УДК 006.015, 004.82 DOI: 10.17587/it.27.322-330

Е. А. Гостева, e-mail: kategosteva@yandex.ru, **В. В. Ланин**, ст. преподаватель, e-mail: vlanin@live.com,
Национальный исследовательский университет "Высшая школа экономики", г. Пермь

Реализация системы интеллектуального поиска информации в промышленных стандартах

Статья посвящена разработке системы интеллектуального поиска информации по промышленным стандартам. Представлено описание предметной области, проектирование и основные стадии разработки системы: создание модуля извлечения информации из промышленных стандартов, реализация веб-сервиса с помощью фреймворка Flask и клиентского веб-приложения на React JS. Использование данной системы инженерами и разработчиками программного обеспечения позволит эффективно управлять базой определений промышленных стандартов, правильно их интерпретировать и использовать в соответствии с выбранной областью знаний.

Ключевые слова: *промышленные стандарты, умное производство, интеллектуальный поиск*

Введение

Извлечение информации из текстов и представление ее в виде формальной системы знаний — важная задача в области автоматической обработки текстов на естественном языке. Постоянно растущий объем текстов и документов, которые находятся в свободном доступе, с одной стороны, расширяет границы для изучения различных областей знаний, с другой — затрудняет процесс поиска и выделения значимой информации. Особенно это важно при использовании стандартов в промышленности, где нужно быстро оперировать данными для ускорения процесса производства, выявления новых трендов и имеющихся наработок, уменьшения потери прибыли из-за недостаточности информации. Основная проблема на данный момент заключается в невозможности изучить всю содержащуюся в различных документах информацию за достаточно непродолжительное время. В процессе ее решения появилось направление "information extraction", задача которого заключается в извлечении структурированных данных из слабоструктурированных или неструктурированных документов для облегчения процесса обработки и анализа информации. Задачи из этой области актуальны в производстве, экономике и научной деятельности [1—5].

В рамках данной работы объектом исследования являются промышленные стандарты, предметом — системы интеллектуального поиска в промышленных стандартах. Цель — разработка системы интеллектуального поиска информации в промышленных стандартах. Для достижения поставленной цели необходимо проанализировать подходы и инструментальные средства для извлечения информации из документов, описать функциональное назначение системы, реализовать модуль извлечения информации, разработать веб-сервис и клиентское приложение для работы с ним.

1. Анализ задачи интеллектуального поиска по стандартам

Современной компании необходимо эффективно использовать имеющуюся информацию, уметь применять ее на практике при разработке продуктов, развитии сферы услуг, создании потребительской ценности. Большая роль в данных процессах отводится управлению знаниями и непрерывному обучению, поэтому лидерами новой экономики стали компании, которые научились быстрее своих конкурентов находить и внедрять существующие знания в свое производство.

"Корпоративные знания рассматриваются менеджментом как один из важнейших активов или как интеллектуальный капитал компании, а управление знаниями — как одно из эффективных направлений управления бизнесом" [6]. Формирование системы управления знаниями позволяет не только поддерживать конкурентоспособность, но и повысить скорость реакции на изменения внешней среды, а также избежать

проблемы недополученной прибыли. Когда все сотрудники говорят "на одном языке", систематизация, получение, классифицирование и передача знаний выходит на более высокий уровень, сокращается время включения сотрудников и принятия решений.

Корпоративные знания имеют несколько категорий — знания о бизнес-процессах (основная производственная информация), корпоративной культуре (взаимодействие с заказчиками, клиентами, коллегами), внешней среде компании (информация о покупателях и конкурентах, знания о продукте), личные знания (компетенции сотрудников). В рамках данной работы рассматриваются источники знаний о бизнес-процессах компании — стандарты. Стандарт — это документ, который содержит требования, правила и руководящие указания для процесса, продукта или услуги, необходим для достижения оптимальной степени согласованности и упорядоченности в заданной области знаний. Для производства продукта или услуги стандарты полезны определением оптимальных параметров, описанием методов оценки соответствия определенному уровню качества и условий их использования [7]. Главным преимуществом использования стандартов является передача технологий. Поскольку стандарты включают результаты достижений в какой-либо области науки, они отражают современное состояние в области технического развития. На данный момент существует несколько сотен промышленных стандартов, среди которых для разработки системы были выбраны стандарты проекта oneM2M в области Интернета вещей [8].

Проект oneM2M — это глобальное партнерство, основанное в 2012 г. ведущими организациями по разработке стандартов в области телекоммуникаций. Цель oneM2M заключается в разработке технических стандартов для обеспечения совместимости в отношении архитектуры, спецификаций API, решений по обеспечению безопасности и регистрации межмашинных технологий и технологий Интернета вещей на основе требований. Опубликованные стандарты позволяют экосистеме поддерживать широкий спектр приложений и услуг, например, Умный город, Умный дом, интеллектуальный транспорт, телематика, общественная безопасность и здравоохранение.

Структура представленных стандартов в большинстве случаев схожа: имеется титульный лист с информацией о документе (номер, название, дата и назначение стандарта), раздел с описанием области применения, раздел со ссылками на другие источники, раздел с основными определениями и сокращениями, далее следуют разделы, содержащие соответствующую назначению стандарта информацию (рис. 1). Все стандарты представлены на сайте oneM2M в виде "PDF" документов [9].

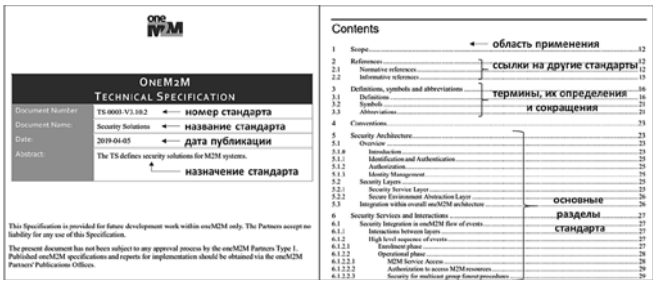


Рис. 1. Пример структуры стандарта oneM2M

2. Анализ задачи извлечения информации из текстов

Задача по извлечению знаний из неструктурированных источников на естественном языке относится к информационному поиску, предполагающему нахождение релевантной информации. Как правило, такие источники не обладают ни одним видом идентификации искомой информации, например, разметкой или метаданными, поэтому извлеченные данные в первую очередь записываются в формальном виде (в виде таблицы или базы данных), что позволяет впоследствии более эффективно их обрабатывать и применять, например, строить различные модели или представлять их в виде связного графа.

К видам извлекаемой информации относятся: именованные сущности, атрибуты объектов, отношения между объектами, факты и события, термины предметной области и их связи, ключевые слова документа, отзывы и мнения о товарах и услугах.

Для решения задачи извлечения информации из тестов используют три подхода: основанный на правилах, основанный на машинном обучении и гибридный, объединяющий два предыдущих.

Первый подход основан на том, что извлекаемая информация имеет определенные языковые конструкции, которые вручную описываются в виде шаблонов и правил их обработки и применяются к тексту, пока не будут найдены все соответствия шаблону. Например, из предложения "Traditionally, the term management refers to the activities (and often the group of people) involved in the four general functions listed below" с помощью правила "ЕСЛИ за словами the term, The term следует существительное ТО извлечь слово с меткой термин" будет извлечен термин "management".

Формальное описание языковой конструкции информации, которую необходимо найти в тексте, называется лексико-синтаксическим шаблоном. Шаблон может быть составлен с помощью регулярных выражений, например, один из вариантов поиска термина будет выглядеть как "[The|the] term NN", где "[The|the]" — варианты написания артикля (в начале или в середине предложения), необходимо наличие одного варианта из квадратных скобок, "term" — обязательное слово, "NN" — указывает на словосуществительное.

Формированием шаблонов и правил занимаются эксперты в данной области знаний, что позволяет достигать высокой точности извлечения информации. Для записи паттернов часто применяются специальные языки и поддерживающие их системы, примеры некоторых из них представлены далее.

Ограничением данного подхода являются тексты общих предметных областей, описание шаблонов для которых становится уже достаточно трудоемким, так как необходимо учесть всевозможные правила и языковые конструкции, а также смену шаблонов при изменении предметной области.

Достоинства данного подхода заключаются в более точных результатах в узких предметных областях (как в данной работе — стандарты в области Интернета вещей) с небольшим разнообразием языковых конструкций (стандарты написаны на английском языке, который имеет строго установленный и соблюдаемый порядок слов в предложении), а также данный подход учитывает особенности заданных слов — регистр букв и их последовательность [10].

Работа второго подхода к извлечению информации может быть основана на методах обучения с учителем, методе обучения без учителя, методе частичного обучения с учителем (бутстрап). Чаще всего проводят обучение с учителем, для этого строят модель, которая сможет отличить искомые данные, а затем обучают ее на выборке, у которой уже имеются вручную расставленные метки для проверки работы модели по критериям точности и полноты [11, 12]. При этом применяют лингвистические ресурсы для классификации сущностей и атрибутов, например, Википедию — проверяют, является ли данное слово определением в данной энциклопедии [13].

В качестве основы модели могут служить деревья принятия решений, логистическая регрессия, скрытые марковские модели, метод опорных векторов, нейронные сети [14, 15]. Подход дает возможность тестировать различные стратегии обучения, учитывается большое число признаков разного вида, а также не требуется привлечение лингвиста для написания шаблонов. Ограничения у данного подхода также имеются: для корректной работы размеченный корпус текстов стандартов (под корпусом понимается множество текстов, соответствующее определенным требованиям) должен иметь достаточно большой объем и высокое качество разметки, при смене метода может потребоваться обучение на новой выборке, необходимо тщательно настраивать выбранную модель для получения более точных результатов, результаты работы обычно плохо объяснимы и исправить возникающие ошибки практически невозможно.

3. Описание шаблонов определений в английском языке

Для написания лексико-синтаксических шаблонов необходимо вначале уточнить понятие "термин", а также сформировать все возможные языковые правила его построения в предложениях.

Термин — слово или словосочетание, обозначающее определенное понятие какой-либо области знаний. Определение — формулировка, разъясняющая содержание и смысл термина. Формальное определение состоит из трех частей: термин — слово или фраза, подлежащая определению; класс объекта или понятия, к которому относится термин; отличительные признаки, отличающие его от всех остальных представителей данного класса.

Самый распространенный способ написать определение — это использовать определительное придаточное предложение, где часть сложного предложения зависит от главного. Ниже представлены примеры конструкций таких предложений, где "()" обозначают необязательную часть, "/" — возможные варианты:

1. (a/an/the) (adjective) noun is/are (a/an/the) (adjective/ adjective and adjective) noun + of + noun/adjective/v-ing.
2. (a/an/the) (adjective) noun is (a/an/the) noun (which/that/where/when/who) + (adverb) past participle + (preposition).
3. (a/an/the) (adjective) noun is/are/ may/can be (adverb) defined/known as / called (a/an/the) noun of noun (which/that where/when/who).
4. (a/an/the) noun + is/are + (adverb) past participle + preposition.
5. (a/an/the) noun/pronoun + can be (adverb) + past participle + preposition.

Согласно Манчестерскому банку академических фраз определения могут быть описаны и с помощью конструкций, представленных на рис. 2 [16].

1. The term 'X' was first used by ...
2. The term 'X' can be traced back to ...
3. Previous studies mostly defined 'X' as ...
4. The term 'X' was introduced by Smith in her ...
5. Historically, the term 'X' has been used to describe ...
6. The term 'X' refers to ...
7. 'X' can broadly be defined as ...
8. 'X' can be loosely described as ...
9. The term 'X' encompasses A), B), and C).
10. 'X' can be defined as ... It encompasses ...

Рис. 2. Примеры конструкций для описания термина

Для проверки работоспособности системы были найдены предложения, построенные по описанным выше конструкциям, например, "Comic books are sequential and narrative publications consisting of illustrations, captions, dialogue balloons, and often focus on super-powered heroes. Astronomy is a branch of scientific study primarily concerned with celestial objects inside and outside of the earth's atmosphere. Most metals are malleable; they can be hammered into flat sheets; nonmetals lack this quality" и другие, некоторые из них не содержат терминов, а только соответствуют представленным конструкциям.

4. Инструменты построения систем извлечения информации

Для определения наиболее подходящего инструмента определены следующие критерии и их веса: наличие визуализации (возможность визуального представления присутствует — 10, возможность визуального представления отсутствует — 0), формы визуализации (чем больше, тем лучше), точность извлечения информации (чем больше подготовленных определений было найдено, тем лучше), сложность составления шаблонов (чем меньше среднее время на подготовку каждого шаблона, тем лучше).

Для сравнения предлагается использовать метод вариантных секторов, поэтому для каждого свойства были назначены следующие коэффициенты: наличие визуализации — 6, формы визуализации — 5, точность извлечения информации — 10, сложность составления шаблонов — 8.

Одним из популярных инструментов анализа текстов является программное средство GATE (General Architecture for Text Engineering), которое представляет собой систему обработки естественного языка, включающую в себя различные методы для обработки текстов, например, анализаторы морфологии и синтаксиса, поисковые инструменты и средства извлечения информации. Данную систему можно использовать в процессе извлечения, анализа и аннотирования информации, для анализа кореферентности, применения методов машинного обучения и онтологий, также имеется возможность добавлять другие плагины.

В ходе применения GATE были использованы два инструмента: система искусственного интеллекта ANNIE (A Nearly-New Information Extraction System) и основанная на машинном обучении библиотека OpenNLP, но ни один из них не показал точных результатов: не все предложения с терминами были найдены, некоторые из которых были не выделены из-за неправильного определения части речи, а также были выделены конструкции, не являющиеся определениями. Еще одним ограничением системы при нахождении определений является то, что определение не найдется, если между хотя бы какими-нибудь частями правила стоит не пробел, а, например, перенос строки [17].

Следующее программное средство, предназначенное для обработки и анализа текстов, — Stanford CoreNLP. Данное программное обеспечение написано на Java, но в результате работы сторонних разработчиков над продуктом появилась возможность работать с Python, Ruby, Perl, Javascript, F# и другими .NET языками. В ядро проекта входит также Stanford Parser (вывод зависимостей и построения дерева), Stanford POS Tagger (определение части речи), Stanford RegexNER (для извлечения именованных сущностей), Stanford EnglishTokenizer (разделение на токены), Stanford TokensRegex (для определения лексико-синтаксических шаблонов) и другие. Имеется также визуальный интерфейс, в котором можно посмотреть работу модулей по определению частей речи (рис. 3), распознаванию именованных сущностей, кореферентность, основные зависимости и другое [18].

Для извлечения информации можно использовать как регулярные выражения, так и именованные переменные. Выводить результаты помимо визуального представления можно

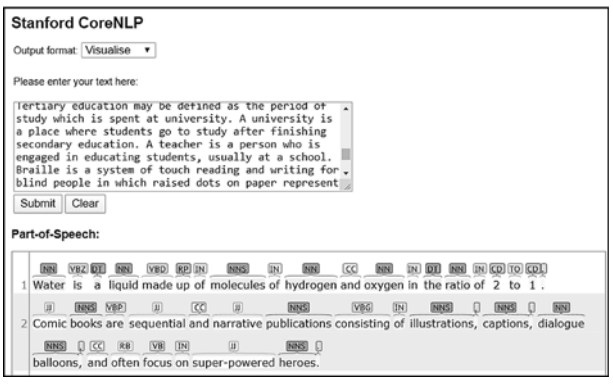


Рис. 3. Результат работы POS Tagging в Stanford CoreNLP

в форматы строки, XML, JSON, CoNLL. К ограничениям данного инструмента можно отнести некоторые неполадки при работе с основным API, однако это можно исправить, использовав другую библиотеку на основе Stanford CoreNLP.

NLTK (Natural Language Toolkit) — открытая платформа для создания программ по обработке естественного языка, написанная на языке Python, имеющая легкие в использовании интерфейсы для многих языковых корпусов, а также библиотеки для обработки текстов для классификации, токенизации, стемминга, разметки, фильтрации и семантических рассуждений. Для поиска информации составляется шаблон за счет последовательности символов (регулярное выражение). К преимуществам данного решения можно отнести подробную документацию, простую реализацию, не требуется подключение к серверу, как в предыдущем решении. К ограничениям — иногда возникают ошибки при разделении текста на предложения и определении части речи [19].

Также было проведено сравнение данных инструментов по определенным ранее критериям с помощью метода вариантных секторов, результаты которого представлены в таблице.

5. Требования к функциональным характеристикам

Система должна обеспечивать графический интерфейс в виде веб-страницы, на которой имеется возможность ввода поискового запроса, отображения документов и связей между ними. Также должны иметься возможности построения запросов по содержанию загруженных в систему стандартов, а также визуализации сети связанных стандартов в виде графа. Результаты работы система предоставляет в виде таблицы с информацией о данном термине или стандарте, графа связей стандартов, отображает документ стандарта. Система должна иметь возможность экспорта результатов поиска в формат "PDF".

Функциональным назначением разрабатываемой системы является автоматизация процесса извлечения знаний из документов промышленных стандартов, в том числе повышение соответствия полученной информации информационной потребности пользователя. Для описания функционального назначения системы применяется диаграмма вариантов использования (рис. 4).

Для отображения архитектуры системы используется диаграмма компонентов, представленная на рис. 5.

Разрабатываемая система состоит из трех компонентов: сервера баз данных, сервиса и web-приложения. В качестве сервера баз данных используется SQLite — кроссплатформенная база данных, которую можно создавать в Python без установки дополнительных инструментов. Отличительные особенности данного решения: быстрота, автономность, надежность, обеспечение полной функциональности, отсутствие ограничений на использование.

Сервис реализован с помощью фреймворка Flask (Python), который в отличие от других решений, например, Django, является простым в использовании, позволяет выбирать, с какой именно базой данных работать, и как именно будут взаимодействовать компоненты.

Рис. 4. Диаграмма прецедентов системы интеллектуального поиска

Результаты сравнения инструментальных средств

Инструменты	Свойство (вес)				
	Наличие визуализации (6)	Формы визуализации (5)	Точность извлечения (10)	Сложность составления шаблонов (8)	Общий вес инструмента
GATE (ANNIE)	10	5	6	5	185
GATE (OpenNLP)	10	5	7	5	195
Stanford CoreNLP	10	10	8	8	254
NLTK	0	0	7	7	126

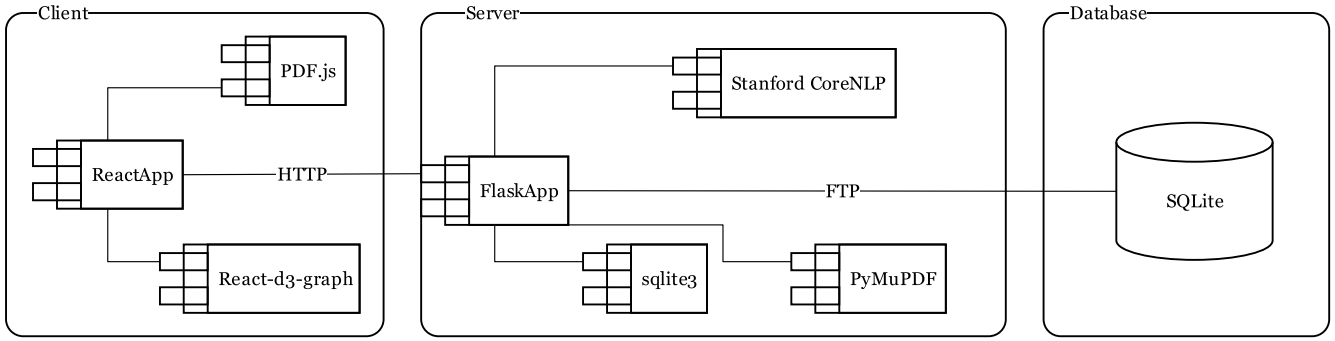


Рис. 5. Диаграмма компонентов разрабатываемой системы

Клиентское web-приложение написано на React JS — библиотека JavaScript для создания пользовательских интерфейсов, отличительными особенностями которой является разработка с помощью небольших и изолированных частей кода — компонентов, упрощенная отладка приложений за счет использования односторонней привязки данных, использование чистых функций, а не сложных и требующих настройки шаблонных классов для формирования интерфейса.

6. Разработка модуля извлечения информации из стандартов

В качестве входных данных на вход модулю поступают стандарты в формате "PDF", поэтому для работы с документами была выбрана библиотека "PyMuPDF", которая является облегченным средством просмотра PDF.

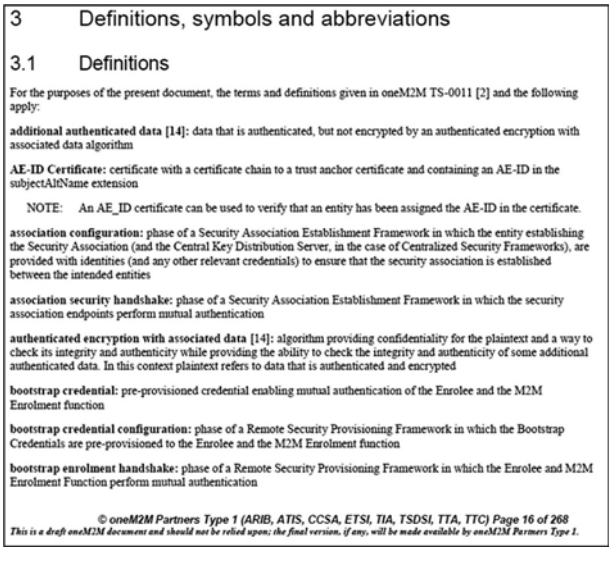
Из каждого стандарта должна быть извлечена следующая информация: название стандарта, номер документа, определения, отсылки к другим стандартам. Информация о документе находится на первой странице и ее можно найти с помощью слов "Document Number" and "Document Name".

Что касается определений, то здесь есть два варианта: это определения, которые вынесены в главу "Definitions", и определения, которые необходимо найти с помощью составленных ранее шаблонов из текста стандарта.

Для работы с уже выделенными в отдельный параграф (пример контента параграфа представлен на рис. 6) терминами была создана функция "extract_definitions_paragraph", которая получает на вход документ стандарта. Для того чтобы извлечь данные определения, необходимо загрузить только интересующие нас страницы, для чего понадобится оглавление ("Contents") — здесь необходимо найти номера страниц третьей главы. Затем данные страницы объединяются, и из них нужно удалить лишнюю информацию, например, знак копирайта, возможные комментарии, номер страницы. Необходимая нам информация представлена в формате "термин: определение". Однако под такой же шаблон попадают некоторые вводные предложения, например, строка "For the purposes of the present document, the terms and definitions given in oneM2M TS-0011 [2] and the following apply: additional authenticated data", поэтому необходимо их удалить, а затем разбить текст по абзацам.

С помощью выбранной библиотеки Stanford CoreNLP и описанных ранее шаблонов начинаем извлекать определения из оставшейся части документа. Для того чтобы найти определения, необходимо разбить весь документ на предложения и определить части речи каждого слова.

Следующий шаг — описание шаблонов с помощью регулярных выражений и поиск определений в тексте с их помощью. В Python для работы с регулярными выражениями есть модуль "re". Метод, который используется для решения данной задачи, называется "re.search(pattern, string)", он осуществляет поиск паттерна по всей строке



3 Definitions, symbols and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in oneM2M TS-0011 [2] and the following apply:

additional authenticated data [14]: data that is authenticated, but not encrypted by an authenticated encryption with associated data algorithm

AE-ID Certificate: certificate with a certificate chain to a trust anchor certificate and containing an AE-ID in the subjectAltName extension

NOTE: An AE_ID certificate can be used to verify that an entity has been assigned the AE-ID in the certificate.

association configuration: phase of a Security Association Establishment Framework in which the entity establishing the Security Association (and the Central Key Distribution Server, in the case of Centralized Security Frameworks), are provided with identities (and any other relevant credentials) to ensure that the security association is established between the intended entities

association security handshake: phase of a Security Association Establishment Framework in which the security association endpoints perform mutual authentication

authenticated encryption with associated data [14]: algorithm providing confidentiality for the plaintext and a way to check its integrity and authenticity while providing the ability to check the integrity and authenticity of some additional authenticated data. In this context plaintext refers to data that is authenticated and encrypted

bootstrap credential: pre-provisioned credential enabling mutual authentication of the Enrolee and the M2M Enrolment function

bootstrap credential configuration: phase of a Remote Security Provisioning Framework in which the Bootstrap Credentials are pre-provisioned to the Enrolee and the M2M Enrolment function

bootstrap enrolment handshake: phase of a Remote Security Provisioning Framework in which the Enrolee and M2M Enrolment Function perform mutual authentication

Рис. 6. Глава стандарта с определениями

(не только в начале). В качестве паттерна будут задаваться закодированные с помощью специальных символов определенные ранее шаблоны, пару из которых рассмотрим для примера.

Рассмотрим шаблон "(a/an/the) (adjective) noun is/are (a/an/the) (adjective/ adjective and adjective) noun + of + noun/adjective/v-ing". Здесь в начале может идти артикль, но не обязательно, на языке регулярных выражений это выглядит следующим образом: "(A|An|The|a|an|the)_DT\s)?". Далее, также необязательно, может идти прилагательное "(\w + _(JJ|JJR|JJS)\s)?", затем идут существительное и один из глаголов "is/are": "\w + _(NN|NNP|NNS)\s(is|are)_VBZ\s". Потом снова добавляем вариативную часть с артиклем: "((a|an|the)_DT\s)?". Далее идет прилагательное, прилагательное и прилагательное, либо ничего, описываем это так: "((\w + _(JJ|JJR|JJS)\s)|((\w + _(JJ|JJR|JJS))\sand_CC\s(\w + _(JJ|JJR|JJS)))\s)?". Далее идет существительное с предлогом "of" и один из вариантов: существительное, прилагательное или герундий, после чего могут быть еще слова, но не обязательно: "\w + _(NN|NNP|NNS)\sof_IN\s\w + _(NN|NNP|NNS|JJ|JJR|JJS|VBG)\w*". Таким образом описываются и другие выделенные шаблоны определений и в результате получаем таблицу, как показано на рис. 7.

Еще одним видом извлекаемой информации являются отсылки к другим стандартам. В данном случае не нужно определять части речи, достаточно задать шаблон для определения номера стандарта, который в общем виде выглядит следующим образом: "\bT(S|R)-\d\d\d\d\b", где "\b" — граница слова, "T" — сокращение от слова "Technical", далее может идти либо "S", что обозначает "Specifications", либо "R" — "Reports", далее обязательно пишется дефис("-") и 4 цифры ("\d"). Используя модуль "re", находим все упоминания других стандартов oneM2M и получаем таблицу, показанную на рис. 8.

Последним шагом в модуле извлечения информации стало выделение найденных определений в документах стандартов с помощью параметра "page.addHighlightAnnot()" (рис. 9).

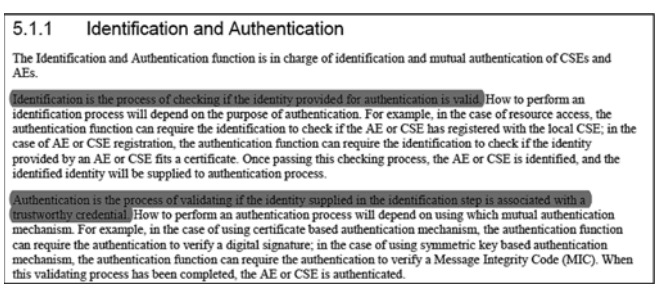
Данный модуль возвращает две таблицы — с терминами и определениями и с отсылками к другим стандартам, которые в дальнейшем будут записаны в базу данных, а также сохраняет документы с выделенными в них терминами и определениями, которые впоследствии выводятся на экран.

	doc_num	doc_name	term	definition
0	TS-0001-V3.15.1	Functional Architecture	access control attributes	set of parameters of the Originator, target r...
1	TS-0001-V3.15.1	Functional Architecture	access decision	authorization reached when an entity's Privil...
2	TS-0001-V3.15.1	Functional Architecture	application layer	comprises oneM2M Applications and related bus...
3	TS-0001-V3.15.1	Functional Architecture	attribute	stores information pertaining to the resource...
4	TS-0001-V3.15.1	Functional Architecture	child resource	sub-resource of another resource that is its ...

Рис. 7. Часть таблицы с определениями

	doc_num	docs_link
0	TS-0001	[TS-0011, TS-0003, TS-0004, TS-0012, TS-0021, ...
1	TS-0002	[TS-0011, TR-0008]
2	TS-0003	[TR-0008, TR-0019, TR-0013, TS-0001, TS-0011, ...
3	TS-0004	[TS-0001, TS-0003, TS-0008, TS-0009, TS-0010, ...

Рис. 8. Часть таблицы с отсылками на другие стандарты


5.1.1 Identification and Authentication

The Identification and Authentication function is in charge of identification and mutual authentication of CSEs and AEs.

Identification is the process of checking if the identity provided for authentication is valid. How to perform an identification process will depend on the purpose of authentication. For example, in the case of resource access, the authentication function can require the identification to check if the AE or CSE has registered with the local CSE; in the case of AE or CSE registration, the authentication function can require the identification to check if the identity provided by an AE or CSE fits a certificate. Once passing this checking process, the AE or CSE is identified, and the identified identity will be supplied to authentication process.

Authentication is the process of validating if the identity supplied in the identification step is associated with a trustworthy credential. How to perform an authentication process will depend on using which mutual authentication mechanism. For example, in the case of using certificate based authentication mechanism, the authentication function can require the authentication to verify a digital signature; in the case of using symmetric key based authentication mechanism, the authentication function can require the authentication to verify a Message Integrity Code (MIC). When this validating process has been completed, the AE or CSE is authenticated.

Рис. 9. Маркировка определений

7. Демонстрация результатов работы системы

Для отображения файла в формате PDF используется библиотека "PDF.js", которая преобразовывает файлы из формата PDF в код HTML5 используя метод рендеринга на клиенте, его преимуществами являются малая нагрузка на сервер и рендер PDF в разных расширениях без потери качества

Также необходимо добавить отображение связей между стандартами. Для создания графа применяется библиотека "react-d3-graph", которая позволяет управлять элементами и устанавливать настройки отображения [20]. На рис. 10 представлен результат поиска стандарта "TS-0003".

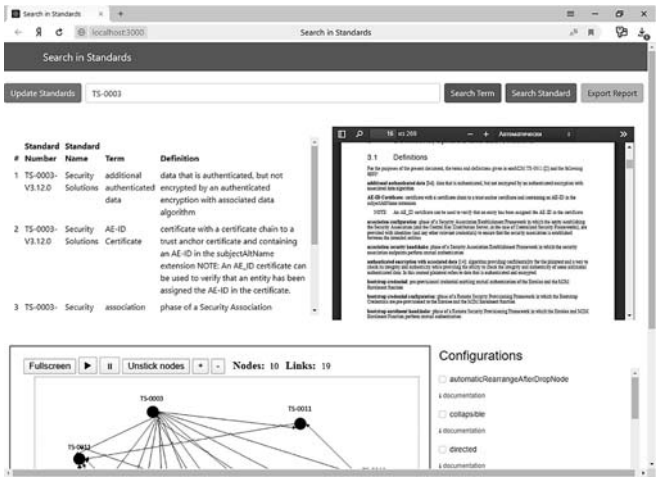


Рис. 10. Результаты поиска стандарта "TS-0003"

Заключение

В рамках данной работы была рассмотрена предметная область, связанная с извлечением информации, что является ключевым этапом в построении сложных систем информационного поиска. Как показал анализ предметной области, это направление довольно неплохо развито и предлагает большой выбор различных методов и инструментальных средств для построения систем извлечения информации. Одним из таких инструментов является Stanford CoreNLP, позволяющий эффективно решать базовые задачи обработки естественного языка, на основе которого и был разработан модуль извлечения информации.

Результатом работы стала реализация системы извлечения информации из промышленных стандартов с помощью языков Python (вебсервис на фреймворке Flask) и React JS (клиентское веб-приложение на основе React), которая отображает контент стандартов в виде таблицы с терминами и определениями, графа со связями между стандартами и сам стандарт в виде файла формата PDF. Применение данной системы позволит ускорить обработку извлеченной информации, что является важной частью в процессе накопления и применения новых знаний. Данное решение является универсальным и может быть использовано в различных областях.

В рамках дальнейшей работы планируется построение сложных поисковых запросов, взаимодействие с другими форматами документов, выгрузка графа связей в документ, а также его доработка: добавление связей терминов и обеспечение drill-down свойств при выборе конкретного термина.

Список литературы

1. **Mannai M., Abdessalem W., Chezala H.** Information Extraction Approaches: A Survey // Information and Communication Technology. Advances in Intelligent Systems and Computing. 2018. Vol. 625. P. 289—297.
2. **Muawia E., Ali A., Mubarak H.** Information Extraction Methods and Extraction Techniques in the Chemical Document's Contents: Survey // ARPN Journal of Engineering and Applied Sciences. 2015. Vol. 10. P. 1068—1073.
3. **Milosevic N., Gregson C., Hernandez R., Nenadic G. A.** Framework for Information Extraction from Tables in Biomedical Literature // International Journal on Document Analysis and Recognition. 2019. Vol. 22. P. 55—78.
4. **Issertial L., Tsuji H.** Information Extraction for Call for Paper // Natural Language Processing: Concepts, Methodologies, Tools, and Applications. 2020. P. 394—409.
5. **Suominen H., Zhou L.** Information Extraction to Improve Standard Compliance // AI 2015: Advances in Artificial Intelligence. 2015. Vol. 9457. P. 644—649.
6. **Lee J.** The Method for Calculating Lost Profit Damages Caused by Patent Infringement in the U. S. Patent Law — Focused on Lost Profit Damages of Lost Sales and Price Erosion // Yonsei Law Reviw. 2018. Vol. 28. P. 223—270.
7. **Hannah M., Leiva C, Noller D.** The Importance of Standards in Smart Manufacturing // MESA International White Paper. 2018. Vol. 58.
8. **Gezer C., Taskin E.** An Overview of oneM2M Standard // 2016 24th Signal Processing and Communication Application Conference (SIU). 2016. P. 1705—1708.
9. **Standards** for M2M and the Internet of Things. URL: http://www.onem2m.org/ (дата обращения: 13.02.2020).
10. **Большакова Е. И., Воронцов К. В., Ефремова Н. Э., Клышинский Э. С., Лукашевич Н. В., Сапин А. С.** Автоматическая обработка текстов на естественном языке и анализ данных: Учеб. пособ. М.: Изд-во НИУ ВШЭ, 2017. С. 83—122.
11. **Bovi C. D., Telesca L., Navigli R.** Large-scale Information Extraction from Textual Definitions through Deep Syntactic and Semantic Analysis // Transactions of the Association for Computational Linguistics. 2015. Vol. 3. P. 529—543.
12. **Espinosa-Anke L., Saggion H., Ronzano F.** Hypernym Extraction: Combining Machine Learning and Dependency Grammar // Lecture Notes in Computer Science. 2015. Vol. 1. P. 372—383.
13. **Астраханцев Н. А.** Автоматическое извлечение терминов из коллекции текстов предметной области с помощью Википедии // Труды ИСП РАН. 2014. № 26 (4). С. 7—20.
14. **Bird S., Klein E., Loper E.** Natural Language Processing with Python. California: O'Reilly Media. 2016. P. 221—257.
15. **Espinosa-Anke L., Carlini R., Saggion H., Ronzano F.** DefExt: A Semi Supervised Definition Extraction Tool // GLOBALEX 2016: Lexicographic Resources for Human language Technology Workshop. 2016. P. 24—28.
16. **Defining** Terms — Academic Phrasebank. URL: http://www.phrasebank.manchester.ac.uk/writing-definitions/ (дата обращения: 18.02.2020).
17. **General Architecture for Text Engineering (GATE).** URL: https://gate.ac.uk/ (дата обращения: 18.02.2020).
18. **Manning C. D., Surdeanu M., Bauer J.** The Stanford CoreNLP Natural Language Processing Toolkit // 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations. 2014. P. 55—60.
19. **Natural** Language Toolkit. URL: https://www.nltk.org/ (дата обращения: 20.02.2020).
20. **React-d3-graph.** URL: https://www.npmjs.com/package/react-d3-graph (дата обращения: 21.04.2020).

E. A. Gosteva, e-mail: kategosteva@yandex.ru, **V. V. Lanin,** Senior Lecture, e-mail: vlanin@live.com,
National Research University Higher School of Economics, Perm, Russian Federation

System Development for Intelligent Search in Industrial Standards

The article is devoted to the description of intelligent information retrieval system development according to industry standards based on the Stanford CoreNLP tool. The description of the subject area, design, and main stages of system development are presented: development of extracting information module from industrial standards, implementation of a web service using the Flask framework, and a client web application on React JS. The use of the developed system by engineers and software developers will make it possible to effectively manage the definition base of industrial standards, understand them correctly and observe them in accordance with the chosen field of knowledge.

Keywords: *industrial standards, smart manufacturing, intelligent search*

DOI: 10.17587/it.27.322-330

References

1. **Mannai M., Abdessalem W., Chezala H.** Information Extraction Approaches: A Survey, *Information and Communication Technology. Advances in Intelligent Systems and Computing,* 2018, vol. 625, pp. 289—297.

2. **Muawia E., Ali A., Mubarak H.** Information Extraction Methods and Extraction Techniques in the Chemical Document's Contents: Survey, *ARPN Journal of Engineering and Applied Sciences,* 2015, vol. 10, pp. 1068—1073.

3. **Milosevic N., Gregson C., Hernandez R., Nenadic G. A.** Framework for Information Extraction from Tables in Biomedical Literature, *International Journal on Document Analysis and Recognition*, 2019, vol. 22, pp. 55—78.

4. **Issertial L., Tsuji H.** Information Extraction for Call for Paper, *Natural Language Processing: Concepts, Methodologies, Tools, and Applications*, 2020, pp. 394—409.

5. **Suominen H., Zhou L.** Information Extraction to Improve Standard Compliance, *AI 2015: Advances in Artificial Intelligence*, 2015, vol. 9457, pp. 644—649.

6. **Lee J.** The Method for Calculating Lost Profit Damages Caused by Patent Infringement in the U. S. Patent Law — Focused on Lost Profit Damages of Lost Sales and Price Erosion, *Yonsei Law Reviw*, 2018, vol. 28, pp. 223—270.

7. **Hannah M., Leiva C, Noller D.** The Importance of Standards in Smart Manufacturing, *MESA International White Paper*, 2018, vol. 58.

8. **Gezer C., Taskin E.** An Overview of oneM2M Standard, *2016 24th Signal Processing and Communication Application Conference (SIU)*, 2016, pp. 1705—1708.

9. **Standards** for M2M and the Internet of Things, available at: http://www.onem2m.org/ (date of access: 13.02.2020).

10. **Bolshakova E. I., Vorontsov K. V., Efremova N. E., Klyshinsky E. S., Lukashavich N. V., Sapin A. S.** Automatic processing of text in natural language and data analysis, Moscow, The HSE Publishing House,. 2017, pp. 83—122 (in Russian).

11. **Bovi C. D., Telesca L., Navigli R.** Large-scale Information Extraction from Textual Definitions through Deep Syntactic and Semantic Analysis, *Transactions of the Association for Computational Linguistics*, 2015, vol. 3, pp. 529—543.

12. **Espinosa-Anke L., Saggion H., Ronzano F.** Hypernym Extraction: Combining Machine Learning and Dependency Grammar, *Lecture Notes in Computer Science*, 2015, vol. 1, pp. 372—383.

13. **Astrachantsev N. A.** Automatic term acquisition from domain-specific text collection by using Wikipedia, *Proceedings of the Institute for System Programming of RAS,* 2014, no. 26 (4), pp. 7—20 (in Russian).

14. **Bird S., Klein E., Loper E.** Natural Language Processing with Python, California, O'Reilly Media, 2016, pp. 221 -257

15. **Espinosa-Anke L., Carlini R., Saggion H., Ronzano F.** DefExt: A Semi Supervised Definition Extraction Tool, *GLOBALEX 2016: Lexicographic Resources for Human language Technology Workshop*, 2016, pp. 24—28.

16. **Defining** Terms — Academic Phrasebank, available at: http://www.phrasebank.manchester.ac.uk/writing-definitions/ (date of access: 18.02.2020).

17. **General** Architecture for Text Engineering (GATE), available at: https://gate.ac.uk/ (date of access: 18.02.2020).

18. **Manning C. D., Surdeanu M., Bauer J.** The Stanford CoreNLP Natural Language Processing Toolkit, *52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2014, pp. 55—60.

19. **Natural** Language Toolkit, available at: https://www.nltk.org/ (date of access: 20.02.2020).

20. **React-d3-graph.** available at: https://www.npmjs.com/package/react-d3-graph (date of access: 21.04.2020).