Применение инструментов интеллектуального анализа текстов в юриспруденции

© Д.С. Зуев ©А.А. Марченко ©А.Ф. Хасьянов

Казанский (Приволжский) федеральный университет,
Казань, Россия

dzuev11@gmail.com anton.marchenko@kpfu.ru ak@it.kfu.ru

Аннотация. Описана архитектура системы интеллектуального анализа текстов в юриспруденции, способной на имеющейся базе данных судебных документов выявлять общие зависимости, предоставлять для ознакомления юридические дела, близкие по тематике, рекомендовать наиболее вероятные исходы судебного рассмотрения или помечать важные места, на которые следует обращать внимание при процессуальных действиях с использованием инструментов текстовой аналитики.

Ключевые слова: аналитика и управление данными, интенсивное использование данных, электронные библиотеки, кластеризация, рекомендательная система, микросервисная архитектура.

Text Mining Tools in Legal Documents

© D.S. Zuev ©A.A. Marchenko ©A.F. Khasiannov

Volga Region Federal University,
Kazan, Russia

dzuev11@gmail.com anton.marchenko@kpfu.ru ak@it.kfu.ru

Abstract. We present the architecture of the system for the intellectual textual analysis in jurisprudence based on microservices. The system can identify common dependencies on an existing database of legal documents, provide legal cases close to each other, familiarize them with the most probable outcomes of judicial review or mark out important places during procedural actions.

Keywords: analytics and data management, data intensive domains, digital libraries, clustering, recommender system, microservices.

1 Введение

Как известно, информационное общество характеризуется высоким уровнем развития информационно-коммуникационных технологий (ИКТ) и их интенсивным использованием всеми и всюду. В основе ИКТ лежит информация, а сами они во многом определяют содержание, масштабы и темпы развития других технологий.

Интересным направлением разработки специализированных автоматизированных информационных систем является создание интеллектуальных систем, способных на имеющейся базе данных судебных документов выявлять общие зависимости, предоставлять судьям для ознакомления близкие по тематике дела, рекомендовать наиболее вероятные исходы или помечать важные места, на которые судебным работникам следует обращать внимание при процессуальных действиях. Подобная система, на наш взгляд, поможет участникам судебного процесса точнее оценивать свои позиции или выбирать лучшую стратегию поведения, а судьям – в сжатые сроки формировать подборку связанных документов, не тратя для вынесения вердикта лишнего времени на поиск во всем архиве документов.

Проведенные исследования по семантическому структурированию информации в других предметных областях (см., например, [1, 2]), анализ инструментов текстовой аналитики (см, например, [3]) и наработки по применению семантических технологий при работе с юридическими документами [4] говорят о реализуемости поставленной задачи.

2 Интеллектуальная система «Робот-юрист»

2.1 Цели и задачи

«Робот-юрист» – это информационная система, которая должна позволять участникам юридического процесса правильно проводить подготовку дела, а также осуществлять планирование судебной деятельности. Эта система ориентирована на арбитражные суды, занимающиеся рассмотрением споров в сфере предпринимательства. В целом наш проект направлен на развитие российского правового государства, обеспечение доступности, открытости и прозрачности правосудия, формирование у граждан

Труды XIX Международной конференции «Аналитика и управление данными в областях с интенсивным использованием данных» (DAMDID/RCDL'2017), Москва, Россия, 10–13 октября 2017 года

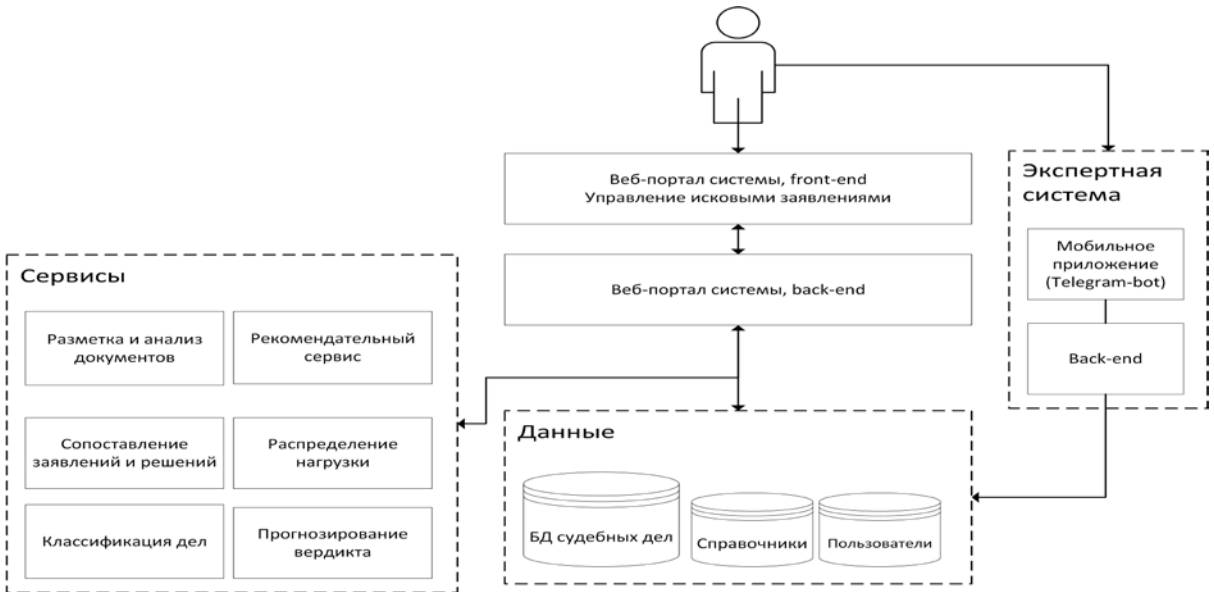

Рисунок 1 Архитектура системы

правосознания, основанного на верховенстве Права.

Задача системы – помочь определить характер спора, осуществить поиск и проверку действия правовых норм, регулирующих спорные правоотношения, оказывать содействие в установлении компетентного суда (подсудность, подведомственность), статуса участников спора (действующее, ликвидированное, банкрот), определении круга обстоятельств, имеющих значение для рассмотрения спора, характера спорного правоотношения, нормы права, подлежащей применению (действует ли данная норма), а также проверять достаточность и комплектность представляемых документов. Отдельными функциями планируются обеспечение возможности оформления искового заявления, а также вычисление (по предоставленным исходным данным на основе архива судебных дел) вероятности принятия того либо иного решения.

Для достижения поставленных целей были поставлены следующие задачи: создание портала для формирования шаблонов исковых заявлений с отслеживанием их жизненного цикла; разметка и анализ существующей базы судебных решений, исковых заявлений (классификация заявлений и решений, извлечение сущностей и фактов); подбор аналогичных дел и решений, рекомендательный сервис; сопоставление исковых заявлений и судебных решений; распределение судебных дел между судьями с учетом их специализации и текущей загрузки и прогнозирование вероятного решения по предоставленным исходным данным.

Каждая из выделенных задач является автономным модулем разрабатываемой информационной системы, а сама система – практическая демонстрация совместного использования ряда семантических технологий и инструментов текстовой аналитики.

2.2 Архитектура системы

Текущие парадигмы разработки предусматривают два концептуально различных подхода к дизайну приложений. Первый вариант – «монолитные приложения», когда вся логика по обработке запросов выполняется в рамках единственного процесса, при этом используются возможности конкретного языка программирования для разделения приложения на классы и функции. Однако любые изменения, даже самые небольшие, требуют перекомпиляции всего дистрибутива информационной системы и последующего обновления всех ее модулей. С течением времени изменения в логике работы одного модуля начинают влиять на функции других модулей.

Другой подход – это построение среды, в которой отдается предпочтение слабым связям, абстрагированию низкоуровневой логики, гибкости, а также возможности многократного использования и обнаружения компонентов [5, 6], сервис-ориентированная архитектура (Service-Oriented Architecture, SOA). Такая архитектура строится на сервисах, а не на приложениях. Сервисы – это дискретные программные компоненты, предоставляющие четко определенную функциональность и используемые в составе многих приложений. Каждый сервис представляет собой изолированную сущность с минимумом зависимостей от других совместно используемых ресурсов. Таким образом, возникает возможность изменять отдельные сервисы, не затрагивая при этом всю систему. Дальнейшим развитием парадигмы сервис-ориентированной архитектуры можно считать появление архитектуры микросервисов [7]. Термин «Microservice Architecture» получил распространение в последние несколько лет для описания способа проектирования приложений в виде набора независимо развертываемых сервисов.

Архитектурный стиль микросервисов – это подход, при котором единое приложение строится как набор небольших сервисов, каждый из которых работает в рамках собственного процесса и взаимо-

действует с остальными. Сервисы построены вокруг бизнес-потребностей и развертываются независимо с использованием полностью автоматизированной среды. Централизованное управление минимизировано, а сами сервисы могут быть написаны на разных языках программирования и использовать разные технологии хранения данных. Более того, внутри каждого микросервиса вполне может быть задействована собственная база данных (см. [7]).

С учетом достаточно большого количества модулей системы наиболее логичным путем для создания «Робота-юриста» стало применение архитектуры микросервисов.

Архитектура системы приведена на Рис. 1. Нами выделено несколько групп сервисов, взаимодействующих между собой с помощью программного интерфейса (API). Каждый из них реализует одну из соответствующих функциональных задач. На схеме выделены серверная и клиентская часть веб-портала системы, а также слой доступа к данным – база данных судебных дел и решений, нормативно-справочная информация. В виде отдельного модуля разрабатывается экспертная система, в автоматическом режиме оказывающая консультации по юридическим вопросам в формате взаимодействия с виртуальным собеседником – Telegram-Ботом.

2.3 Разметка массива документов

Разметка существующего массива документов необходима для дальнейшего обучения сервисов системы. Для реализации этой задачи использовался инструмент для быстрого структурированного аннотирования текстов BRAT [8]. BRAT – это веб-система с открытым исходным кодом, разработанная группой разработчиков в университетах Токио и Манчестера. Результаты разметки получаются в виде, удобном для дальнейшей машинной программной обработки.

Судебные решения и дела открыты и доступны для просмотра в интернете и представляют собой массив неразмеченных документов, в котором ориентироваться непросто. Важна собственно разметка текстов судебных дел для выделения классов и подклассов сущностей, их зависимостей с целью дальнейшего построения модели машинного обучения.

На текущий момент времени, в рамках создания прототипа системы, принято решение о первоначальной разметке сравнительно небольшого количества документов (около 3000). Важно отметить, что самих типов споров, значит, и классов связанных документов может быть достаточно много. С целью упрощения работы на начальном этапе мы обрабатывали судебные дела, относящиеся только к нескольким категориям судебных споров.

Размеченный текст будет использоваться для обучения подсистемы поиска аналогов и прогнозирования вердикта по делу. В качестве результата работы получаем размеченный текст, который записывается в БД судебных дел для дальнейшей обработки.

Для первоначальной разметки были выделены основные сущности, такие, как «Истец», «Ответчик», «Предмет спора», «Действующие нормы». На текущий момент времени определено 56 сущностей, которые необходимо выделять внутри судебных решений для дальнейшей обработки. Множество выделенных сущностей будет уточняться по мере увеличения объема размеченного текста. На сегодняшний день проведена всего лишь первая итерация данного процесса.

2.4 Рекомендательный сервис

Одной из важнейших задач формируемой информационной системы являются поиск и предоставление аналогичных решений по схожим судебным искам. Таким образом, необходим сервис поиска аналогичных документов, или рекомендательный сервис.

Существуют два основных типа рекомендательных систем: контент-ориентированные и социальные (коллаборативной фильтрации) (см., например, [9]). Первые основаны на представлении предпочтений пользователей путем анализа содержимого рекомендательных элементов. Системы второго типа моделируют предпочтения, оценивая близость профилей пользователей. Ниже под рекомендательным сервисом будем понимать информационную систему, которая: 1) формирует модель предметной области на основе массива документов (включая подготовительные операции – приведение к векторному виду, кластеризацию и т. п.); 2) получает на вход документ и выдает список документов, близких к входному.

По сравнению с поисковыми системами рекомендательные системы наиболее полезны, когда у пользователя возникают трудности с формулировкой эффективного поискового запроса.

Подходы к организации рекомендательных сервисов могут быть разными, в [1] описан подход с использованием онтологий и предпочтений пользователей. Учитывая специфику предметной области и разрабатываемой системы, использовать предпочтения пользователей не корректно.

Алгоритм работы сервиса можно разделить на два этапа. На подготовительном этапе обрабатываются все имеющиеся документы: вырезаются знаки пунктуации, термы приводятся к единому виду (для слов с разными окончаниями и суффиксами). Далее документ приводится к векторному виду. Для представления массива документов в виде числовых векторов, отражающих важность использования каждого слова из некоторого набора слов (количество слов набора определяет размерность вектора), в каждом документе используется мера TF-IDF [3, 10]. На основе массива векторов происходит кластеризация.

На первом шаге необходимо определить количество K кластеров, мы использовали для этого формулу $K = N_{doc}/10$, где N_{doc} – общее количество об-

рабатываемых документов. Далее производится собственно кластерный анализ по методу *K-means* (метод К-средних, [3, 11]). Полученные результаты сохраняются для дальнейшего использования.

На основном этапе работы на вход сервису подается идентификатор документа. Производится приведение его к векторной форме, которая обрабатывается моделью, причисляется к определенному кластеру. На выходе алгоритм выдает первые *n* документов из того же кластера, что и входной документ, количество выдаваемых документов настраивается, на данном этапе реализации системы *n* определено равным 10.

Процесс переобучения модели следует проводить периодически, например, раз в сутки, либо после существенного изменения всего корпуса документов.

Обработка массива из 3250 документов занимает 5 мин (Intel® Core™ i7-3632QM CPU @ 2.20GHz × 8), что на текущем этапе развития системы «Робот-юрист» является приемлемым показателем быстродействия. Сервис реализован на языке Python, взаимодействие с другими модулями системы происходит по внутреннему согласованному протоколу взаимодействия.

2.5 Классификация судебных дел

Одной из проблем судебного делопроизводства является процедура определения категории и характера спора. Правильное определение категории судебного спора важно, поскольку влияет на назначение судьи на соответствующий процесс, а назначаемый судья должен иметь максимальный опыт рассмотрения подобных споров. На текущий момент выявлено около 60 различных категорий судебных споров, которые встречаются с разной частотой. За определение категории судебного дела отвечает модуль классификации судебных дел. Процесс классификации с ростом количества обрабатываемых документов может быть очень затратным по времени, поэтому с архитектурной точки зрения было решено вынести данную функциональность как отдельный микросервис с реализацией обмена с другими модулями системы в асинхронном режиме. К тому же определение категории спора (судебного дела) не является задачей, требующей мгновенного ответа.

На уровне межсервисного взаимодействия общий алгоритм обработки документа выглядит следующим образом: на вход подается идентификатор документа; из документа выделяются ключевые слова и их количество; проводятся анализ и подбор класса дела; алгоритм возвращает идентификатор класса судебного дела, который становится дополнительным свойством документа. При добавлении нового класса проводятся анализ допустимых ключевых слов и повторное обучение нейронной сети.

К сожалению, на текущий момент нами окончательно не выбран оптимальный способ реализации данной задачи – рассматривается реализация алгоритма с использованием глубинного обучения и сверточных нейронных сетей или с использованием латентно-семантического анализа.

2.6 Создание шаблонов исковых заявлений

Отдельной задачей является сопоставление судебных актов и заявлений по рассмотренным делам, поскольку сами исковые заявления, в отличие от базы знаний принятых решений, являются закрытыми и не публикуются в сети интернет. В рамках разработки системы «Робот-юрист» актуальной является задача связывания вновь поданного искового заявления и близких результатов судебных процессов для дальнейшей обработки. В этом случае необходимо иметь заявление в размеченном виде, удобном для машинной обработки. Для этого необходимо либо отдельно предусматривать процесс разметки массива электронных копий бумажных исковых заявлений, либо формировать заявления изначально в электронном виде и далее распечатывать готовое заявление с помощью системы. Второй вариант является предпочтительным, и его было предложено реализовать в рамках создания прототипа системы.

Для получения экземпляров исковых заявлений сразу в электронном виде был предложен механизм веб-портала – шаблонизатора заявлений. При подаче пользователем системы искового заявления система формирует печатную версию заявления в соответствии с регламентирующими нормативными документами РФ, а электронная копия документа автоматически размечается и сохраняется в базе данных системы с определенным статусом.

Процесс организован следующим образом: пользователь авторизуется на портале системы; ему предоставляется ряд экранных форм с полями ввода для заполнения данных. После окончания ввода данных пользователь сохраняет заявление в системе; в базе данных системы появляется размеченный вариант документа для дальнейшего анализа, а пользователю предоставляется печатная форма заполненного искового заявления.

Веб-портал предусматривает несколько ролей пользователей с различной функциональностью, также предложена и реализована статусная модель судебного дела для удобства отслеживания жизненного цикла документа в системе.

2.7 Экспертная система

В рамках проекта также разрабатывается решение по автоматизации предоставления экспертных консультаций по вопросам юридического характера. Решение представляет собой экспертную систему (ЭС) (см., например, [12]) – компьютерную систему, способную частично заменить эксперта-специалиста в разрешении какой-либо проблемы юридического характера.

В рамках проекта реализована экспертная система в области защиты интеллектуальной собственности. Важными вопросами в автоматизации предоставления экспертных консультаций являются надежность решений и удобство использования,

поэтому решения ЭС подкрепляются ссылками на соответствующие нормативные документы, указанные юристами при формировании базы знаний.

Были определены наиболее часто встречающиеся сценарии и вопросы в данной области права. На текущий момент реализованы 13 типовых сценариев поведения ЭС, которые практически полностью покрывают всевозможные случаи в данной области права.

В качестве пользовательского интерфейса к экспертной системе был выбран интерфейс чат-бота или, другими словами, виртуального собеседника, реализованного в виде Telegram-Бота (далее – бота). Совпадение логики процессов взаимодействия с ботом и ЭС позволяет предоставить удобный доступ к инструментам юридического консультирования со всех платформ, для которых доступен сам мессенджер (Telegram). Логика работы модуля представляет собой конечный автомат, а использование бота в качестве интерфейса к ЭС позволяет снизить трудозатраты на разработку пользовательского интерфейса и сконцентрироваться на функционале ЭС вследствие простоты разработки.

3 Заключение

Теоретические исследования в рамках текстовой аналитики показывают наличие готовых или практически готовых инструментов для реализации функций отдельных модулей системы. Необходимы лишь их грамотное объединение и применение в отдельно взятых предметных областях. «Робот-юрист» должен стать именно такой демонстрацией применения известных подходов и алгоритмов в юриспруденции.

На данный момент завершен первый этап создания системы – закончено проектирование системы и реализован прототип системы «Робот-юрист», производится разметка документов. Для успешного завершения работ и перевода в опытную эксплуатацию требуется дальнейшая оптимизация как различных алгоритмов текстовой аналитики, так и пользовательского интерфейса. Выбранная архитектура построения приложения позволяет производить модификацию отдельных модулей системы, не затрагивая общего механизма взаимодействия. Также необходимы апробация инструментов системы на большем массиве документов и рефакторинг программного кода.

Поддержка

Работа выполнена за счет средств субсидии, выделенной Казанскому федеральному университету для выполнения государственного задания в сфере научной деятельности, проект 2.8712.2017/БЧ.

Литература

[1] Елизаров, А. М., Жижченко, А. Б. Жильцов, Н. Г., Кириллович А. В., Липачёв, Е. К.: Онтологии математического знания и рекомендательная система для коллекций физико-математических документов. Докл. Академии наук, 467 (4), с. 392-395 (2016). doi: 10.1134/S1064562416020174

[2] Елизаров, А. М., Липачёв, Е. К., Невзорова О. А., Соловьев, В. Д.: Методы и средства семантического структурирования электронных математических документов. Докл. Академии наук, 457 (6), с. 642-645 (2014). doi 10.7868/S0869565214240049

[3] Ингерсолл, Грант С., Мортон, Томас С., Фэррис, Эндрю Л.: Обработка неструктурированных текстов. Поиск, организация и манипулирование / Пер. с англ. Слинкин А. А. М.: ДМК Пресс, 414 с.: ил. (2015)

[4] Peroni, S.: SemanticWeb Technologies and Legal Scholarly Publishing Law, Springer, Governance and Technology Series, 15 (2014). doi 10.1007/978-3-319-04777-5

[5] Gold, N. et al.: Understanding Service Oriented Software. IEEE Software, 21 (2), pp. 71-77 (2004)

[6] Jones, S.: Toward an Acceptable Definition of Service. IEEE Software, 22 (3), pp. 87-93 (2005)

[7] Fowler, M.: Microservices a definition of this new architectural term. https://martinfowler.com/ articles/microservices.html

[8] Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., Tsujii, J.: Brat: a Web-based Tool for NLP-Assisted Text Annotation. Proc. of the Demonstrations Session at EACL (2012)

[9] Ricci, F., Rokach, L., Shapira, B., Kantor, P. B.: Recommender Systems Handbook. N.Y.: Springer (2011)

[10] https://ru.wikipedia.org/wiki/TF-IDF

[11] https://ru.wikipedia.org/wiki/K-means

[12] Джарратано, Дж., Райли, Г.: Экспертные системы. Принципы разработки и программирование. 4-е издание. Вильямс, 1152 с. (2007)