М.Г. Карпов [1,2], Д.С. Лобунцов [1], И.Д. Чхайло [1], И.Г. Захарова [1,2]

[1] Тюменский государственный университет, г. Тюмень

[2] Научно-технический университет «Сириус», г.Сочи

УДК 04.85

ИЗВЛЕЧЕНИЕ СТРУКТУРИРОВАННОЙ ИНФОРМАЦИИ ИЗ ТЕКСТОВ ИСКОВЫХ ЗАЯВЛЕНИЙ

Аннотация. В работе исследуются методы извлечения структурированной информации (именованные сущности и факты) из текстов исковых заявлений. Проанализированы особенности обработки текстов на естественном языке и на их основе разработаны и реализованы алгоритмы извлечения информации. Разработаны архитектуры нейронных сетей для решения поставленных задач. Проведены вычислительные эксперименты, показавшие достаточно высокий уровень точности извлечения именованных сущностей и фактов.

Ключевые слова: машинное обучение, нейронные сети, извлечение фактов, обработка естественного языка, именованные сущности.

Введение

Любая организация в процессе своего функционирования генерирует большие объемы текстов, которые используют естественный язык (ЕЯ). В совокупности устные и письменные формы текстов на таком языке занимают доминирующую долю при взаимодействии субъектов, что делает их самыми многочисленными.

В сравнении с искусственными языками ЕЯ обладает отличиями, которые усложняют его обработку в информационных системах. Ключевой

особенностью ЕЯ является его высокий уровень избыточности, который позволяет облегчить понимание смыслов между участниками коммуникации в условиях неограниченного времени. Негативный эффект избыточности ЕЯ состоит в больших объемах передаваемого сообщения, который складывается из множественного дублирования передаваемой информации, традиционно используемых лексических связок и конструкций [1]. Информация в неизменном виде не может транслироваться в системах предприятия в условиях дефицита времени ввиду отсутствия необходимости в ее избыточности и в ее целостности. Отсюда для бизнеса возникает необходимость предварительной обработки такого текста – извлечения валидной информации, которая в дальнейшем будет участвовать в бизнес-процессах.

Цель данной работы состоит в разработке и исследовании алгоритмов интеллектуального анализа исковых заявлений для минимизации временных затрат на их первичную обработку на примере конкретной банковской организации.

Постановка задачи

Даны тексты исковых заявлений. На базе данных текстов необходимо:

- исследовать и реализовать алгоритмы обработки текстов документов;
- исследовать и реализовать алгоритмы выявления логических ошибок в рабочих текстах.

Результатом работы алгоритмов является преобразование неструктурированного текста исковых заявлений в структурированные данные.

Положим, что t – анализируемый текст; F – множество фактов (знаний) из предметной области иска; факт $f_i \in F$ имеет набор атрибутов $\{q_j\}$, которые описываются грамматическими характеристиками и

отношениями с другими атрибутами; $t_k \in t$ – токен, являющийся значением атрибута. Тогда формально факт можно описать следующим образом: $\langle f_i, q_j, t_k \rangle$.

Для именованных сущностей: t – анализируемый текст; N – множество именованных сущностей предметной области; именованная сущность $n_i \in N$ описывается грамматическими характеристиками; $t_k \in t$ – токен, являющийся значением именованной сущности. Тогда формально именованную сущность можно описать следующей парой: $\langle n_i, t_k \rangle$.

Описание данных

Был проведен анализ исследуемых данных, в ходе которого были выявлены следующие особенности.

В датасете общим объемом 9000 текстов 6330 текстов содержали заголовки, которые позволили их отнести к категории исковых заявлений. Содержание датасета представлено в таблице 1. Из них было выделено 85 типов исков, которые предметом возникших правоотношений. На рисунке 1 представлено 8 наиболее многочисленных типов текстов.

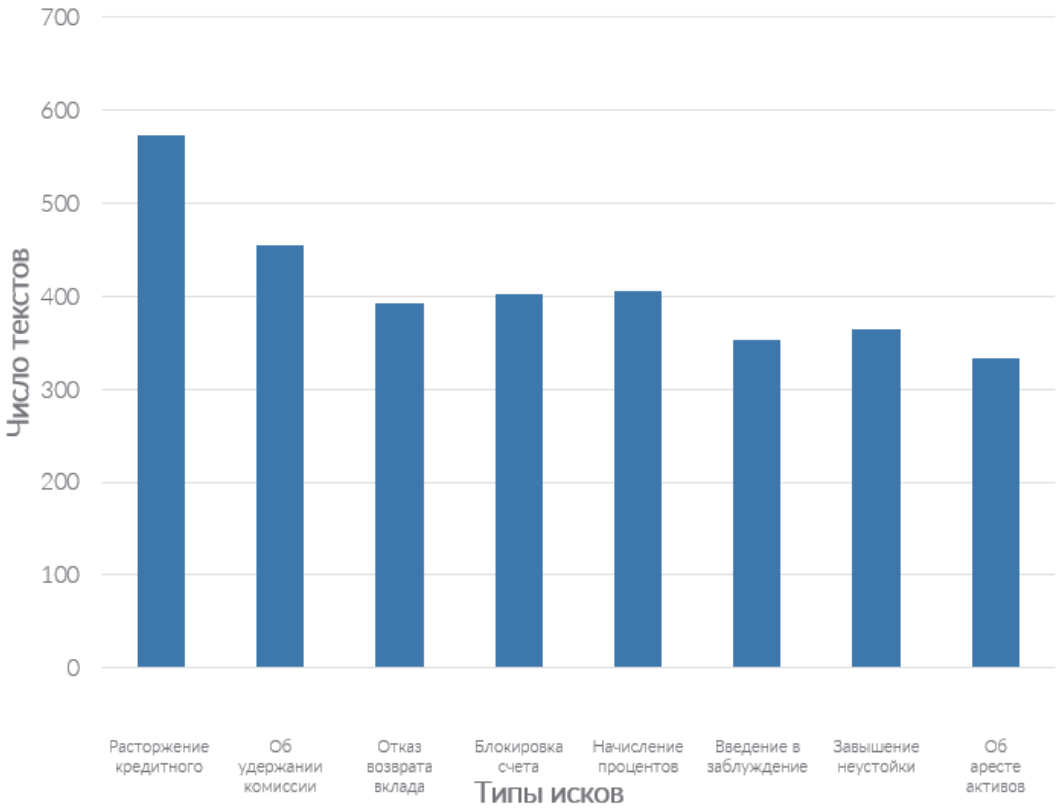

Рис. 1. Кол-во текстов исков по типам.

Таблица 1. Описание набора данных.

Параметр	**Значение**
Всего документов	9115
Типы документов	Иски Жалобы Ответные иски
Кол-во исков	6330
Кол-во жалоб	584
Кол-во ответных исков	2201
Кол-во типов исков	85

Результаты решения первой задачи

В поставленной перед нами задаче, необходимо извлекать из текста как классические именованные сущности, такие как имена, локации и организации, так и нестандартные. Среди нестандартных именованных сущностей есть такие, которые относятся ко всем заявлениям, например

ссылки на документы, законодательство РФ, уставы компаний и т. д., и такие, которые характерны только некоторым типам документов, например кадастровый номер.

Тестирование работы библиотеки на нашем наборе данных показало, что библиотека Natasha [2] хорошо справляется с извлечением классических именованных сущностей, для некоторых сущностей точность извлечения фиксируется меньше 0.42. В основном такая проблема возникает при извлечении адресов, в которых присутствует нестандартное написание. Пример такого неполного извлечения предоставлен на рисунке 2, информация об участке не извлечена, хотя является валидной частью адреса.

```
>>> address_extractor.find("село Успенка, ул. Спортивная, участок № 23")
Match(start=0, stop=28, fact=Addr(parts=[AddrPart(value='Успенка', type='село'), AddrPart(value='Спортивная', type='улица')]))
```

Рис. 2. Пример неполного извлечения информации библиотекой Natasha.

Для решения такого рода ситуаций, мы дополнили соответствующие грамматики путем изменения исходного кода библиотеки, так как Natasha не позволяет расширить грамматику изнутри. После внесения этих изменений мы получили достаточно хорошее качество извлечения классических именованных сущностей.

Среди требуемых к извлечению именованных сущностей встречаются строго регламентированные, например кадастровый номер недвижимости или ИНН. В таких случаях имеется возможность решить задачу с помощью использования регулярных выражений. Однако Natasha позволяет написать правила и для таких случаев. Мы воспользовались Natasha для того, чтобы извлеченная информация имела унифицированный вид в качестве объекта в программе.

Ссылки на нормативные акты, законодательные акты РФ и т. д. являются сущностями без заданной структуры. Однако исковые заявления написаны на контролируемом языке, что выражается в шаблонности

формулировок и некоторыми общими правилами написания. Благодаря этой особенности мы написали грамматику для извлечения именованных сущностей данного типа. Для написания правила мы составили словарь, который содержит перечисление основных законодательных актов и возможных ссылок внутри документа (раздел, ссылка и т. д.).

F-мера для написанной грамматики оказалась равной 0.596, что является недостаточно хорошим показателем. Поэтому было принято решение использовать подход с использованием машинного обучения.

Для обучения нейронной сети мы используем библиотеку Keras [3]. Для построения модели мы использовали архитектуру BiLSTM-CRF [4] со скрытым слоем TimeDistributed. На вход модели подаются последовательность векторных представлений слов, полученных из модели word2vec (из библиотеки gensim), на выходе ожидаются предсказанные теги для каждого токена.

- Bidirection-LSTM используется для предсказания вероятности каждого из возможных тегов для токена. Двунаправленность слоя позволяет учитывать контекст и до, и после токена.
- TimeDistributed это обертка, которая позволяет применить один слой (Dense) для каждого элемента из последовательности независимо. Используется для сохранения отношения один к одному в задачах классификации последовательности.
- На слое CRF сеть обучается ограничениям, которые существуют в выходном результате. Например, теги I из схемы BIOES могут идти только после тегов B. Без данного слоя результатом предсказания будет самый вероятный тег для каждого токена, что может привести к некорректным результатам.

На выходном слое используется функция активации ReLu. Данная функция активации имеет формулу $f(x) = \max(0, x)$, то есть реализует

простой пороговый переход в нуле. Из преимуществ данной функции можно выделить гораздо меньшую вычислительную сложность по сравнению с остальными функциями активации, что позволяет сократить время обучения модели. А также при применении ReLu скорость сходимости стохастического градиентного спуска гораздо выше [5].

Модель обучалась 19 эпох, батчами по 1000 документов. Процесс обучения занял около 22 минут. При обучении модель показывала accuracy 0.9317. На нашем наборе данных обученная модель показала f-меру 0.8671, что показывает прирост качества решения по сравнению с разработанной грамматикой более 30% и является решением довольно хорошего качества.

Результаты решения второй задачи

Учитывая обилие фактов и предметных областей в исследуемых текстах, было принято решение об использовании машинного обучения для извлечения фактов. Для применения данного метода необходимо произвести разметку текста, выделив в каждом тексте интересующие нас атрибуты.

Из формального описания задачи следует, что функция определения значения атрибута задается как t_k = *function_fact*$\langle f_i, q_j \rangle$. Очевидно, что ошибка в фактах определяется как *function_fact*$\langle f_i, q_j \rangle$. != *function_fact*$\langle f_i, q_j \rangle$.

Учитывая обилие фактов и предметных областей в исследуемых текстах, было принято решение об использовании машинного обучения для извлечения фактов. Для применения данного метода необходимо произвести разметку текста, выделив в каждом тексте интересующие нас атрибуты.

Например, для факта «факт оплаты услуг представителя» (см. Рис. 3) можно выделить следующие атрибуты:

- Субъект

- Сумма оплаты
- Подтверждение (квитанция, справка)

Согласно договору на оказание юридических услуг от 13.03.2013 года, заключенного между Сацута Р.Н. и Индивидуальным предпринимателем Шапоревой А.Р., Сацута Р.Н. оплатил услуги представителя в размере 20 000 рублей, что подтверждается квитанцией к приходному кассовому ордеру и справкой. Указанная сумма не является завышенной по данной категории дел.

Рис. 3. Фрагмент договора

Выбор таких атрибутов был произведен на основании анализа 80 текстов, которые используют данный факт. Если атрибуты Субъект и Сумма оплаты описываются грамматиками, то атрибут Подтверждение не обладает грамматическими признаками. Эта проблема была решена путем составления словаря значений атрибутов.

Для обучения нейронной сети была использована библиотека Keras [3]. На вход модели подаются последовательность векторных представлений слов, полученных из модели word2vec (из библиотеки gensim), на выходе ожидаются предсказанные теги для каждого токена.

Для построения модели мы использовали следующую архитектуру:

- Три входа, где первый вход (LSTM 256) используется для токена (вектор длиной 300), второй вход (LSTM 256) для контекста (длина 3 для токенов слева и справа), третий для контекста и токена (DENCE 512)
- Первый скрытый слой. Для первого входа DENSE 128. Для второго DENSE 128, для третьего DENSE 256.
- Второй скрытый слой. Производится конкатенация векторов.
- Третий скрытый слой. Используется архитектура DENSE 128.
- Четвертый скрытый слой. Используется архитектура DENSE 64.

- Пятый скрытый слой. Используется архитектура DENSE 32
- Классификация. Функция потери – кросс-энтропия.

Модель обучалась 16 эпох, батчами по 1000 документов. Процесс обучения занял около 62 минут. При обучении модель показывала accuracy 0.81. На нашем наборе данных обученная модель показала f-меру 0.79, что демонстрирует достаточную эффективность извлечения. Архитектура подобрана экспериментально.

В данной работе были предложены решения задач обработки естественного языка в исковых заявлениях. В частности, были протестированы различные методы решения задачи извлечения именованных сущностей, и реализован наиболее качественный. Также была предложена и реализована архитектура искусственной нейронной сети для решения задачи извлечения фактов.

Благодарности

Статья подготовлена в рамках разработки образовательного кейса для НТУ Сириус при финансовой поддержке РФФИ в рамках научного проекта № 19-37-51028.

СПИСОК ЛИТЕРАТУРЫ

1. Е. В. Грудева Избыточность языка и избыточность текста: некоторые размышления // Труды ИЛИ РАН. Том VI, часть 2. СПб., 2010
2. Github URL: https://github.com/natasha/natasha (дата обращения: 16.05.2020)
3. Keras URL: https://keras.io (дата обращения: 18.05.2020)
4. Lample G., Ballesteros M., Subramanian S., Kawakami K., Dyer C. Neural architectures for named entity recognition // arXiv preprint. arXiv, 2016
5. Krizhevsky A., Sutskever I., Hinton G. E. ImageNet Classification with Deep Convolutional Neural Networks // In NIPS. 2012.