Современные подходы к механизмам извлечения причинно-следственных связей из неструктурированных текстов на естественном языке

Т.С. Умаров, И.Ю. Баженова

***Аннотация*— Статья посвящена исследованию методов извлечения причинно-следственных связей из неструктурированного текста на естественном языке. Автоматическое извлечение причинно-следственных связей из текстов на естественном языке является сложной открытой проблемой в области искусственного интеллекта. Описываются способы выражения явных причинно-следственных связей, применяемых в технологиях извлечения информации. В статье дается краткий обзор современных инструментальных средств поддержки интеллектуального анализа данных, позволяющих выполнять классификацию данных, использовать методы статистического анализа, средства кластеризации и сегментации, применять инструменты визуализации, а также пакеты для анализа текстов (Text Mining) и поиска информации (Information Retrieval).**

Авторами статьи приводятся результаты разработки метода извлечения причинно-следственных связей на основе существующих когнитивных сервисов IBM Watson и StanfordCoreNLP, использующий сервисы Natural Language Understanding, StanfordParse, Natural Language Classifier и Stanford Classifier. В статье описывается программный комплекс созданный с целью исследования различных методов обнаружения каузальных связей в текстах на естественном языке. Входящий в состав данного комплекса программный инструментарий реализован в облачной инфраструктуре IBM Bluemix и предоставляет набор сервисов, позволяющих применять извлечение и классификацию связей в неструктурированном тексте, выполнять тестирование анализируемых методов, использовать средства администрирования для работы с сервисами и данными.

***Ключевые слова*—причинно-следственные связи, когнитивные сервисы, интеллектуальный анализ данных, методы извлечения информации.**

I. Введение

В современном мире значительная часть информационных ресурсов представлена в виде неструктурированного текста на естественном языке. Это материалы различных статей, документов, хранимые в открытых источниках веб-страницы, текстовые файлы. Для структурированных данных механизмы выборки достаточно хорошо специфицированы. А вот для неструктурированных текстов на естественном языке встает задача интеллектуального анализа текста.

Интеллектуальный анализ данных (Text Mining) представляет собой набор технологий и методов, цель которых заключается в извлечении ключевой и наиболее значимой информации из неструктурированного текста. В область знания Text Mining входят такие направления компьютерной лингвистики, как:

- Интеллектуальный анализ данных - Data mining;
- Анализ данных в сети Интернет – Web mining;
- Поиск информации - Information Retrieval;
- Извлечение информации - Information Extraction;
- Обработка текста на естественном языке - Natural Language Processing;

Технология интеллектуального анализа данных (ИАД) достаточно точно определяется в [1]: «Data Mining – это процесс обнаружения в больших сырых данных ранее неизвестных, объективных, полезных на практике знаний, необходимых для принятия каких-либо решений».

Основу математических методов Data Mining следует разделить на две группы: статистические методы (анализ временных рядов, корреляционный и регрессионный анализ, дискриминантный анализ и др.) и методы компьютерной математики (деревья решений, нечеткая логика, системы обработки экспертных знаний, искусственные нейронные сети и др.). Суть статистических методов состоит в том, что для них используют, в основном, большие массивы данных для формирования эффективных решений. А особенность методов компьютерной математики заключается в использовании взаимодействия кибернетических и математических алгоритмов обработки информации. Во многих случаях они позволяют построить доказательное решение задачи извлечения признаков из данных, не используя при этом большие объемы информации. Для решения таких задач могут использоваться алгоритмы основанные на применении деревьев решений, позволяющих реализовать построение задачи в виде графа, в котором его вершинам соответствуют некоторые правила вывода. Их функция заключается в

Статья получена 31 мая 2019.
Умаров Т.С., студент магистратуры ВМК (МГУ им. М.В.Ломоносова), (e-mail: umarov.tokhir@gmail.com).
Баженова И.Ю., МГУ им. М.В.Ломоносова, (e-mail: birina748@rambler.ru).

классификации данных или вычислении последствий решений по причинно-следственным связям между вершинами на различных уровнях иерархии. Наиболее часто они имеют вид двоичных переменных. Стоит отметить, что ограниченность в построении правил логического вывода является недостатком данного подхода, т.е. задачи обычно решаются формированием некой цепочки пересмотра признаков объекта, а не нахождением его закономерностей. В этой связи решения получаются не всегда оптимальными. Однако наглядность отображения хода решения стоит добавить к достоинствам данного подхода.

Технологии Web Mining предназначены для выявления взаимосвязей, фактов и событий в источниках, размещенных в сети Интернет. В соответствии с [2] можно выделить следующие виды технологий Web Mining:

- Анализ использования веб-ресурсов (Web Usage Mining). Извлечение данных из логов веб-серверов, целью которого является выявление предпочтений пользователей различных ресурсов;
- Извлечение веб-структур (Web Structure Mining). Анализ взаимосвязей между веб-страницами;
- Извлечение веб-контента (Web Content Mining). Анализ содержания документов, которые, в свою очередь, кластеризуются и классифицируются для группировки их по смысловой близости;

В целом, технологии Web Mining предназначены для анализа веб-ресурсов, поиска и извлечения информации из сети, а также выявления полезных тенденций в интернет пространстве.

Из-за увеличения количества неструктурированной информации, особенно в сети Интернет, роль такой процедуры как извлечение информации все больше возрастает и осложняется сам процесс извлечения. Современные алгоритмы извлечении информация основываются на статистических подходах с применением машинного обучения, так как необходимость в использовании большого количества помеченных, не зависящих от типа и области текстовых данных и в автоматическом извлечении неявных шаблонов из текста означает, что методы машинного обучения могли бы справляться лучше, чем изначально предложенные лингвистические методы. Таким образом, начиная с начала 2000-х годов парадигма решения проблемы автоматического извлечения информаций из неструктурированных данных стала тяготеть к статистике и машинному обучению (ML).

Технологии ML в задаче извлечения информации из текстов формулируется как решение задачи классификации с использованием статистических моделей. Методы машинного обучения, используемые для решения задачи, делятся на несколько этапов.

- Обучение «с учителем»: обучение на основе учебной коллекции, включающей явно специфицированные (вручную) именованные сущности. Методы этой группы оценивают параметры для положительно определенных примеров корпуса и при работе с новым корпусом используют значения этих параметров. Сюда относятся Байесовский классификатор, скрытые марковские модели, принцип максимума энтропии, деревья принятия решений, метод опорных векторов, условные случайные поля и др.
- Частичное обучение «с учителем»: от предыдущего подхода отличается тем, что исходная учебная коллекция содержит очень маленький набор начальных данных. При помощи реализации метода бутстреппинга осуществляется итеративное обучение классификатора.
- Обучение «без учителя»: для решения задачи не требуют предварительного создания корпуса примеров. Такие методы способны сделать вывод по сырому текстовому материалу.

Задачу Text Mining можно понимать в когнитивном контексте. Это обусловлено тем, что критерий качества выделения информации из информации определяется человеком. В целом когнитивные методы и процессы помогают преобразовать неявное знание в явное.

На этапе когнитивного исследования выполняется формирование связей в информационной структуре и определение направления связей. В этом отношении основной моделью представления знаний могут быть причинно-следственные связи.

Современные подходы извлечения причинно-следственных связей предполагают наличие средств автоматического извлечения данных связей в процессе поиска информации [3], поддержки принятия решений [4], или прогнозирование будущих событий [5]. Данная связь позволяет сконцентрировать важную информацию о том, как различные события и сущности должны восприниматься по отношению друг к другу. В частности, считается, что причинно-следственная связь играет очень важную роль в познании человека из-за его способности влиять на принятие решений.

Автоматическое извлечение причинно-следственных связей из текстов на естественном языке является сложной открытой проблемой в области искусственного интеллекта. Решив задачу извлечения причинно-следственных связей, можно реализовать потребность в создании такого инструмента, который автоматически искал бы множество текстовых материалов в глобальной сети и получал значимые причинно-следственные связи и образовывал из них причинные цепочки, чтобы обнаружить ранее неизвестные отношения между сущностями или событиями.

II. Механизмы определения причинно-следственных связей

Под причинно-следственной связью понимают связь между явлениями, при которой одно явление влечет за собой другое явление. Первое явление называется причиной, при наличии определенных условий порождает другое явление, называемое следствием. Одним из простейших способов выражения причинно-следственных связей между двумя событиями являются предложения типа «событие А вызвано событием Б» или «из события А следует событие Б». Причинность может

быть выражена с использованием множества различных типов предложений и иметь разнообразные синтаксические представления. Автор статьи [3] Кристофер Кху определил следующие способы выражения явных причинно-следственных связей на английском языке:

- с помощью причинных союзов для соединения двух фраз или предложений;
- с помощью причинных глаголов;
- с помощью результирующих конструкций, в которых после глагола следует фраза, описывающая состояние объекта в результате действия. Для обнаружения результирующих предложений использовались синтаксические паттерны вида Глагол-Существительное-Прилагательное, в которых результирующей фразой является прилагательное. Например, в предложении «я закрасил автомобиль желтым» прилагательное «желтым» является результирующей фразой, которое описывает результат действия «закрасил автомобиль»;
- с помощью условных выражений «если-то». Конструкции «если-то» часто указывают на то, что антецедент (часть «если») является причиной консеквента (часть «то»);
- с помощью причинных прилагательных или наречий.

К сожалению данная классификация не может быть обобщена для других языков, поэтому автоматическое извлечение причинно-следственных связей остается сложной технологической задачей.

Возможность автоматического извлечения причинно-следственных связей также исследуется в проекте Big Mechanism [6], спонсируемый агентством перспективных исследовательских проектов Министерства обороны США или DARPA. Цель данного проекта [7] заключается в создании такой системы, которая бы читала все существующие в мире статьи или научные журналы на различных языках, извлекала фрагменты из текстов, содержащие казуальные механизмы, конструировала из данных фрагментов общую причинно-следственную модель и базу знаний, на основе которых можно было подтверждать или предсказывать какие-либо гипотезы с приведением определенных фактов или объяснений. На

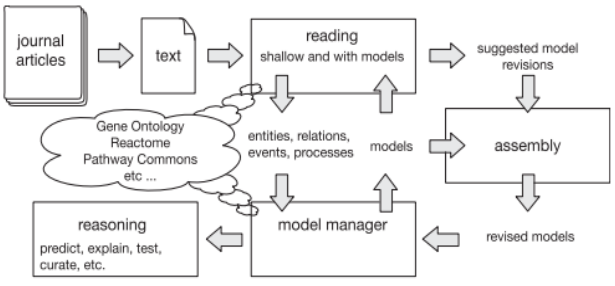

Рис.1. Общая архитектура системы Big Mechanism

Другим широко известным проектом является проект IBM Watson [8], использующий когнитивный подход для выполнения анализа большого количества различных внешних источников информации, выявления неочевидных зависимостей между разными видами хранимых данных. Этот проект позволяет, учитывая специфику данных, достаточно оперативно давать релевантный ответ на полученный запрос. При этом Watson справляется с задачей, во многих случаях, даже лучше человека и, более того, обработка данных идет гораздо быстрее, работа ведется с гораздо большими объемами.

Для того, чтобы научить систему анализировать сложные смысловые конструкции, с учетом эмоций и прочих факторов, специалисты использовали глубокую обработку естественного языка. А именно — вопросно-ответную систему контентной аналитики DeepQA (Deep Question-Answering) [9]. При анализе определенного вопроса, для того, чтобы дать правильный ответ, система старается оценить как можно более обширный контекст. Упрощённый алгоритм работы Watson при ответе на вопрос, заданный на естественном языке, выглядит следующим образом:

1. В каждом вопросе проводится синтаксический анализ для выделения основных особенностей вопроса.
2. Система генерирует ряд гипотез (варианты ответа), анализируя базу данных с фразами, в которых с определённой долей вероятности могут содержатся правильный ответ.
3. Система выполняет глубокое сравнение языка, на котором был задан вопрос, с языками, на которых содержится каждый один из возможных вариантов ответа, применяя различные алгоритмы определения логических связей. В системе существует сотни таких алгоритмов, и все они направлены на различную группу сравнений. Например, одни направлены на поиск совпадающих терминов и синонимов, другие рассматривают временные и пространственные особенности, третьи анализируют подходящие источники контекстуальной информации.
4. Далее, каждый логический алгоритм выставляет одну или несколько оценок, показывающие степень соответствия найденного ответа заданному вопросу.
5. Каждой оценке присваивается определённый весовой коэффициент. При этом используется статистическая модель, фиксирующая успешность работы каждого алгоритма при выявлении логических связей. Впоследствии данная модель используется для определения общего уровня уверенности Watson в том, что найденный ответ действительно является верным.
6. Пункты 3-5 повторяются до тех пор, пока Watson не найдет ответы, которые будут иметь наибольшие шансы оказаться правильными.

Создание системы, способной провести глубокую обработку естественного языка, позволило решить и другую проблему — анализ огромного количества информации, которая генерируется ежедневно. Это неструктурированная информация, вроде твитов, сообщений социальных сетей, отчетов, статей и т.п.. IBM Watson научился понимать неструктурированную информацию, находить связи между ней и использовать данные знания в различных целях.

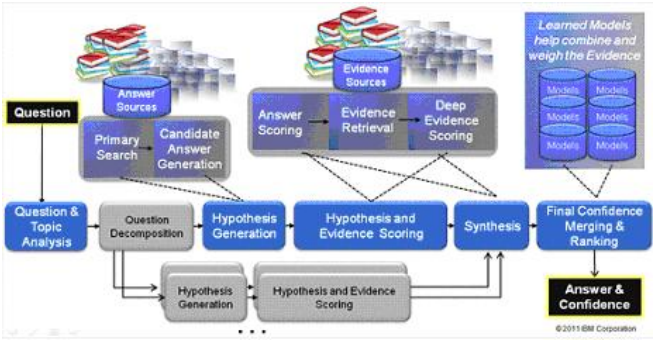

Рис.2. Архитектура Watson: DeepQA

Сейчас когнитивная система IBM Watson, благодаря многолетнему обучению и совершенствованию, может выполнять работу в самых разных сферах: медицине, лингвистике, информационной безопасности и др.

III. Методы извлечения информации

A. Методы извлечения информации на основе правил

Системы извлечение информации с помощью правил основаны на применении заданного набора правил. Такие системы используют экспертные знания для решения различных задач, которые обычно требуют человеческого интеллекта. Экспертное знание часто представляется в виде правил или как данные в компьютере, которые могут повторно применяться для решения задачи. В основном правила задаются в виде *ЕСЛИ X => ТО Y,* где в качестве паттерна *X* могут выступать регулярные выражения, словари, части речи или другие правила. Будем говорить, что фрагмент текста аннотируется соответствующей функцией Y, если он удовлетворяет одному из правил X.

Существует два главных принципа алгоритма изучения правил: восходящий метод (bottom-up) и нисходящий метод (top-down). В случае восходящего метода правила распространяются от исключений до общих случаев, а при нисходящем методе наоборот, от общих случаев до исключений. Данный подход используется в алгоритмах Whisk [10], LP2 [11] и Rapier [12]. Важным преимуществом данных алгоритмов является то, что степень достоверности извлеченной информации всегда является очень высоким. Однако множество правил определяются для конкретной предметной области, что является существенным недостатком.

B. Методы извлечения на основе классификации

Извлечение информации из текстовых документов с использованием статистических моделей основано на разделении исходного текста на вектор слов (токенов) и аннотировании каждого из этих слов классом из заданного вектора классов. При решении задачи классификации слова (объекта) в корпусе текстов определяется набор признаков, на основании которых объекты будут сопоставляться. Признак может принимать как булевы значения, так и числовые значения.

Таким образом, задача извлечения информации из текста сводится к классификации объектов, для решения которого существуют следующие подходы:

- Рационалистический подход. Идентификация объектов происходит на основе продукционных правил, которые задаются вручную.
- Машинное обучение. Задача поиска правил формулируется как решение задачи классификации с использованием статистических моделей
- Гибридный подход. Является объединением двух предыдущих подходов.

В случае машинного обучения модель классификации разделяет на два этапа: обучение и прогнозирование. При процессе обучения находится модель из аннотированных данных, которая может разделить обучающую выборку, а при процессе прогнозирования модель, которую нашли при обучении, используется для определения того, должен ли непомеченный экземпляр быть классифицированным.

Наиболее известными методами модели классификации являются следующие: метод опорных векторов (Support Vector Machines, SVM [13]), метод k-ближайших соседей (k-Nearest Neighbors, kNN [14]) и метод Naive Bayes [15].

IV. Методы извлечения причинно-следственных связей

К методам извлечения причинно-следственных связей можно отнести следующие методы:

- *Метод Гирджу и Молдована*

Полуавтоматический метод обнаружения лексико-синтаксических примеров объясняющих причинно-следственные связи. Данный алгоритм состоит из двух основных процедур [16]:

- обнаружение лексико-синтаксических примеров, выражающих причинно-следственные связи. При этом используется явный шаблон, определяющий причинно-следственную связь типа *NP1-Simple Causative Verb-NP2*. Алгоритм извлекает все существующие NP, в которых присутствуют причинно-следственные связи типа WordNet (электронный тезаурус для английского языка).

- утверждение и классификация противоречивых примеров. Автоматическое извлечение существительных и глаголов соответствующих NP1 и NP2, таким образом чтобы лексико-синтаксические шаблоны выражали причинно-следственную связь. В алгоритме в качестве NP1 и NP2 рассматривается основа именной фразы.

Вышеуказанные методы фокусировались в основном на решении проблемы противоречивости путем тщательного отбора характеристик обучающего алгоритма. При этом не учитывались проблема выявления более сложных – имплицитных причинно-следственных связей, которые включают в себя заключения на основе имеющихся знаний и семантического анализа.

- *Метод Ринка, Бежана и Гарабагиу*

В данном методе был применен корпус, использованный до этого в работе Бетхарда и Мартина [18], внешние характеристики, лексические файлы и

гипонимы из WordNet, оценка событий, основанная на подсчетах сети [17].

Этот подход состоит из 3-х фаз:

1. Создание графического представления текста. Здесь отражаются вся лексическая, семантическая и синтаксическая информация предложений текста.
2. Граф текста разбивается на множество более конкретных базовых графических шаблонов или подграфов, отражающих причинно-следственные отношения между событиями. Эти подграфы сортируются в соответствии с их релевантностью для определения причинно-следственных отношений
3. На последнем этапе, для того, чтобы решить действительно ли данные события являются казуальными, используется бинарный классификатор, который основан на том, какие извлеченные подграфы могут быть сопоставлены с графическим представлением предложения. Классификатор использует метод опорных векторов и обучен на основе корпуса [18], состоящий из статей журнала Wall Street.

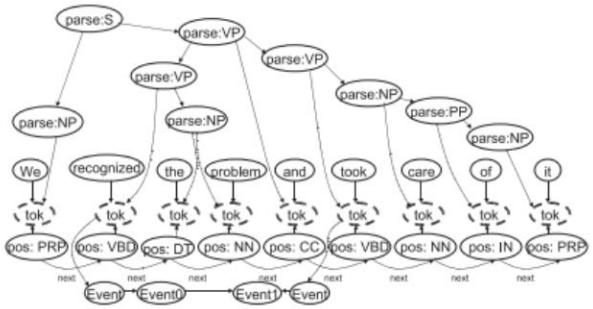

Рис. 3. Графическое представление текста метода Ринка

Здесь, tok = токены в предложении;
POS = часть речи (part-of-speech);
стрелки ↓ указывают на лексическую зависимость между POS.

Этот метод позволяет в определенной мере учитывать контекст при распознавании причинно-следственных отношений. Кроме того процесс более автоматизирован, т.к. сочетание признаков и структура текста определяется без применения ручных настроек. Однако при применении данного метода, невозможно избежать ошибок. Так, ввиду неспособности определять точное значение противоречивых глаголов (их значения зависят от контекста) система может ошибочно показать наличие причинно-следственной связи там, где ее нет и наоборот.

- *Метод Соргенте, Веттигли и Меле.*

Для нахождения пар "причина-следствие" в некотором предложении этот метод сначала анализирует заданное предложение на наличие причинно-следственных шаблонов с использованием байесовского классификатора и конфигурации данных, предложенной в рамках задачи 8 SemEval-2010 [19]. Если таковые имеются, производится грамматический разбор предложения и применяется определенные правила.

По сравнению с предыдущими методами, данный подход более сложный, но в то же время он продуктивнее.

V. Инструментальные средства поддержки интеллектуального анализа данных

В настоящий момент наиболее популярные инструменты аналитического ПО предоставляют широкий спектр механизмов классификации данных, методов статистического анализа, средств кластеризации и сегментации, инструментов визуализации, а также пакеты для анализа текстов (Text Mining) и поиска информации (Information Retrieval). К числу универсальных современных инструментов интеллектуальной обработки данных можно отнести следующий инструментарий:

• Apache OpenNLP [20]. Интегрированный пакет инструментов обработки текста, работающих на основе машинного обучения. Пакет работает на платформе Java и содержит решения большинства основных задач обработки естественного языка, в частности средства токенизации текста, разбиения на предложения, морфологической разметки, извлечения именованных сущностей, синтаксического разбора предложения, и др. Как правило, эти задачи активно применяются при построении сложных систем обработки текста. В состав OpenNLP включены инструменты машинного обучения на основе как методов максимальной энтропии, так и на основе персептрона. Имеется возможность интеграции с пакетом Apache UIMA [21].

• SAP HANA [22]. Предоставляет единую платформу для извлечения и анализа большого объема структурированных и неструктурированных данных в реальном времени из различных источников: социальные сети, блоги, онлайн-обзоры, сообщения электронной почты и обсуждения на форумах. Текстовая аналитика в SAP HANA - это набор таких лингвистических и статистических инструментов, как морфологическая разметка, извлечение именованных сущностей, извлечение семантических отношений, анализ тональности, оценка точности и полноты и другие.

• IBM Watson [23]. Инструмент предлагает широкий спектр возможностей для Data Mining и Text Mining: распознавание естественного языка, динамическое обучение системы, построение и оценка гипотез.

• Polyanalyst [24]. Многофункциональный набор, поддерживающий широкий спектр алгоритмов Data Mining. Последние версии включают в свой состав анализ текстов, лес решений, анализ связей. Присутствует поддержка технологий OLE DB for Data Mining и DCOM.

• Свободно распространяемый пакет программ Stanford CoreNLP [25]. Представляет собой набор алгоритмов машинного обучения для решения задач интеллектуального анализа данных. Stanford CoreNLP реализован на Java и запускается практически со всех платформ. В основном разрабатывается для работы с английским, но так же поддерживает арабский, китайский, французский и немецкий. Вместе с библиотекой отдельным пакетом доступен набор моделей языков.

Одной из основных тенденций является повышение степени интеграции предлагаемых библиотек инструментов. Это проявляется в стремлении включения поставщиками в состав решений растущего числа методов и технологий.

VI. Разработка метода обнаружение причинно-следственных связей

На основе анализа методов извлечения причинно-следственных связей и анализа когнитивных сервисов IBM Watson и библиотеки StanfordCoreNLP разработанный подход разделяет процесс на две основные процедуры подобно методу Сорgенте, Веттигли и Меле [19]:

1. Извлечение связей на основе правил. Для извлечения ключевых слов из текста используется сервис Natural Language Understanding и StanfordParser.

2. Классификация извлеченных связей на причинно-следственные и не причинно-следственные. Для выполнения данной процедуры используется сервис Natural Language Classifier и Stanford Classifier.

Процесс извлечения данных предлагается разделить на этапы, проиллюстрированные на рисунке 4.

Рис. 4. Алгоритм обнаружения причинно-следственных связей

Первым этапом решения задачи автоматической извлечения данных из текстов является преобразование поступающего текста. Цель предварительной обработки данных заключается в преобразовании в структурированный формат неструктурированного документа для конкретной предметной области.

Существует множество лингвистических и математических методов, позволяющих извлекать сущности, но, как правило, разделяют данные методы на два подхода. Первые подходы основывались на составленных вручную правилах, что требовало обширные познания в грамматике языка и делало такую систему ориентированной на ограниченное количество языков, или составлении списков рассматриваемых слов в справочниках, основным недостатком которого была необходимость в их постоянной поддержке и обновлении.

Позже применялись подходы, использующие статистические модели с применением методов машинного обучения, в частности, машинного обучения с учителем. В таких подходах использовали скрытые марковские модели (Hidden Markov Models, HMM), методы максимальной энтропии (Maximum Entropy, MaxEnt), деревья решений (Decision Tree, DT), наивный байес (Naive Bayes), опорных векторов (Support Vector Machine, SVM)

Общий принцип, примененный для решения задачи извлечения сущностей, можно разбить на следующие шаги:

- Определение признаков. Данные признаки будут применены ко всем токенам (словам) исследуемого текста и будут использованы сегментатором для определения кандидатов в сущность;
- Обучение сегментатора и классификатора сущностей на основе размеченных данных;
- Подсчет значений признаков для всех токенов в исследуемом тексте;
- Выделение набора кандидатов с помощью обученного сегментатора;
- Подсчет значений признаков для кандидатов в сущность, полученные из предыдущего пункта;
- Классификация кандидатов в сущность.

В каноническом виде на вход задаче извлечения сущностей подается предложение из предыдущего шага, а результатом данного этапа будем считать выделение множества сущностей из текста. В случае NLU ключевые слова извлекаются автоматически сервисом, а при использовании StanfordParser применяется правило.

Под извлечением связи из текста на естественном языке подразумевается, что пара извлеченных сущностей находятся в непосредственной близости или же являются частью одного и того же предложения. Сформулируем задачу данного этапа следующим образом: на входе имеется предложение и множество сущностей, извлеченные из рассматриваемого предложения. Нужно определить, существует ли связь между парой сущности и сгенерировать триплет, если связь извлечена.

Классификация используется для отнесения каждого документа к определенному классу с заранее известными признаками, полученными на этапе обучения. В современных системах классификация применяется, например, в таких задачах: группировка документов по общим признакам, размещение документов в определенные папки, избирательное распространение новостей подписчикам.

В соответствии с [26] существуют следующие виды связей:

- Причина-Следствие (Cause-Effect).
- Инструмент-Агент (Instument-Agency).
- Продукт-Производитель (Product-Producer).
- Сущность-Источник (Entity-Origin).
- Сообщение-Тема (Message-Topic).
- Часть-Целое (Part-Whole).
- Контент-Контейнер (Content-Container).

Разработанный метод ориентирован на реализацию связи Причина-Следствие и, следовательно, он позволяет строить бинарный классификатор с заданными классами: causal-relation и not-causal-relation.

В нашем случае задачу классификации связей следует разделить на два основных этапа: построение модели и классификация триплета на основе результата предыдущего шага. На первом этапе строим модель или классифицирующую функцию при помощи обучения на примерах, которая могла бы разделить обучающую выборку. Для построения данной классифицирующей функции используются NLC и Stanford Classifier. Далее, происходит классификация триплета, полученного на входе данного компонента, к двум предопределенным классам: causal-relation и not-causal-relation.

Для данной задачи был выбран подход обучения с учителем. В результате классификации получаем вероятностные оценки соответствия триплета к каждому классу. Следовательно, триплет будем соотносить тому классу, у которого больше вероятностная оценка.

Разработанный алгоритм обнаружения причинно-следственных связей состоит из следующих этапов:

1. Обнаружение предложений. Первоначально, исходный документ разбивается по предложениям, и для каждого предложения выполняются шаги 2-5.
2. Извлечение ключевых слов. Извлечение ключевых слов из предложения производится сторонним сервисом через API, на вход которого подается сам текст, а на выходе получаем множество ключевых слов со степенью достоверности.
3. Фильтрация ключевых слов. Из полученного массива удаляются все ключевые слова, степень достоверности у которых меньше оценки 0.32, т.к. по результатам тестирования было выявлено, что все рассматриваемые ключевые слова имели степень достоверности выше данной оценки.
4. Извлечение отношений между парой ключевых слов и генерация триплета. Сначала, для всех ключевых слов, полученного после этапа фильтрации, определятся позиция слова в предложении. После чего, для каждой ближайшей паре ключевых слов извлекается отношение, определяющее все слова, расположенные между указанной парой ключевых слов После извлечения отношения для каждой пары ключевых слов генерируется триплет.
5. Классификация триплета. Для классификации связи используется внешний программный модуль, который ранжирует триплет по двум обученным классам: causal-relation и not-causal-relation. На выходе данного модуля получаем список классов с их степенью достоверности, и триплет соотносится тому классу, у которого степень достоверности больше всего.
6. Вывод результатов. Записать результаты работы алгоритма в файловый документ, в котором перечислены все извлеченные отношения и соответствующий им класс.

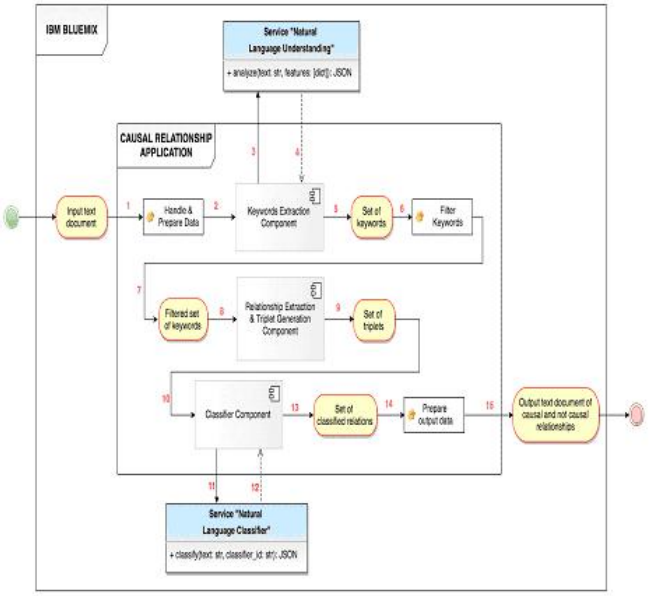

Рис. 5. Алгоритм обнаружения причинно-следственных связей

Предложенный алгоритм был реализован в разработанном программном комплексе (ПК).

В рамках реализации данного ПК использовались следующие инструменты:

- Язык программирования. Программа разработана на языке Python 3.7.3 с использованием системы управления пакетами pip 19.0.3. (используеме пакеты: Django==2.2, dateutils, psycopg2-binary, pytz, ibm-watson, nltk, numpy, requests, sqlparse).
- Система контроля версий Git. Git – распределенная система контроля версий, предоставляет каждому разработчику локальную копию всей истории разработки.
- Среда разработки. Разработка, в основном, велась на IDE PyCharm 2016 – интегрированная среда разработки для языка программирования Python, которая предоставляет средства для графического отладчика, анализа кода и контроля версий системы. Для изменения конфигурационных файлов на сервере использовался редактор Vim.
- Фреймворк. Программа реализована на фреймворке Django, который использует паттерн проектирования MVC (model-view-controller).
- Облачная среда. Программа развернута и администрируется на облаке IBM Bluemix.
- Сервисы IBM Watson. На облаке Bluemix запущены когнитивные сервисы NLU и NLC, обращение к которым происходит через технологии REST API.
- ПК клиента. Для фронтенда использовался язык Nodejs/npm с технологиями html/css/js/bootstrap
- База данных. Для хранения данных бала выбрана PostgreSQL 9.6.

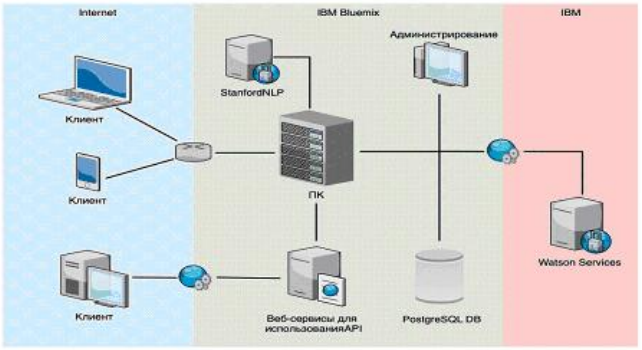

Рис.6. ПК для обработки причинно-следственных связей в текстах на естественном языке

VII. Заключение

Предложенный в работе алгоритм обнаружения причинно-следственных связей в неструктурированных текстах на естественной языке позволяет автоматизировать поиск информации с применением открытого API IBM Watson и Stanford CoreNLP.

В дальнейшем в разработанный алгоритм можно включить применение машинного обучения и правил на Watson Knowledge Studio, что позволит значительно улучшить качество извлечения ключевых слов или сущностей с помощью аннотирования различных данных. Также имеется возможность дальнейшего улучшения качества извлекаемых из текстов связей посредством формирования новых паттернов с использованием StanfordParser.

С учетом все возрастающего объема неструктурированных текстовых данных проблема обнаружения причинно-следственных связей и выборки отвечающей условиям информации остается важной и актуальной задачей.

Библиография

[1] В. А. Дюк, А. В. Флегонтов, И. К. Фомина. Применение технологий интеллектуального анализа данных в естественнонаучных, технических и гуманитарных областях.
[2] Технологии Text mining и Web mining [Электронный ресурс]. URL: https://nauchforum.ru/archive/MNF_tech/4(33).pdf
[3] C. S. Khoo, J. Kornfilt, R. N. Oddy, and S. H. Myaeng. Automatic extraction of cause-effect information from newspaper text without knowledge-based inferencing.
[4] E. J. M. Ackerman. Extracting a causal network of news topics.
[5] K. Radinsky, S. Davidovich, and S. Markovitch. Learning causality for news events prediction.
[6] DARPA Big Mechanism [Электронный ресурс]. URL: http://www.darpa.mil/program/big-mechanism
[7] A. Rzhetsky. The Big Mechanism Program: Changing How Science Is Done.
[8] J. Best. IBM Watson: The Inside Story Of How The Jeopardy-Winning Supercomputer Was Born, And What It Wants To Do Next
[9] The DeepQA Project [Электронный ресурс]. URL: https://www.research.ibm.com/deepqa/deepqa.shtml
[10] S. Soderland. Learning information extraction rules for semi-structured and free text.
[11] F. Ciravegna. (LP)2, an Adaptive Algorithm for Information Extraction from Web-related Texts.
[12] M. E. Cali. Relational Learning Techniques for Natural Language Information Extraction.
[13] К. В. Воронцов. Лекции по методу опорных векторов
[14] k-nearest neighbors algorithm [Электронный ресурс]. URL: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
[15] Naïve Bayes Classifier [Электронный ресурс]. URL: https://en.wikipedia.org/wiki/Naive_Bayes_classifier
[16] R. Girju, D. Moldovan. Text Mining for Causal Relations.
[17] B. Rink, C. Bejan, S. Harabagiu. Learning Textual Graph Patterns to Detect Causal Event Relations.
[18] S. Bethard, W. Corvey, S. Klingenstein, J. H. Martin (2008). Building a Corpus of Temporal-Causal Structure.
[19] A. Sorgente, G. Vettigli, F. Mele. Automatic extraction of cause-effect relations in Natural Language Text.
[20] Apache OpenNLP [Электронный ресурс]. URL: https://opennlp.apache.org/
[21] Apache UIMA [Электронный ресурс]. URL: https://uima.apache.org/
[22] SAP Hana [Электронный ресурс]. URL: https://sap.com/products/hana.html
[23] IBM Watson Services on IBM Bluemix [Электронный ресурс]. URL: https://console.ng.bluemix.net/catalog/
[24] Polyanalyst [Электронный ресурс]. URL: http://www.megaputer.ru
[25] Stanford CoreNLP «Core Natural Language software» [Электронный ресурс]. URL: https://stanfordnlp.github.io/CoreNLP/
[26] R. Girju, M. Hearst, P. Nakov, V. Nastase, S. Szpakowicz, P. Turney, D. Yuret. Classification of Semantic Relations between Nominals. // SemEval 2007 task 8.

Modern approaches to the mechanisms of causal relationships extraction from unstructured natural language texts

T.S. Umarov, I.U. Bazhenova

***Abstract*— The article is devoted to the study of methods of extracting causal relationships from unstructured text in a natural language. Automatic extraction of causal relationships from natural language texts is a complex open problem in artificial intelligence. The ways of expressing explicit causal relationships used in information extraction technologies are described. The article provides a brief overview of modern data mining support tools that allow classifying data, using statistical analysis methods, clustering and segmentation tools, using visualization tools, as well as text analysis and information retrieval packages.**

The authors of the article present the results of developing a method for extracting cause-and-effect relationships based on the existing IBM Watson and StanfordCoreNLP cognitive services using the Natural Language Understanding, StanfordParse, Natural Language Classifier and Stanford Classifier services. The article describes a software package designed to investigate various methods for detecting causal connections in natural language texts. The software toolkit included in this complex is implemented in IBM Bluemix cloud infrastructure and provides a set of services that allow you to extract and classify relationships in unstructured text, test the analyzed methods, use administrative tools for working with services and data.

***Keywords*— cause-effect relationships, cognitive services, data mining, information extraction methods.**

REFERENCES

[1] V. A. Dyuk, A. V. Flegontov, I. K. Fomina. Primeneniye tekhnologiy intellektual'nogo analiza dannykh v yestestvennonauchnykh, tekhnicheskikh i gumanitarnykh oblastyakh.
[2] Tekhnologii Text mining i Web mining [Elektronnyy resurs]. URL: https://nauchforum.ru/archive/MNF_tech/4(33).pdf
[3] C. S. Khoo, J. Kornfilt, R. N. Oddy, and S. H. Myaeng. Automatic extraction of cause-effect information from newspaper text without knowledge-based inferencing.
[4] E. J. M. Ackerman. Extracting a causal network of news topics.
[5] K. Radinsky, S. Davidovich, and S. Markovitch. Learning causality for news events prediction.
[6] DARPA Big Mechanism [Электронный ресурс]. URL: http://www.darpa.mil/program/big-mechanism
[7] A. Rzhetsky. The Big Mechanism Program: Changing How Science Is Done.
[8] J. Best. IBM Watson: The Inside Story Of How The Jeopardy-Winning Supercomputer Was Born, And What It Wants To Do Next
[9] The DeepQA Project [Электронный ресурс]. URL: https://www.research.ibm.com/deepqa/deepqa.shtml
[10] S. Soderland. Learning information extraction rules for semi-structured and free text.
[11] F. Ciravegna. (LP)2, an Adaptive Algorithm for Information Extraction from Web-related Texts.
[12] M. E. Cali. Relational Learning Techniques for Natural Language Information Extraction.
[13] K. V. Vorontsov. Lektsii po metodu opornykh vektorov
[14] k-nearest neighbors algorithm [Электронный ресурс]. URL: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
[15] Naïve Bayes Classifier [Электронный ресурс]. URL: https://en.wikipedia.org/wiki/Naive_Bayes_classifier
[16] R. Girju, D. Moldovan. Text Mining for Causal Relations.
[17] B. Rink, C. Bejan, S. Harabagiu. Learning Textual Graph Patterns to Detect Causal Event Relations.
[18] S. Bethard, W. Corvey, S. Klingenstein, J. H. Martin (2008). Building a Corpus of Temporal-Causal Structure.
[19] A. Sorgente, G. Vettigli, F. Mele. Automatic extraction of cause-effect relations in Natural Language Text.
[20] Apache OpenNLP [Электронный ресурс]. URL: https://opennlp.apache.org/
[21] Apache UIMA [Электронный ресурс]. URL: https://uima.apache.org/
[22] SAP Hana [Электронный ресурс]. URL: https://sap.com/products/hana.html
[23] IBM Watson Services on IBM Bluemix [Электронный ресурс]. URL: https://console.ng.bluemix.net/catalog/
[24] Polyanalyst [Электронный ресурс]. URL: http://www.megaputer.ru
[25] Stanford CoreNLP «Core Natural Language software» [Электронный ресурс]. URL: https://stanfordnlp.github.io/CoreNLP/
[26] R. Girju, M. Hearst, P. Nakov, V. Nastase, S. Szpakowicz, P. Turney, D. Yuret. Classification of Semantic Relations between Nominals. // SemEval 2007 task 8.