УДК 343.140.02:004.891.2

Винокуров Михаил Андреевич
студент 2 курса
специальности «Безопасность информационных технологий в правоохранительной сфере»
Института кибербезопасности и цифровых технологий
РТУ МИРЭА
vinokurov.m.a@edu.mirea.ru

Научный руководитель: Ксенофонтов Николай Валерьевич
старший преподаватель КБ-4

Института кибербезопасности и цифровых технологий
РТУ МИРЭА
ksenofontov@mirea.ru

Нейронная сеть для анализа российской арбитражной судебной практики

Введение

Целью данной работы является создание нейронной сети для анализа российской арбитражной судебной практики по банкротствам. В наши дни количество арбитражных дел неуклонно растет, поэтому вопрос полуавтоматизации стоит весьма остро. К примеру, в сервисе «Картотека арбитражных дел»[1], по состоянию на 2024-06-07T13:30+03:00, в картотеке за период от 2024-06-06 до 2024-06-07 зарегистрировано 1808 банкротных дел.

Однако, стоит отметить, что до машинноисполняемого права ещё далеко ввиду динамичности и системности права. Судебное усмотрение в арбитражном процессе предполагает применение, в том числе, доктрин, часть из которых и сформировалась благодаря дискреционным полномочиям суда: доктрина нечистых рук, поднятая корпоративная вуаль, эстоппель (запрет на противоречивые действия), frustration of contract, contra proferentem (который нашёл явное отражение в пункте 11 Постановления Пленума Высшего Арбитражного Суда РФ от 14 марта 2014 г. N 16 «О свободе договора и ее пределах»). Добавим ещё механизмы вроде contra legem и судебного нормотворчества ad hoc, насчёт которых до сих пор идут споры, и получим ад для искуственного интеллекта. Стоит отметить, что судебное усмотрение для арбитражного является критически важным для выполнения задач, описанных в статье 2 АПК РФ.

А особенно важной, по мнению автора, является задача, описанная в пункте 3 статьи 2 АПК РФ:

Справедливое публичное судебное разбирательство в разумный срок независимым и беспристрасным судом.[2]

Выделенная задача опирается на этико-нравственные оценочные категории, к примеру, справедливость.

[1] Картотека арбитражных дел. 2020. URL: `https://kad.arbitr.ru/` (дата обр. 09.06.2024).

[2] Пункт 3 статьи 2 // «Арбитражный процессуальный кодекс Российской Федерации» от 24.07.2002 N 95-ФЗ (ред. от 29.05.2024). 04.07.2020.

1 Техническое требование

1.1 Функциональные требования

- Создание модели, обученной на судебной практике.
- Создание системы, позволяющей по запросу находить наиболее ревелатные дела.
- Борьба с «галлюцинациями», вызванными особенностями модели.

1.2 Нефункциональные требования

- Реализация на ЯП Java.
- Использование bindings for CUDA для ускорения обучения модели.
- CLI интерфейс.

1.3 Особенности реализации

- Снято требование «Размер используемого хипа не должен превышать 500мб » пункта 3 части 3 Требований по выполнению курсовой работы из-за особенностей проекта.

1.4 Необходимые ресурсы и технологии

- Java[3]
- Word2Vec[4]
- Java bindings for CUDA
- Судебная прак тика по банкротствам. (могут возникнуть проблемы из-за подпункта 1 пункта 5 статьи 15 Федерального закона от 22.12.2008 № 262-ФЗ (ред. от 14.07.2022) «Об обеспечении доступа к информации о деятельности судов в Российской Федерации»)

2 Описание выполнения задания

Начнём с изучения произвольного дела Арбитражного суда РФ. Рассмотрим дело А82-8071/2024. На данный момент там есть одно-единственное определение о принятии к производству заявления о признании гражданина банкротом и назначении судебного заседания. Как и все определения в картотеке, оно находится в PDF. Поскольку PDF является бинарным форматом, его надо перевести в более простой для обработки формат. Для этого воспользуемся классом PDFParser, исходный код которого расположится ниже:

```
1  package ru.inferno_geek.arbitrage_analysis;
2
3  import org.apache.pdfbox.pdmodel.PDDocument;
4  import org.apache.pdfbox.text.PDFTextStripper;
5
6  import java.io.File;
7  import java.io.IOException;
8  import java.nio.file.Path;
9  import java.nio.file.Files;
10
11 public class PDFParser {
12
```

[3] *Н. В. Ксенофонтов, И. А. Иванова, И. Д. Котилевец.* Кроссплатформенная среда исполнения программного обеспечения. РТУ МИРЭА, 2023. ISBN 978-5-7339-1693-4.

[4] DeepLearning4j. 2023. URL: `https://deeplearning4j.konduit.ai/` (дата обр. 09.06.2024).

```
    public void parsePDF(Path path) throws IOException {
        File file = new File(path.toString());
        PDDocument document = PDDocument.load(file);
        PDFTextStripper pdfTextStripper = new PDFTextStripper();
        String text = pdfTextStripper.getText(document);
        Path newTxt = Files.createFile(path.resolveSibling(path.getFileName() + ".txt"));
        Files.write(newTxt, text.getBytes());
        document.close();
    }
}
```

Листинг 1. Класс PDFParser

После перевода PDF в более простой формат мы можем воспользоваться классом Vectorizer для получения текстовых векторов для дальнейшего обучения нейронных сетей.

```
package ru.inferno_geek.arbitrage_analysis;

import java.io.File;
import java.nio.file.Path;
import java.nio.file.Files;
import java.nio.file.Paths;
import org.deeplearning4j.models.word2vec.Word2Vec;
import org.deeplearning4j.text.sentenceiterator.BasicLineIterator;
import org.deeplearning4j.text.sentenceiterator.SentenceIterator;
import org.deeplearning4j.text.tokenization.tokenizer.preprocessor.CommonPreprocessor;
import org.deeplearning4j.text.tokenization.tokenizerfactory.DefaultTokenizerFactory;
import org.deeplearning4j.text.tokenization.tokenizerfactory.TokenizerFactory;

public class Vectorizer {


    public static Path vectors = Paths.get("src/main/resources/vectors.txt");

    public static void vectorize(Path path) throws Exception {

        // Create file if it doesn't exist
        if(!vectors.toFile().exists()) {
        Files.createFile(vectors);
        }
        // Gets Path to Text file
        String filePath = path.toString();

        // Strip white space before and after for each line
        SentenceIterator iter = new BasicLineIterator(filePath);
        // Split on white spaces in the line to get words
        TokenizerFactory t = new DefaultTokenizerFactory();

        /*
            CommonPreprocessor will apply the following regex to each token: [\d\.:,"'\(\)\[\]|/?!;]+
            So, effectively all numbers, punctuation symbols and some special symbols are stripped off.
            Additionally it forces lower case for all tokens.
         */
        t.setTokenPreProcessor(new CommonPreprocessor());

```

```
        Word2Vec vec = new Word2Vec.Builder()
                .minWordFrequency(5)
                .iterations(1)
                .layerSize(100)
                .seed(42)
                .windowSize(5)
                .iterate(iter)
                .tokenizerFactory(t)
                .build();
        vec.fit();
        Files.write(vectors, vec.toJson().getBytes());
    }
}
```

Листинг 2. Класс Vectorizer

3 Благодарности

Ксенофонтов Николай Валерьевич за методологию BDSM без партнёра.
Котилевец Игорь Денисович за мотивацию и поддержку и полезные материалы.

4 GitHub

`https://github.com/Infernogeek1/court-practice-analysis-java`

Список использованных источников

1. Картотека арбитражных дел. — 2020. — URL: `https://kad.arbitr.ru/` (дата обр. 09.06.2024).
2. Пункт 3 статьи 2 // «Арбитражный процессуальный кодекс Российской Федерации» от 24.07.2002 N 95-ФЗ (ред. от 29.05.2024). — 04.07.2020.
3. *Н. В. Ксенофонтов, И. А. Иванова, И. Д. Котилевец.* Кроссплатформенная среда исполнения программного обеспечения. — РТУ МИРЭА, 2023. — ISBN 978-5-7339-1693-4.
4. DeepLearning4j. — 2023. — URL: `https://deeplearning4j.konduit.ai/` (дата обр. 09.06.2024).