального количества синтетических данных, которые следует добавить, был проведен эксперимент. Была изучена зависимость между количеством синтетических данных и качеством модели, измеряемым через показатель ошибок.

Исходные данные включали в себя тренировочный набор данных объемом 11500, валидационный набор данных объемом 1000 и тестовый набор данных объемом 3700. График позволяет нам увидеть, что при использовании синтетических данных объемом 1200 мы получаем наилучший результат.

Classification Report

	precision	recall	f1-score	support
outer	1.00	1.00	1.00	85916
ms_major_amount	1.00	1.00	1.00	5407
ms_major_currency	1.00	1.00	1.00	5019
ms_minor_amount	0.90	0.90	0.90	29
ms_minor_currency	0.96	0.85	0.90	27
ms_major_amount_bracketed	1.00	1.00	1.00	15
outer_like_brackets	0.93	0.87	0.90	47
accuracy			1.00	96460
macro avg	0.97	0.95	0.96	96460
weighted avg	1.00	1.00	1.00	96460

Рис. 4.5: метрики качество модели

4.4 Обновление эмбеддинг аннотации

Теперь продолжная эксперименты обновим эмбеддинги аннотации.

Обновление эмбеддинга аннотации включает несколько этапов, в соответствии с подходом, представленным в статье [2]. Опишем эти этапы:

1. Разбиение аннотаций на токены
2. Разбиение токенов на символы: Каждый токен разбивается на отдельные символы. Например, токен "USD"может быть разделен на символы "U "S и "D".
3. Эмбеддинг символов: Символы токенов преобразуются в векторные представления при помощи слоя эмбеддинга символов. Это позволяет модели учесть более мелкие детали и свойства символов в аннотации.
4. Эмбеддинг токенов: Разделенные токены также преобразуются в векторные представления при помощи слоя эмбеддинга токенов. Это позволяет модели учиться на основных характеристиках и смысле каждого токена.
5. Применение сверточной нейронной сети (CNN) к эмбеддингу символов: Эмбеддинг символов проходит через сверточный слой (CNN), который позволяет модели извлекать локальные контекстуальные признаки из после-

довательности символов. Это может быть полезно для распознавания особенностей, связанных с символами, например, специальными символами или редкими символами.

6. Конкатенация эмбеддинга символов и эмбеддинга токенов: Результаты эмбеддинга символов и эмбеддинга токенов конкатенируются в единый вектор аннотации. Это позволяет модели объединить информацию о символах и общем смысле токена в одном векторе.

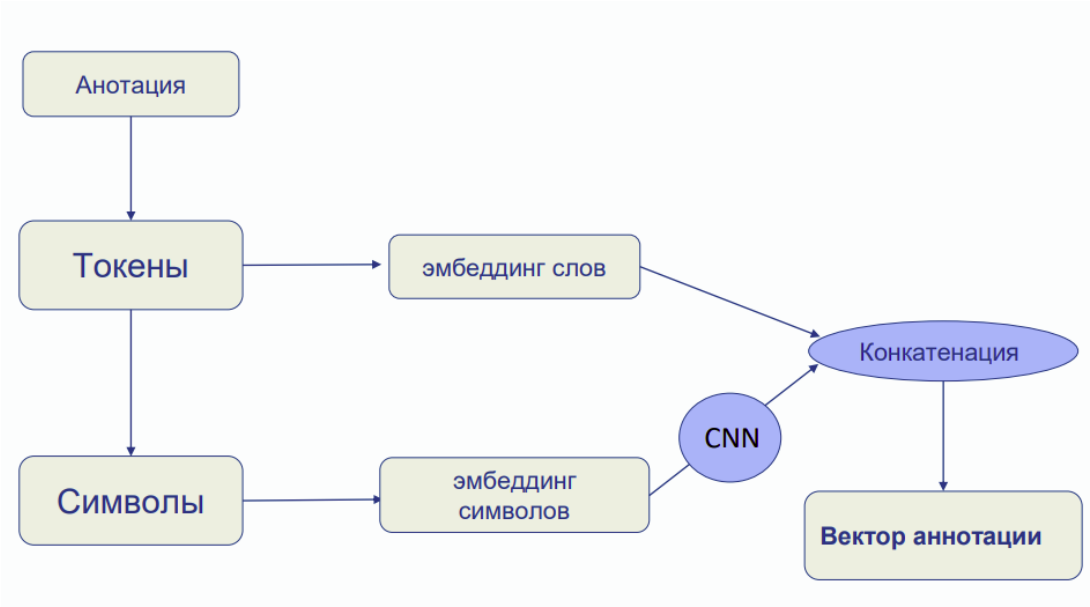


Рис. 4.6: Диаграмма построения вектор аннотации

Таким образом, обновление эмбеддинга аннотации позволяет модели более полно учесть как символьную, так и семантическую информацию в аннотациях, что может улучшить ее способность к правильной классификации и нормализации денежных сумм.

4.5 Добавление Attentions и CRF слой

После обновления эмбеддингов аннотации в модели, использовано следующая архитектура для классификации и нормализации денежных аннотаций:

1. BiLSTM: Входные эмбеддинги аннотаций передаются через слой Bidirectional LSTM (BiLSTM).
2. Self-Attention: После BiLSTM был добавлен слой само-внимания (self-attention). Self-attention механизм позволяет модели обрабатывать анно-

тации, учитывая взаимосвязи между различными токенами. Он позволяет модели фокусироваться на важных признаках внутри аннотации, учитывая их контекстуальные и семантические свойства.

3. CRF слой: Для моделирования зависимостей между метками классификации денежных аннотаций вы добавили слой условного случайного поля (CRF). CRF слой учитывает контекстуальные и последовательные свойства аннотаций, позволяя модели принимать более обоснованные решения при классификации. CRF слой помогает улучшить точность и последовательность предсказаний модели.

4. Выходной линейный слой: В конце модели находится выходной линейный слой, который предоставляет классификацию денежных аннотаций.

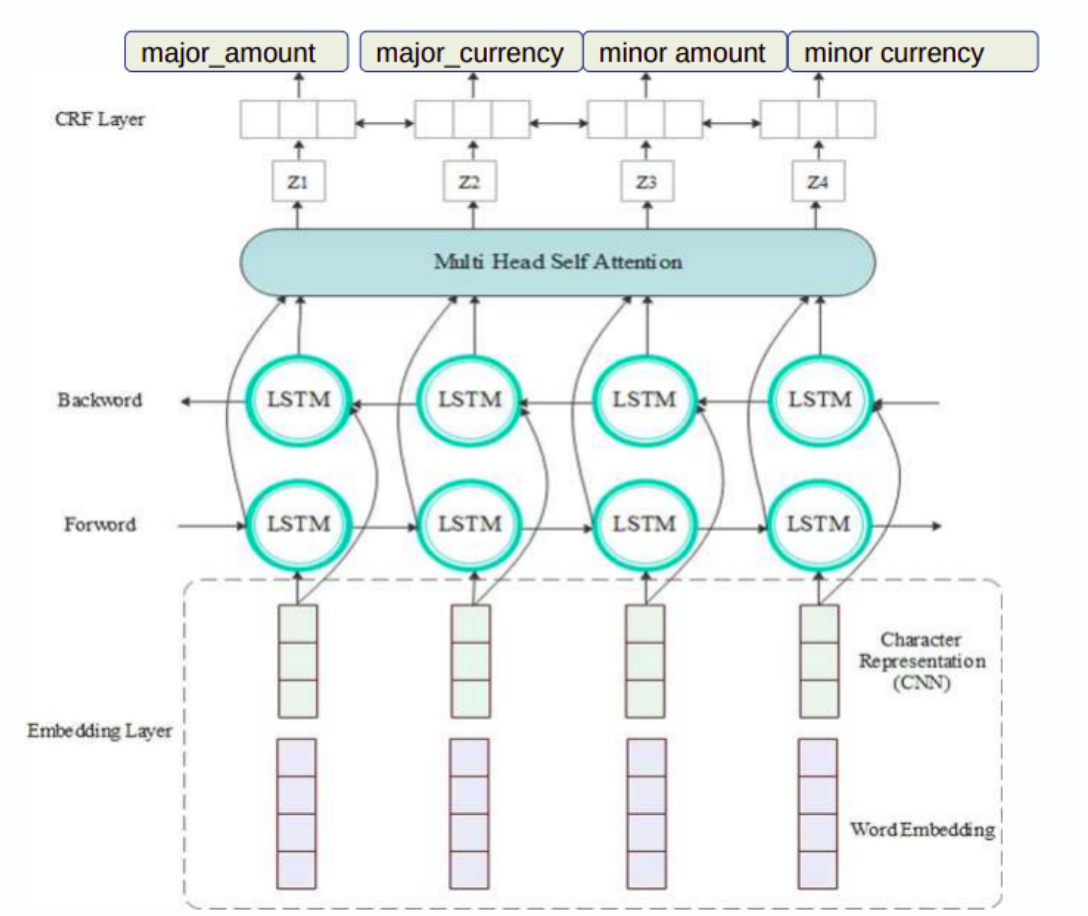
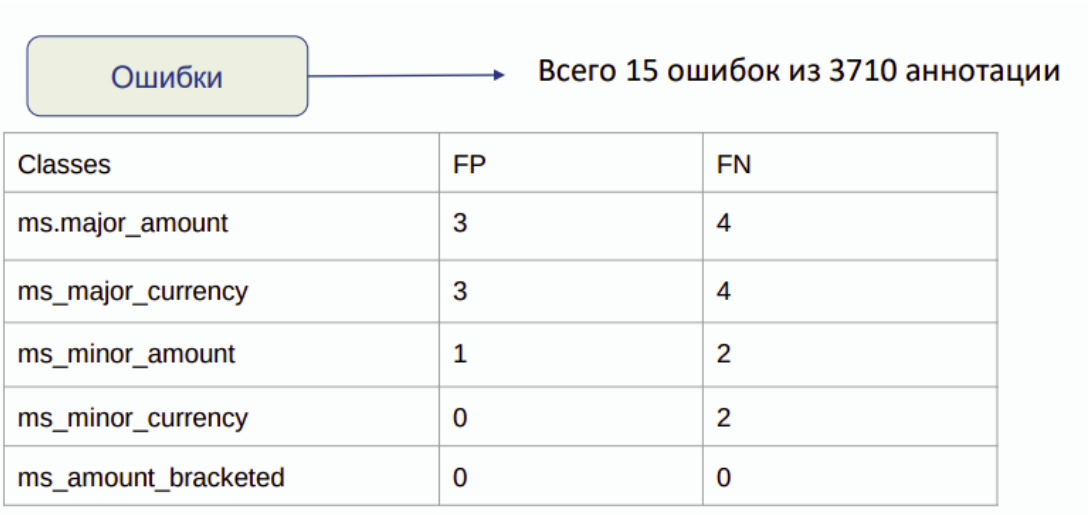


Рис. 4.7: BiLSTM+attention+CRF model

получаем такие метрики

	precision	recall	f1-score	support
ms_major_amount	1.00	1.00	1.00	5404
ms_major_currency	1.00	1.00	1.00	5056
ms_minor_amount	0.95	0.91	0.93	22
ms_minor_currency	1.00	0.87	0.93	15
ms_major_amount_bracketed	1.00	1.00	1.00	15
accuracy			1.00	10512
macro avg	0.99	0.95	0.97	10512
weighted avg	1.00	1.00	1.00	10512

Рис. 4.8: метрики качество BiLSTM+attention+CRF модели



Classes	FP	FN
ms.major_amount	3	4
ms_major_currency	3	4
ms_minor_amount	1	2
ms_minor_currency	0	2
ms_amount_bracketed	0	0

Рис. 4.9: Ошибки модели

На основе анализа метрик, можно заметить значительное улучшение распознание классов.

Добавление синтетических данных

При добавлении синтетических данных в эксперименте был использован тот же алгоритм, что и в предыдущем случае, с некоторыми изменениями в гиперпараметрах для создания случайных выборок.

Для генерации синтетических данных были изменены определенные гиперпараметры, такие как диапазон значений и распределение денежных сумм, а также вероятность введения случайных ошибок или шума в аннотации. Изменение этих гиперпараметров позволяет создавать разнообразные синтетические данные, варьи-

рующиеся по характеристикам и сложности.

При проведении эксперимента с моделью, основанной на описанной архитектуре, и увеличении количества синтетических данных до 500, было замечено, что количество ошибок на задаче классификации и нормализации денежных аннотаций снизилось до 13. Однако, значительных и резких изменений не было замечено.

	precision	recall	f1-score	support
ms_major_amount	1.00	1.00	1.00	5694
ms_major_currency	1.00	1.00	1.00	5226
ms_minor_amount	0.94	0.91	0.92	40
ms_minor_currency	1.00	0.92	0.96	27
ms_major_amount_bracketed	1.00	1.00	1.00	25
accuracy			1.00	11012
macro avg	0.99	0.96	0.97	11012
weighted avg	1.00	1.00	1.00	11012

Рис. 4.10: BiLSTM + attention + CRF + synthetic data

Глава 5

Заключение

В данном исследовании мы провели серию экспериментов с целью улучшения классификации денежных аннотаций. В этой главе мы обобщим основные результаты и сделаем выводы на основе полученных данных.

В первом эксперименте мы использовали модель, основанную на BiLSTM, для классификации. Затем мы решили улучшить результаты, добавив синтетические данные. Результаты этого эксперимента показали некоторое улучшение метрик классификации по сравнению с исходными значениями.
Из метрик Рис. 4.3 и Рис. 4.5

F1 - Score

Classes	Char BiLSTM	Char BiLSTM + synthetic data
ms.major_amount	1.00	1.00
ms_major_currency	1.00	1.00
ms_minor_amount	0.79	0.9
ms_minor_currency	0.82	0.9
ms_amount_bracketed	0.9	1.00

macro average	0.9	0.96

С добавлением синтетических данных мы наблюдали улучшение распознавания для классов "minor amount "minor currency"и "amount bracketed". Модель начала пред-

сказывать эти классы более точно и надежно. Это отразилось на значении макро-среднего (macro average), которое увеличилось с **0.9** до **0.96**. Таким образом, использование синтетических данных оказало положительное влияние на качество классификации и нормализации денежных аннотаций.

Во втором эксперименте мы решили улучшить модель, внесши изменения в ее архитектуру. Мы начали с изменения структуры построения эмбеддингов, добавили слои self-attention и CRF, чтобы модель могла лучше учитывать контекстуальные и последовательные свойства аннотаций. Результаты этого эксперимента превзошли результаты первого эксперимента, демонстрируя значительное улучшение метрик классификации и нормализации. Это указывает на то, что изменения в архитектуре модели позволили более точно улавливать зависимости и контекст аннотаций.

Затем мы провели дополнительный эксперимент, добавив синтетические данные в улучшенную модель. Результаты этого эксперимента показали незначительное, но все же положительное улучшение метрик. Это свидетельствует о том, что добавление синтетических данных дополнительно улучшает производительность модели.

Из метрик Рис. 4.8 и Рис. 4.10

F1 - Score

Classes	Char BiLSTM	Char BiLSTM + synthetic data	BiLSTM + ATTENTION + CRF	BiLSTM + ATTENTION + CRF + synthetic data
ms.major_amount	1.00	1.00	1.00	1.00
ms_major_currency	1.00	1.00	1.00	1.00
ms_minor_amount	0.79	0.9	0.93	0.92
ms_minor_currency	0.82	0.9	0.93	0.96
ms_amount_bracketed	0.9	1.00	1.00	1.00
macro average	0.9	0.96	0.972	0.976

С добавлением синтетических данных мы наблюдали улучшение распознавания для класс "minor amount". F1-мера для класса "minor currency"повысилась с 0.93 до 0.96, указывая на хорошее улучшение в точности и полноте предсказа-

ний. Для класса "minor amount"значение F-меры изменилось незначительно с 0.93 до 0.92. Однако, общее качество предсказаний модели улучшилось, что подтверждается повышением значения макро-среднего (macro average) с **0.972** до **0.976**.

В заключение, наши эксперименты позволили нам значительно улучшить наши результаты. В результате перехода от простой модели к более сложной архитектуре, мы достигли значительного улучшения результатов. Этот шаг в нашем исследовании привел к более точному и надежному предсказанию и классификации денежных аннотаций. Кроме того, внедрение генерации синтетических данных оказало существенное влияние на метрики нашей модели. Этот подход позволил нам значительно улучшить результаты и повысить эффективность работы нашей системы. Таким образом, наша работа демонстрирует важность использования сложных архитектур и генерация разнообразных данных для достижения хороших результатов в задачах классификации и нормализации денежных аннотаций.

Литература

[1] *Chiu, Jason PC.* Named entity recognition with bidirectional LSTM-CNNs / Jason PC Chiu, Eric Nichols // *Transactions of the Association for Computational Linguistics.* — 2016. — Vol. 4. — Pp. 357–370.

[2] Named-entity recognition for Indonesian language using bidirectional LSTM-CNNs / Wibowo Gunawan, Dwi Suhartono, Frans Purnomo, Arry Ongko // *Procedia Computer Science.* — 2018. — Vol. 135. — Pp. 425–432.

[3] *Ma, Xuezhe.* End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF / Xuezhe Ma, Eduard Hovy // Proceedings of the 54th annual meeting of the association for computational linguistics (Volume 1: Long papers). — Vol. 1. — 2016. — Pp. 1064–1074.

[4] *Huang, Zhiheng.* Bidirectional LSTM-CRF models for sequence tagging / Zhiheng Huang, Wei Xu, Kai Yu // *arXiv preprint arXiv:1508.01991.* — 2015.

[5] Neural architectures for named entity recognition / Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian et al. // *arXiv preprint arXiv:1603.01360.* — 2016.