П

УДК 004.912

Я. С. Пикалёв
Государственное учреждение «Институт проблем искусственного интеллекта», г. Донецк
83048, г. Донецк, ул. Артема, 118-б

ОБЗОР АРХИТЕКТУР СИСТЕМ ИНТЕЛЛЕКТУАЛЬНОЙ ОБРАБОТКИ ЕСТЕСТВЕННО-ЯЗЫКОВЫХ ТЕКСТОВ

Ya. S. Pikalyov
Public institution «Institute of Problems of Artificial intelligence», c. Donetsk
83048, Donetsk, Artema str., 118-b

OVERVIEW OF ARCHITECTURES OF SYSTEMS FOR INTELLIGENT PROCESSING OF NATURAL LANGUAGE TEXTS

Я. С. Пикальов
Державна установа «Інститут проблем штучного інтелекту», м. Донецьк
83048, м. Донецьк, вул. Артьома, 118-б

ОГЛЯД АРХІТЕКТУР СИСТЕМ ІНТЕЛЕКТУАЛЬНОЇ ОБРОБКИ ПРИРОДНО-МОВНИХ ТЕКСТІВ

В статье рассмотрены современные платформы, в которых реализованы функции автоматической обработки естественно-языковых текстов: морфологический и синтаксический анализ, классификация и тематическая рубрикация текста, определение языка, тональной и эмоциональной окраски, выделение сущностей, поддержка диалога с пользователем и т.п. Рассмотренные системы используют методы глубокого обучения и словари, предоставляют API для реализации функций интеллектуальной обработки текстов для пользовательских задач.
Ключевые слова: автоматическая обработка текста, токенизация, стемминг, выделение именованных сущностей, архитектура NLP-систем.

The article discusses modern platforms in which the functions of automatic processing of natural language texts are implemented: morphological and syntactic analysis, classification and thematic heading of text, definition of language, tonal and emotional coloring, highlighting of entities, support of dialogue with the user, etc. The considered systems use deep learning methods and dictionaries, provide an API for implementing intelligent word processing functions for user tasks.
Key words: Natural Language Processing, tokenization, stemming, named entity recognition, architecture of NLP systems.

У статті розглянуті сучасні платформи, в яких реалізовані функції автоматичної обробки природно-мовних текстів: морфологічний і синтаксичний аналіз, класифікація та тематична рубрикація тексту, визначення мови, тональної та емоційного забарвлення, виділення сутностей, підтримка діалогу з користувачем і т.п. Розглянуті системи використовують методи глибокого навчання і словники, надають API для реалізації функцій інтелектуальної обробки текстів для задач користувача.
Ключові слова: автоматична обробка тексту, токенізація, стемінг, виділення іменованих сутностей, архітектура NLP-систем.

Введение

NLP (Natural Language Processing), или обработка естественного языка, – это область искусственного интеллекта (ИИ), задачей которой является разработка методов и систем, обеспечивающих общение с компьютерами на естественном языке (ЕЯ). Проблематика ЕЯ-общения находится на стыке таких наук, как лингвистика, психология, логика и философия, каждая из которых исследует лишь отдельные аспекты процесса коммуникативного воздействия. ИИ, как прикладная дисциплина, моделирует в рамках NLP-систем основные аспекты ЕЯ-общения.

Сложность создания средств человеко-машинного общения, которые предназначены для конечного неподготовленного пользователя, обусловлена отсутствием единой теории языкового общения, которая способна охватить все аспекты взаимодействия коммуникантов. Случайная, зависящая от контекста и неформализуемая правилами природа естественных языков приводит к тому, что на процесс взаимодействия налагаются ограничения, вследствие которых не выполняются требования конечных пользователей.

Перспективы развития сегмента NLP обусловлены применением NLP-систем в повседневных бизнес-процессах, заточенных под конкретные задачи:

– оптическое распознавание символов;
– распознавание и синтез речи;
– машинный перевод;
– анализ настроений, анализа социальных сетей;
– анализ тональности текста;
– семантический поиск;
– аннотирование и классификация документов;
– вывод информации на естественном языке (используется в диалоговых системах – чат-ботах).

В настоящий момент растет интерес ведущих ИТ-компаний к разработке и внедрению технологических решений, разработанных на основе NLP, позволяющих осуществлять интеллектуальную обработку ЕЯ-текстов. Обзор архитектур и возможностей современных NLP-систем представлен ниже.

Цель данной работы – анализ архитектур современных систем, осуществляющих интеллектуальную автоматическую обработку ЕЯ-текстов, а также возможностей их использования разработчиками NLP-систем.

Аналитический программный комплекс 3i NLP Platform

3i NLP Platform – программный продукт консорциума 3i Technologies, предназначенный для статистического анализа массивов текстовой информации на ЕЯ. Продукт осуществляет морфологический анализ текстовых массивов на русском и английском языках, выявляет сущности, рассчитывает тональности для документов и сущностей, визуализирует результат обработки и решает прочие задачи с использованием технологий ИИ. NLP-обработка в системе 3i NLP Platform [1] осуществляется последовательно несколькими основными модулями. Процесс обработки представлен на рис. 1.

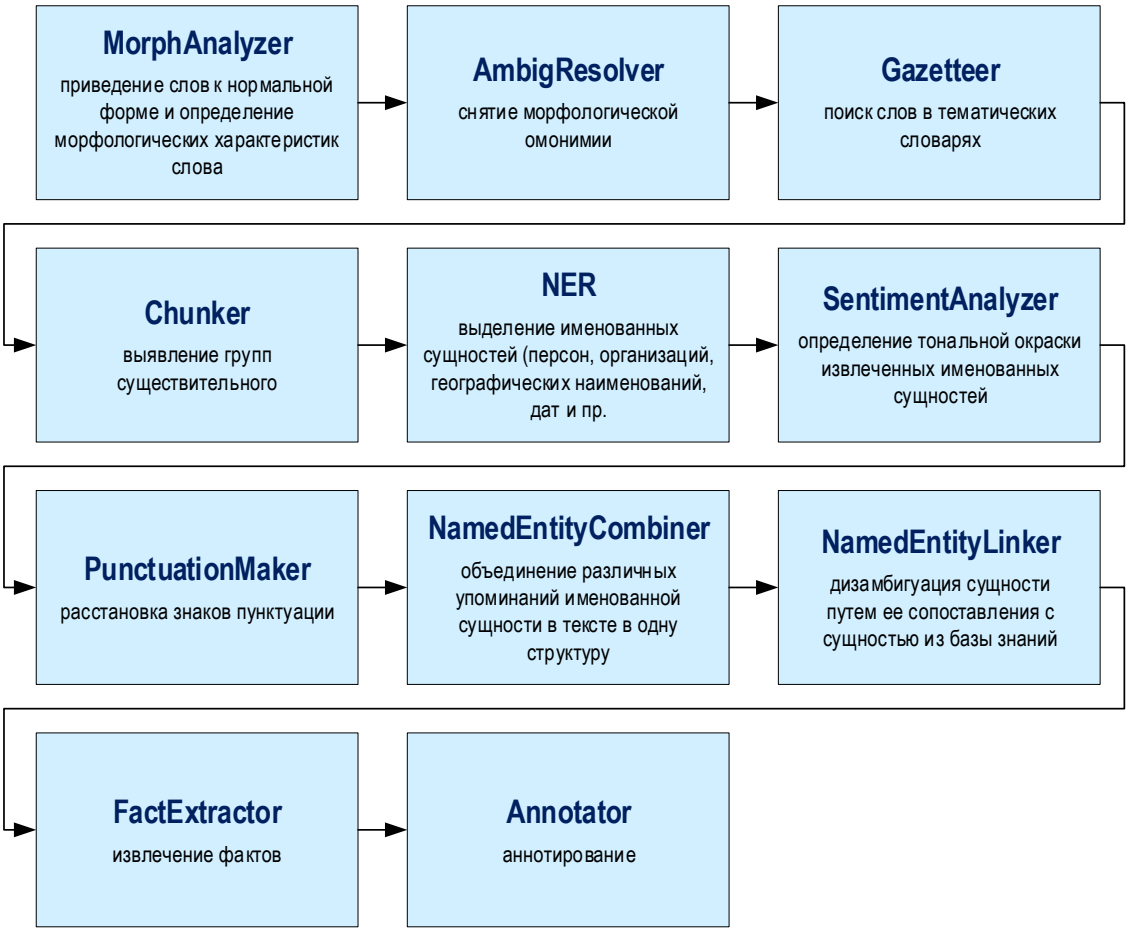

Рисунок 1 – Последовательность NLP-обработки в 3i NLP Platform

Опишем кратко подходы, на основе которых реализован каждый модуль.

Модуль морфологии MorphAnalyzer используется для приведения слов к нормальной форме и выявления морфологических характеристик слова, таких как падеж, род число, часть речи и т.д. В основе модуля для русского языка лежит словарь Зализняка (подобные словари используются и для иностранных языков), широко используемый для данной задачи в русскоязычных системах обработки текста. Для русского и английского языка в качестве основы использовались словари проекта АОТ. Все словари обработаны, объединены, исправлен ряд ошибок и неточностей. Для базовой поддержки могут использоваться словари формата Hunspell, iSpell и т.д.

Для обработки слов, не содержащихся в словаре, используется алгоритм предсказания незнакомых слов. Алгоритм основан на поиске знакомого слова с максимально совпадающей флексией (окончанием), после чего искомое слово получает тот же тип словоизменения и морфологических характеристик. Также используется информация о регистре первой буквы слова, что позволяет сделать предсказание более точным для слов с совпадающими словоформами, такими как «газа» (родительный падеж от «газ») и «Газа» (сектор Газа), «кроликов» (родительный падеж от «кролик») и «Кроликов» (фамилия).

Для типичных окончаний фамилий организована особая обработка, в автомат добавлены соответствующие варианты. Все словари представлены в памяти в виде детерминированных конечных автоматов. В конечных состояниях автомата хранится

морфологическая информация о слове и его тип склонения. Автоматы реализованы с помощью алгоритмов собственной разработки, без привлечения сторонних библиотек. Модуль поддерживает многопоточность.

Модуль AmbigResolver предназначен для снятия морфологической омонимии. В результате работы модуля происходит полная дизамбигуация, т.е. из вариантов разбора слова, полученных от модуля морфологии, выделяется один наиболее вероятный полный разбор и соответствующая ему нормальная форма слова. Модуль основан на машинном обучении. В качестве алгоритма выбран алгоритм условных случайных полей (CRF – Conditional Random Fields), конкретная реализация алгоритма – свободная библиотека CRFsuite.

Машинное обучение использует в качестве параметров обучения варианты морфологического разбора самого слова, а также контекстных слов, отстоящих от него на регулируемое расстояние. В качестве корпуса используется подкорпус со снятой омонимией OpenCorpora, а также корпус, составленный из новостных статей, обработанных собственной системой и подготовленный группой штатных лингвистов компании. Модуль поддерживает многопоточность.

Модуль Gazetteer предназначен для поиска слов в тематических словарях. В результате найденные слова приводятся в нормальную форму (если с этим не справился модуль Морфологии) и им приписывается новая информация (семантический класс). Примерами семантического класса являются геолокация, организация, имя, отрицание и т.д. Семантический класс может оказаться финальным, но также он может изменяться и использоваться в качестве внутренней служебной информации при работе последующих модулей.

В качестве базовых словарей используются материалы, собранные из различных источников (СМИ, различные корпусы, словари GATE), в том числе Википедии. Словари постоянно пополняются. Генерацию правил в полуавтоматическом режиме осуществляет специальный алгоритм. Для сложно склоняемых и часто упоминаемых сущностей составлены словари специального вида, где перечислены все формы, в которых может упоминаться сущность, включая аббревиатуры и варианты написания. Словари представлены в модуле автоматом, построенным по алгоритму Ахо-Корасик.

Модуль Chunker предназначен для выявления групп существительного. Эта информация используется впоследствии другими модулями для более точного определения границ сущностей и объектов сентимента. Модуль основан на применении машинного обучения. В качестве алгоритма выбран алгоритм условных случайных полей (CRF – Conditional random fields), реализация алгоритма выполнена на основе свободной библиотека CRFsuite.

Материалами для машинного обучения являются собственный размеченный корпус новостных текстов и автоматический корпус, полученный после обработки новостных текстов сторонним синтаксическим анализатором и применения правил по выделению групп существительного из размеченного им текста. Модуль поддерживает многопоточность.

Модуль NER (Named Entity Recognition) предназначен для выделения именованных сущностей, таких, как геолокации, организации, персоны и пр. Для работы модулю необходима информация от предыдущих модулей. Работа модуля основана на правилах, в которых заданы последовательности информационных меток, полученных от предыдущих модулей, которые при определенных условиях объединяются в одну именованную сущность. На основе результатов работы модуля Chunker, алго-

ритм дополняет автоматически выделенные названия организаций. Это необходимо для исправления сложных названий организаций, которые были выделены не полностью в силу ограниченности правил. Также происходит пост-обработка персон, в процессе которой модуль пытается согласовать нормальные формы имени, фамилии и отчества персоны. Это необходимо для персон, содержащих словоформы с двоякой интерпретацией, такие как имя «Александра», которое может употребляться в женском роде именительном падеже или мужском роде родительном падеже. При согласовании учитывается статистика упоминания словоформы в документе.

В конце работы модуля происходит финальная обработка сущностей, в процессе которой одинаковые сущности объединяются, служебная информация удаляется из итоговых результатов. Далее происходит поиск «потерянных» сущностей, например, отдельных фамилий или имен, которые относятся к уже выделенным полным сущностям. Так же на данном этапе происходит восстановление регистра сущностей. Внутреннее представление правил NER – автомат, построенный по алгоритму Ахо-Корасик с дополнительными модификациями, необходимыми для нечеткой обработки правил. Модуль поддерживает многопоточность.

Модуль SentimentAnalyzer используется для определения тональной окраски именованных сущностей (object-based sentiment analysis) и отношения к их различным аспектам (aspect-based sentiment analysis). Работа алгоритма основана на гибридном подходе – использовании словарей сентиментов (тонально-окрашенных слов) и машинном обучении. В качестве алгоритма выбран алгоритм условных случайных полей (CRF – Conditional random fields), реализация алгоритма выполнена на основе свободной библиотека CRFsuite. Далее по установленным правилам определяется принадлежность сентиментов к ближайшим именованным сущностям. В правилах учитывается:

- расстояние от сущности до сентимента;
- наличие между сущностью и сентиментом других сущностей или местоимений;
- наличие у сущности однородных членов, к которым возможно тоже относится данный сентимент.

Помимо словарей сентиментов используются также словари модификаторов, к которым относятся отрицания, усиления и уменьшения силы тональности. Отрицания меняют полярность сентимента на противоположную, модификаторы увеличивают или уменьшают силу сентимента. Сентимент определяется для каждого упоминания именованной сущности, и в конце работы модуля он суммируется. Агрегация происходит с учетом количества упоминаний сущности и длины документа. В итоге каждая уникальная сущность в документе получает одну из четырех оценок – нейтральная, позитивная, негативная и смешанная. Нейтральная оценка дается сущностям, по отношению к которым не выражено никакого сентимента, либо этот сентимент незначителен в масштабах текста. Смешанная оценка ставится сущностям, по отношению к которым выражено достаточное количество как позитивных, так и негативных сентиментов. Итоговому документу тоже проставляется общая оценка тональности, рассчитываемая с учетом длины текста и всех тональностей именованных сущностей в нем.

Модуль PunctuationMaker предназначен для автоматической расстановки знаков пунктуации (точек и запятых) в текстах, полученных путем распознавания речи и оптического распознавания символов. Модуль основан на машинном обучении. В качестве алгоритма выбран алгоритм условных случайных полей (Conditional random fields, CRF), существующая реализация алгоритма основана на свободной

библиотеке CRFsuite. Машинное обучение использует в качестве параметров обучения нормальную форму и морфологические характеристики самого слова и его контекстных соседей. В качестве корпуса используется подкорпус со снятой омонимией OpenCorpora. В дальнейшем планируется использовать информацию о паузах в речи и смене диктора при обработке аудиоинформации. Модуль поддерживает многопоточность.

Модуль NamedEntityCombiner предназначен для объединения различных упоминаний именованной сущности в тексте в одну структуру. Для этой задачи на данном этапе используется текстовая информация, правила и словари именованных сущностей. Модуль сравнивает инициалы и полные имена, фамилии и полные ФИО, аббревиатуры и полные названия организаций. Для персон модуль распознает такие упоминания, как «Путин», «В. В. Путин», «Президент РФ», «Владимир Владимирович» и т.д. Для организаций модуль объединяет полные названия организаций, краткие названия, аббревиатуры.

Модуль NamedEntityLinker предназначен для сопоставления именованной сущности с сущностью из базы знаний и ее дизамбигуации. Таким образом, модуль способен отличать Майкла Джордана – баскетболиста от Майкла Джордана – ученого и других людей, с таким же именем. Если сущность содержится в базе знаний, то после ее определения, мы можем получить информацию о ее связях, роде деятельности (для персон и организаций) и тому подобную информацию. В процессе работы модуль использует информацию о контексте слов, сущностях, упоминаемых вместе с ними, категории, к которой был отнесен текст. Далее данная информация сопоставляется с различными кандидатами из базы знаний, в результате чего выделяется наиболее вероятный кандидат. Сопоставление происходит с применением машинного обучения и правил, составленных как вручную, так и автоматически. Если же сущности с таким именем нет в базе знаний, имеется возможность ее пополнить, указав для сущности автоматически выделенную информацию о связях, категории и т.д., либо добавить эту информацию вручную.

Модуль FactExtractor (извлечение фактов) выделяет из текста именованные сущности, которые связаны между собой какой-либо логической и заранее определенной связью в один факт. Используя словарь ядер и выделенные системой сущности разных типов, модуль извлечения фактов устанавливает взаимосвязи между несколькими сущностями и формирует определенную структуру из неструктурированных текстовых данных:

- сущности – действующие лица факта (персоны, организации и др.);
- действия, объединяющее эти сущности;
- местоположение, дату и другую вспомогательную информацию.

В модуле реализованы набор правил для каждого типа факта, составленный лингвистами и описанный на языке шаблонов, а также словарь так называемых, ядер факта – глагольных групп, вокруг которых именованные сущности объединяются в факт заданного типа. Каждому факту присваивается уникальный идентификатор, позволяющий строить аналитику, агрегировать факты, объединенные общими полями и отображать их в виде графов.

Модуль аннотирования Annotator извлекает из текста ранжированный по важности список предложений. Важность предложений определяется плотностью ключевых слов, именованных сущностей и фактов.

Инструментарий для обработки текста Apache OpenNLP

Библиотека Apache OpenNLP [2], [3] представляет собой набор инструментов на основе машинного обучения для обработки текста на естественном языке. Поддерживает наиболее распространенные задачи NLP, такие как: токенизация, сегментация предложений, POS-тегирование, извлечение именованных сущностей, синтаксический анализ и разрешение кореференции. Процесс обработки представлен на рис. 2.

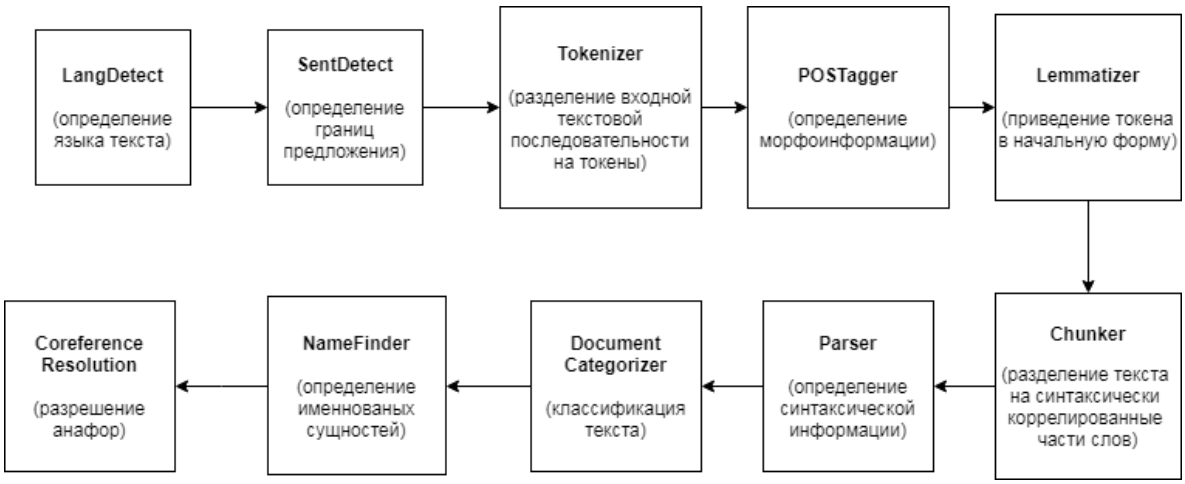

Рисунок 2 – Последовательность NLP-обработки в Apache OpenNLP

Модуль Langdetect предназначен для определения языка документа (по стандарту ISO-639-3). По умолчанию нормализуется текст, и извлекаются n-граммы (размерности 1, 2 и 3). Модель основана на следующих алгоритмах классификации: максимальной энтропии, многослойного персептрона, а также наивном байесовском классификаторе.

Детектор предложений Sentdetect помогает обнаружить границы предложений. Предложение определяется как самая длинная последовательность символов, обрезанная пробелами, между двумя знаками препинания. Первое и последнее предложение составляют исключение из этого правила. Предполагается, что первый непробельный символ является началом предложения, а последний непробельный символ – концом предложения.

Модуль Tokenizer сегментирует входную последовательность символов в токены (слова, знаки препинания, цифры и т. д.). OpenNLP предлагает несколько реализаций токенизатора:

- простой токенизатор – последовательности одного и того же класса символов являются токенами;
- токенизатор на основе пробелов – непробельные последовательности идентифицируются как токены;
- обучаемый токенизатор – определяет границы токена на основе вероятностной модели.

В OpenNLP (как и во многих системах) токенизация представляет собой двухэтапный процесс: сначала определяются границы предложений, затем идентифицируются токены в каждом предложении.

Name Finder может обнаруживать именованные объекты и числа в тексте. Модель зависит от языка и типа объекта, для которого она была обучена. OpenNLP предлагает ряд предварительно обученных моделей.

Document Categorizer отвечает за классификацию документов. OpenNLP может классифицировать текст по заранее определенным категориям. Классификация документов основана на принципе максимальной энтропии. Предварительно обученные модели для задачи классификации текстов отсутствуют, но классификатор документов можно обучать на аннотированных данных в формате OpenNLP Document (один документ в строке, содержащий категорию и текст, разделенные пробелом).

Модуль POSTagger определяет морфологическую информацию для токена (pos-тег), используя текущий токен и его контекст. Токен может иметь несколько pos-тегов в зависимости от токена и контекста. OpenNLP POSTagger использует вероятностную модель для прогнозирования правильного pos-тега из набора тегов. Чтобы ограничить возможные pos-теги, дополнительно используется словарь тегов, который увеличивает производительность определения морфологической информации.

Модуль Lemmatizer возвращает для токена и тега части речи начальную форму слова, которую обычно называют его леммой. Токен может быть неоднозначно получен из нескольких основных форм или словарных слов, поэтому для поиска леммы необходим pos-тег слова. В настоящее время в OpenNLP реализованы как статистические, так и словарные лемматизаторы.

Модуль Chunker реализует разделение текста на синтаксически коррелированные части слов, такие как группы существительных, группы глаголов, но не определяет их внутреннюю структуру и их роль в предложении.

Модуль Parser реализует функционал синтаксического анализатора, формируя синтаксическое дерево, содержащее синтаксическую информацию и информацию о связях между словами в предложении.

Система разрешения связей между словами Coreference Resolution в OpenNLP связывает несколько упоминаний сущности в документе вместе. Реализация этого модуля в OpenNLP на текущий момент ограничена упоминаниями имен существительных, другие типы упоминаний не могут быть разрешены.

Обработка естественного языка с помощью языковых служб Microsoft Azure Cognitive Service

Главным отличием Microsoft Azure Cognitive Service является использование контейнеров для интеллектуальной обработки данных, что обеспечивает гибкость в выборе расположения для развертывания и размещения служб. Контейнеры изолированы друг от друга и от базовой операционной системы. При этом они расходуют меньше ресурсов, чем виртуальная машина. Также контейнеры обеспечивают высокую пропускную способность и низкую задержку. Контейнеры не ограничивают число транзакций в секунду (TPS), их можно увеличивать и уменьшать для удовлетворения спроса, предоставив необходимые аппаратные ресурсы.

Text Analytics API [4] – это облачная служба, которая предоставляет расширенную обработку необработанного ЕЯ-текста и состоит из четырех модулей: анализ тональности, извлечение ключевых фраз, определение языка и распознавание сущностей. Эти модули можно использовать с REST API или с клиентской библиотекой для .NET, Python, Node.js, Go или Ruby. Процесс обработки представлен на рис. 3.

Рисунок 3 – Последовательность NLP-обработки в Microsoft Azure Cognitive Service

Модуль распознавания языка определяет единый код языка для каждого документа, представленного по запросу, на разных языках, диалектах и некоторых местных наречиях. Код языка сопряжен с показателем, указывающим степень оценки.

Модуль предварительной обработки вызывается перед выполнением каждого из модулей (кроме определения языка текста). Этим модулем проводится следующая предварительная обработка (предварительная обработка указывается пользователем, т.е. в случае, если предварительная нужна; а также пользователь выбирает, какие функции предварительной обработки должны быть применены):

- определение морфологической информации;
- удаление слов, относящихся к определенной части речи;
- удаление стоп-слов (список стоп-слов привязан к языку и может дополняться пользователем);
- лемматизация;
- определение границ предложения;
- приведение символов к нижнему регистру;
- обработка цифровых комплексов (цифро-буквенные комплексы не удаляются);
- удаление специальных символов, которые задаются пользователем;
- удаление дублирующих символов (например, “aaa”);
- удаление email, URL;
- развертывание глагольных конструкций (реализовано лишь для английского языка: «wouldn't stay there» = «would not stay there»);
- разделение слов на основе специальных символов, которые задаются пользователем.

Модуль анализа тональности использует необработанный входной текст для получения сведений о тональности выражений (положительная или отрицательная). Этот API возвращает оценку тональности (0 или 1) для каждого документа, где 1 означает положительную тональность. В настоящий момент API анализа тональности поддерживает английский, немецкий, испанский и французский языки. Другие языки находятся на этапе предварительной версии. Модель обучена с использованием обширного текста со связями тональности. В настоящее время нет возможности предоставления собственных данных для обучения. В модели во время анализа текста используется комбинация методов. Методы включают обработку текста, анализ частей речи, позицию слов и их векторное представление. Вышеупомянутые модели основаны на нейросетевом подходе [5].

Модуль извлечения ключевых фраз в тексте поможет быстро определить основные слова. Например, для входного текста «Еда была вкусной и персонал был замечательным» API вернет основные тезисы: «еда» и «замечательный персонал».

Модуль распознавания именованных сущностей определяет сущности и их распределение по категориям, таким как текст, люди, места, организации, дата и время, количество, проценты, валюта и многое другое. Также может определять известные сущности и связать их с дополнительной информацией в Интернете.

Платформа анализа данных IRELA

IRELA [6] – платформа анализа данных на основе Big Data, Machine Learning, Data Science. Платформа IRELA состоит из 9 модулей и позволяет решать широкий спектр задач. Имеется возможность предоставлять доступ как ко всем модулям, так и к конкретно выбранным (или их функциональным частям). То есть каждый модуль (кроме базового) является самостоятельным.

Базовый модуль лежит в основе платформы, и на него устанавливаются все остальные модули. Он собирает, нормализует и очищает данные, создает внутреннюю базу знаний компании, подключает внешние источники данных и позволяет осуществлять простой полнотекстовый поиск. В данном модуле осуществляется: сбор неструктурированных данных; нормализация и очистка данных; создание внутренней базы знаний; подключение внешних источников данных; простой полнотекстовый поиск.

Модуль поиска отвечает за поиск в неструктурированных данных компании и в подключаемых внешних базах, учитывая специальную терминологию. Например, модуль поможет найти предыдущие похожие разработки компании, что позволит сократить время на реализацию новых проектов и повысить их качество. В данном модуле осуществляется: семантический поиск с возможностью контекстного расширения; поиск в заданной предметной области с учетом отраслевой терминологии; интеллектуальная сортировка по релевантности.

Модуль мультиязычности позволяет искать по коллекциям документов на любых языках без необходимости ручного перевода запроса, что сокращает время поиска для пользователей, не владеющих нужными иностранными языками, и исключает ошибки перевода. Данный модуль осуществляет: работу с документами на разных языках без необходимости перевода; выделение тематик в текстовых коллекциях; построение графа связей между объектами текстов.

Модуль заимствований сравнивает документы на наличие содержательных пересечений, анализируя базу данных заказчика или подключенные базы. Также IRELA находит наиболее близкие документы, показывая пересекающиеся блоки и схожие отрывки текста. Внедрение этого модуля позволит сократить сроки проверки документов и проведения экспертизы, повысить оригинальность материалов. Данный модуль осуществляет: выявление прямых и косвенных заимствований; анализирует тематику документа и временные тренды; формирует список цитируемых источников; находит скрытые связи с персонами и организациями.

Модуль рекомендаций дает подсказки и формирует автоматические ответы, облегчая работу службы поддержки. Данный модуль позволяет: автоматическое формирование ответов на запросы; предлагать пользователю варианты ответов на внешние запросы на основе предыдущих ответов или базы знаний; учитывать результаты ранее принятых решений.

Модуль классификации необходим для маршрутизации документов: модуль самостоятельно анализирует текст, присваивает тему или категорию и привязывает к адресату. В данном модуле осуществляется: маршрутизация документов; классификация неструктурированных данных; отнесение документа к одному или нескольким разделам тематических классификаторов.

Модуль извлечения объектов находит в тексте ключевые элементы, на основе которых можно провести автореферирование текста (составить конспект документа). Алгоритм обнаруживает как стандартные типы объектов – имена, названия организаций, даты, номера телефонов, денежные суммы, так и специфичные для клиента,

такие как номера договоров. Модуль осуществляет: автозаполнение форм найденными в тексте словами нужного типа; выявление ключевых элементов из текста (имен, дат, сумм); обнаружение специфических сущностей; автореферирование текста.

Модуль эмоциональной окраски анализирует эмоциональную окраску текста и делает вывод, является ли она положительной, отрицательной или нейтральной. Автоматическое определение тональности текста поможет компаниям с большим потоком внешних обращений оценивать свою репутацию, отслеживать отзывы и своевременно реагировать на срочные вопросы. Данный модуль осуществляет: своевременное выявление и реакцию на срочные вопросы; быстрый анализ обратной связи/обращений клиентов; обучение системы для определения более сложных эмоциональных градаций.

Модуль визуализации – универсальный модуль, который позволяет визуализировать любые данные и создавать веб-формы. Модуль позволяет принимать управленческие решения станет проще благодаря мгновенному графическому сравнению данных по различным параметрам. Также в данном модуле осуществляется потоковая визуализация с фильтрацией данных по индивидуальным потребностям, что позволяет отслеживать процесс выполнения работ в реальном времени.

Единая система анализа видео, изображений, речи и текста IDOL

IDOL – это решение на основе ИИ для поисковой системы предприятия, чатботов, государственных аналитических ресурсов с открытым исходным кодом и анализа неструктурированных данных. IDOL [7-9] обеспечивает унифицированную аналитику текста, речи и видео с поддержкой более 1000 форматов данных. Система также обеспечивает доступ к 150 хранилищам данных (например, Documentum, Dropbox и т. д.), а также индексирует данные без перемещения и нарушения работы. Алгоритмы, применяемые в IDOL основаны на машинном обучении и глубоких нейронных сетях. При поиске информации и обнаружении знаний, используя набор данных на естественном языке, IDOL осуществляет формирование ответов на вопросы, используя предварительный опыт общения с пользователем. Это позволяет строить простые и контекстуальные диалоги между пользователем и системой. IDOL не опирается на глубокое знание грамматической структуры языка, а получает понимание через контекст употребления слов. Это особенно полезно при анализе разговорного или неформального языка, который не следует лингвистическим правилам NLP-систем. Процесс NLP-обработки в IDOL представлен на рис. 4.

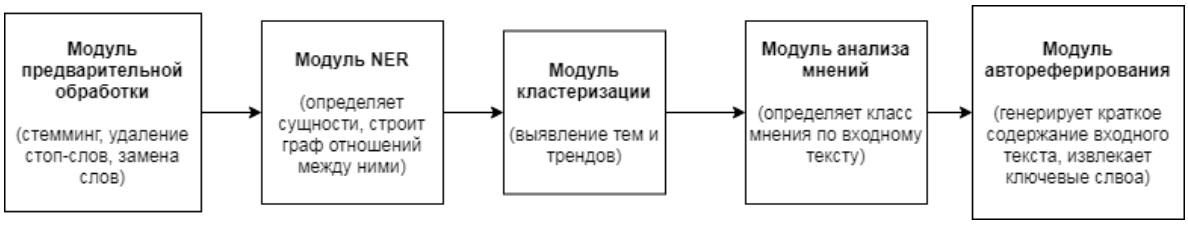

Рисунок 4– Последовательность NLP-обработки в IDOL

Модуль предварительной обработки осуществляет предварительную обработку, которая состоит из нескольких этапов:

– разбиение предложений и токенизация символов важны для языков, в которых используются слова, не разделенные пробелами. IDOL может быть легко настроен для разбивки текста на предложения и токенизации символов в n-граммы заданного размера с большой точностью;

– стемминг сводит слова с загнутыми корнями к их корню и позволяет IDOL группировать слова с похожими основными значениями. Это позволяет пользователям получать соответствующую информацию, даже если конкретная форма слова отсутствует в индексе. Например, запрос «бег» будет автоматически получать информацию о «кроссовках», а также о «бегунах» или «местах для бега»;

– удаление стоп-слов. Такие слова, как «а», «и» и «the» не имеют никакого концептуального значения. IDOL может автоматически определять такие слова и исключать их из анализа, чтобы повысить производительность и точность результатов;

– замена слов на их синонимы. Синонимы позволяют пользователям строить концептуальные отношения между словами и фразами. Когда пользователь ищет в системе слово «колледж», IDOL определяет, что колледж и университет представляют одну и ту же концепцию, и поэтому также автоматически ищет «университет». IDOL также может быть настроен для обработки аналогичных терминов, как гипонимы или гиперонимы.

В **модуле NER** система находит и классифицирует элементы в тексте по заранее определенным категориям, таким как имена и местоположение людей. IDOL использует грамматические методы для извлечения сущностей из любой части неструктурированной информации. Несколько основанных на грамматике методов доступны «из коробки» (например, имена, адреса, организации, номера телефонов или номера социального страхования). Кроме того, IDOL позволяет создавать и развертывать пользовательские объекты для достижения определенных целей.

Модуль кластеризации автоматически разбивает большой набор данных на части так, что сходная информация, даже в разных форматах данных, группируется вместе. Каждый кластер представляет собой концептуальную область, облегчая выявление тем и новых трендов.

Модуль анализа мнений определяет мнение авторов в документе. Комментарии людей могут быть сложными и многогранными, выражая резкую критику по одной теме и благодарность по другой в одном и том же тексте. В то время как традиционные технологии могут пропустить эти тонкости, возможности анализа настроений IDOL могут определять темы в тексте и классифицировать полярность для каждой темы как положительную, отрицательную или нейтральную.

Модуль автоматического автореферирования генерирует краткое резюме содержимого документа. IDOL имеет возможность создавать различные резюме:

– концептуальное резюме содержащее сведения о наиболее важных концепциях в документе;
– контекстное резюме, относящееся к исходному запросу;
– простое резюме, состоящее из нескольких предложений из начальной части документа.

Платформа текстовой аналитики Lexalytics Intelligence Platform

Lexalytics предоставляет анализ настроений и намерений множеству компаний, использующих SaaS и облачные технологии. В 2014 году Lexalytics приобрела Semantria с целью расширить клиентскую базу с многоязычной поддержкой. Продукт Semantria представляет собой сервис анализа текста SaaS, предлагаемый в виде плагина на основе API и Excel, который измеряет настроение.

В качестве основных алгоритмов для [10], [11] анализа текстовых данных Lexalytics использует алгоритмы машинного обучения. Процесс NLP-обработки в Lexalytics Intelligence Platform представлен на рис. 5.

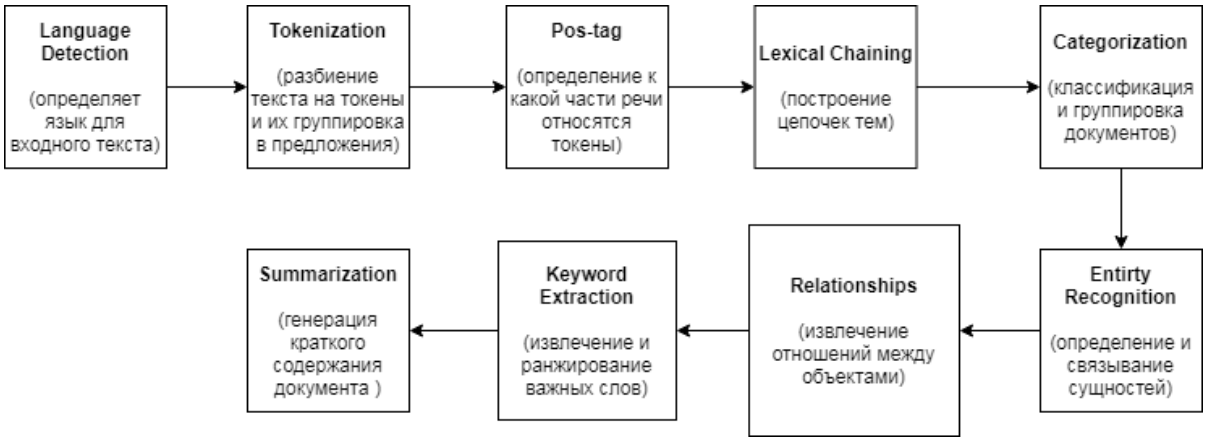

Рисунок 5 – Последовательность NLP-обработки в Lexalytics Intelligence Platform

Модуль Language Detection определяет язык для входного текста. Тексты могут быть смешанными, т.е. содержать фрагменты на разных языках. Модуль не пытается идентифицировать отдельные фрагменты, а определяет язык целого текста.

Модуль Tokenization осуществляет обработку текста – разбиение текста на токены и их группировку в предложения. Для большинства языков токенизация довольно проста: находятся пробелы, знаки препинания и т.п., и по ним выполняется разделение на токены. Однако у каждого языка есть свои особенности, например, немецкий широко использует составные слова, и для некоторых задач, таких, как классификация документа, актуальна токенизация до уровня подслов. В некоторых языках, таких, как китайский, нет пробелов между словами, и для маркировки этих языков требуется использование более сложных статистических моделей. Lexalytics использует модели токенизации для всех поддерживаемых языков.

Модуль POS-tag определяет морфологическую информацию. Большинство инструментов NLP-обработки и интеллектуального анализа текста используют не только набор токенов, но и части речи. В большинстве языков отдельные слова могут иметь несколько типов речи в зависимости от контекста, поэтому определение части речи для токена требует также оценки контекста, в котором появляется слово.

Lexalytics использует модели определения морфологической информации для большинства поддерживаемых языков и при необходимости возвращает теги POS вместе с выводом текста. В качестве стандарта для морфологической информации используется Penn Treebank.

Lexical Chaining используется для создания тематических резюме в тексте. Lexalytics использует расширение синонимов и другие методы для построения цепочек тем в текстах длиннее трех предложений. Эти цепочки формируют основу для дальнейших модулей NLP-обработки. При этом пользователь не может вывести эти цепочки.

Модуль Categorization отвечает за классификацию документов, а также их группировку. Категории могут быть различными: от общих категорий (например, спорт, политика, искусство и т.п.), и до специфичных (например, производство кремниевых пластин). Lexalytics использует различные способы категоризации документов.

1. Запросы. Логические поиски выполняются для всех документов, которые отправляет пользователь, если искомый текст совпадает с текстом документа, хранящегося в базе знаний. То есть классификация на основе запросов используется, когда сегмент текста легко обнаружить. Например, если пользователь ищет вхождения документов, где говорится о «iPhone», и эти вхождения есть в документах, то будет использован запрос. Недостатком использования запросов для категоризации является то, что для текстовых сегментов с большим объемом информации поиск сложен и отнимает много времени для создания подходящего запроса. Также данный процесс осложняется наличием синонимов, двусмысленных терминов и т. п. Например, если пользователь ищет документы о мобильных технологиях, нужно искать различные типы мобильных технологий, исключая те документы, которые говорят о мобильности и технологии отдельно и т. д.

2. Категории. Категории – это сохраненные поиски пользователя, построенные с использованием языка Concept Matrix. Они строятся для всех документов, которые отправляет пользователь. Каждая категория оценивается и получает процентное отношение релевантности к документу. Если оценка превышает пороговое значение для категории, возвращается название категории, оценка релевантности. Запросы по категориям используют концептуальную матрицу для расширения предоставленных условий. Например, категория «еда» хорошо зарекомендовала себя в документе, например, «на днях я съел несколько куриных крылышек, которые были просто потрясающими». Это делает категории хорошим выбором для соответствия более широким сегментам, таким как спорт, еда, искусство или технология. Недостатком использования категорий для категоризации является то, что этот метод не интуитивен. Пользователь может попасть на документы, в которых нет искомых терминов, и система не может сказать, почему здесь они не применяются. Кроме того, категории плохо работают с короткими текстами, такими, как переписки в соцсетях, смс-сообщения, т.к. категориям нужен довольно большой контекст для работы.

3. Автокатегории. Автокатегории – это категории, которые были созданы на основе таксономии Википедии. В Lexalytics реализовано около 4000 автокатегорий, и таксономия имеет три уровня. Стоит отметить, что таксономия и категории не могут быть изменены пользователем. Если одна из категорий попадает в документ, пользователю возвращается имя категории вместе с настроением, узлами-родителями категории, оценкой и URL-адресом страницы Википедии, представляющей категорию. Автокатегории предоставляют в основном широкие предметные области, такие, как сельское хозяйство, физика, компьютеры и т. п.

4. Машинное обучение. Машинное обучение – это метод категоризации, при котором вместо написания пользовательских запросов пользователь собирает набор

документов, каждый из которых помечается соответствующим списком. Как только у него есть набор данных, запускается процесс машинного обучения для создания статистической модели из документов. Например, если пользователь пометил много документов с помощью класса «мобильные телефоны», и у большинства из них было слово «iPhone», то система определит, что данное слово тесно связано с эти классом. Lexalytics поддерживает различные модели машинного обучения. В настоящее время они не поддерживаются в Semantria API.

Модуль Entity Recognition отвечает за распознавание сущностей. Модуль предоставляет пользователю возможность импортировать свой собственный словарь сущностей. Стоит отметить, что сущности могут быть использованы при поиске с использованием синтаксиса запроса (AND, OR, NOT, NEAR, WITH). В данном модуле также используется нормализация сущностей. Например, компании иногда упоминаются по их полному юридическому названию (Cisco Systems Inc), иногда по их части названия (Cisco), а также по их биржевому символу (CSCO). Стоит отметить, что пользователь может переопределить тип сущности, например пометить сущность «Samsung» не как «компания», а как «конкурент». Модуль Entity Recognition также осуществляет связывание сущностей. Например, множественные упоминания об объекте находятся и связаны друг с другом. Модуль связывает множественные упоминания об объекте и сообщает первые 3 их упоминания, упорядоченные по длине символов (например, «Барак Обама», «Обама» и «он»).

Модуль Relationships позволяет извлекает отношения между объектами, такими, как род занятий или компания, нанимающая кого-либо. Они извлекаются на основе текстовых шаблонов. Отношения предварительно определены и могут быть изменены только при непосредственном обращении к команде разработчиков.

Модуль Keyword extraction позволяет извлекать важные слова из фрагмента текста. В Lexalytics данный процесс также называется извлечением темы. Темы можно найти, ища фразы, которые соответствуют шаблонам частей речи, в основном описательные фразы с существительными, такие, как «вкусные морепродукты» или «морские гребешки». Модуль не выделяет отдельные слова. Когда система находит тему кандидата в тексте, ее актуальность рассматривается для всего документа. Если тема не имеет отношения к остальной части документа, она удаляется. Если она сохраняется, она получает оценку релевантности. Эта оценка полезна для ранжирования тем лишь в документе, а не между документами.

Модуль Summarization генерирует краткое содержание для всего документа, а также контекстуальные резюме извлеченных элементов, таких как сущности и темы. Semantria возвращает только краткое содержание документа. Резюме документа содержит наиболее важные предложения в документе.

Платформа анализа текста MonkeyLearn

Платформа MonkeyLean [12] позволяет обучить и интегрировать пользовательские и предварительно обученные модели машинного обучения для таких задач как классификация текста, анализ мнений и извлечение именованных сущностей. MonkeyLearn позволяет пользователям использовать REST API для получения доступа к платформе напрямую или через клиентскую библиотеку. Обеспечена поддержка для следующих языков программирования: Python, Ruby, PHP, Javascript, Java. В качестве основных форматов файлов, используемых для обработки и обучения, используются форматы CSV и Excel.

Как было указано выше, для получения результатов NLP-обработки имеется возможность предварительно обучить модель для каждого модуля (кроме модуля Preprocessing). Для этого предварительно необходимо загрузить данные для обучения непосредственно в платформу или же используя облачные сервисы (например, Google Drive). В случае если данные не размечены, то необходимо их аннотировать вручную. Например, для обучения модуля NER требуется вначале задать список сущностей, а затем аннотировать данные через платформу. Затем модель обучается и выводятся графики с результатами обучения, используя известные метрики (например, точность (precision) и полнота (recall)). После обучения пользователь может подавать массивы данных в платформу, получая результат NLP-обработки. Процесс NLP-обработки в MonkeyLearn представлен на рис. 6.

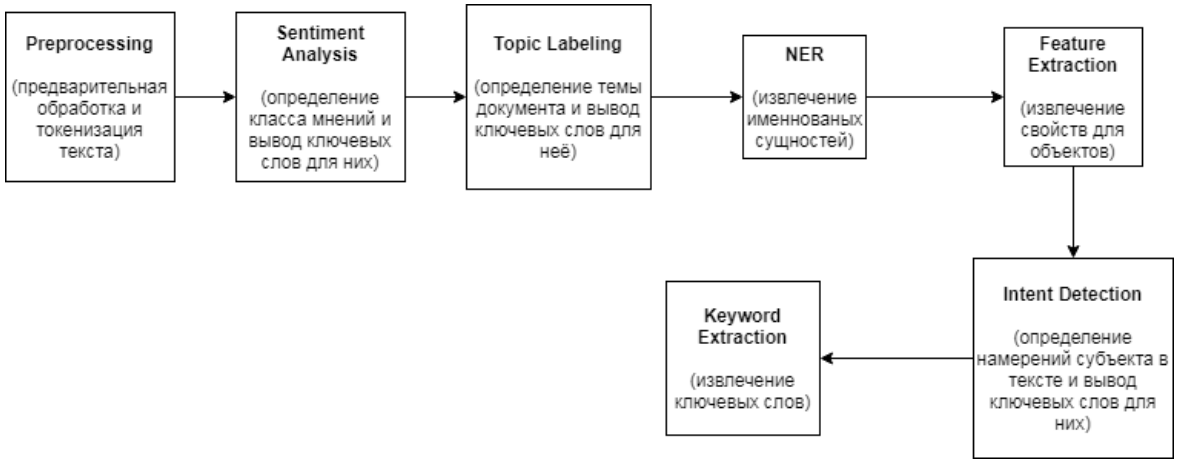

Рисунок 6 – Последовательность NLP-обработки в MonkeyLearn

Модуль Preprocessing реализует предварительную обработку текста (удаление стоп-слов, удаление неконтекстных символов) и его токенизацию.

Модуль Sentiment Analysis позволяет определять в входном тексте нейтральные или негативные мнения, а также выводить ключевые слова, которые определяют это мнение.

Модуль Topic Labeling определяет, к какой тематике относится текстовый массив, а также выводит список слов, извлечённых из данного текстового массива, которые относятся к данной тематике.

Модуль NER позволяет извлекать именованные сущности из текстового документа.

Модуль Feature Extraction использует тот же поход, что и модуль NER, но позволяет извлекать свойства для субъектов (например, размер экрана, название процессора, бренда).

Модуль Intent Detection позволяет определять намерения субъекта в тексте, т.е. действия, которые субъект собирается совершить с интересующим его объектом, а также вывод ключевых слов для данного интента.

Модуль Keyword Extraction позволяет извлекать важные слова из фрагмента текста.

Набор инструментов корпоративной текстовой аналитики Neticle Text Analysis

Neticle Text Analysis [13] позволяет классифицировать документы, определять эмоциональный тон документы и извлекать именованные сущности на основе ключевых слов (словарей). Стоит отметить, что пользователь может дополнять и добавлять словари ключевых слов, а также добавлять синонимы к ним. Neticle Text Analysis оценивает входные тексты, используя значения от -3 до +3 в зависимости от релевантности ключевых слов. Эта оценка носит название «индекс мнения». Процесс NLP-обработки в Neticle Text Analysis представлен на рис. 7.

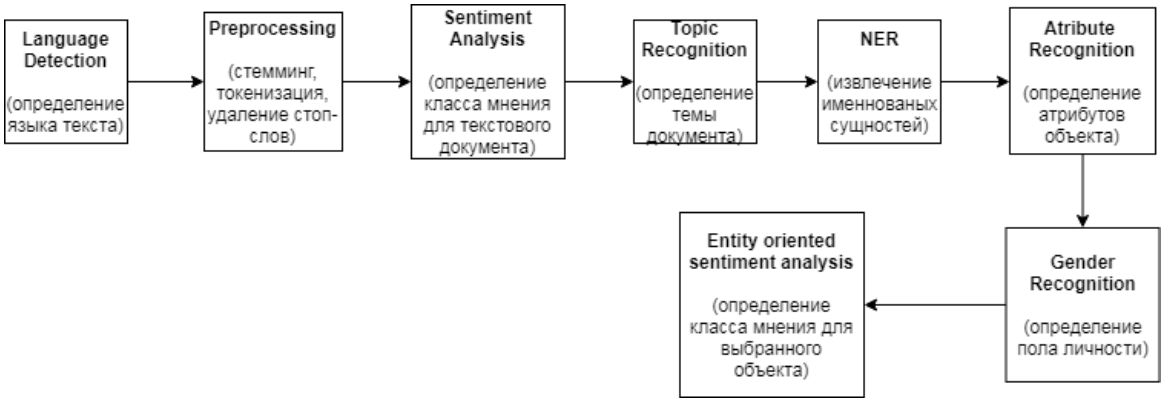

Рисунок 7 – Последовательность NLP-обработки в Neticle Text Analysis

Модуль Language recognition определяет язык для входного текста. На данный момент система поддерживает следующие языки: болгарский, грузинский, немецкий, английский, венгерский, польский, румынский, русский, украинский.

Модуль Preprocessing обеспечивает предварительную обработку текстовых массивов посредством удаления стоп-слов.

Модуль Sentiment analysis определяет позитивный, негативный или нейтральный тон документа.

Модуль Topic recognition отвечает за определение принадлежности текстового массива к определённой тематике. Данный модуль состоит из следующих блоков:

– Business topic recognition, определяет темы, относящиеся к бизнесу (например, IPO);

– Legal topic recognition, определяет юридические темы (например, иск, законодательство);

– Medical topic recognition, определяет темы, относящиеся к медицине (например, болезнь, рецепт);

– HR topic recognition, определяет темы, относящиеся к рекрутингу (например, работа, зарплата).

Модуль NER отвечает за извлечение именованных сущностей из входного текста. Данный модуль состоит из следующих блоков:

– Location recognition, извлекает сущности, относящиеся к местоположению (например, Венгрия, Москва);

– Brand recognition, извлекает сущности, относящиеся к названиям брендов (например, Audi, Mercedes);

– Emotion recognition, извлекает слова, относящиеся к эмоциональным состояниям (например, счастье);

– Person recognition, извлекает сущности, относящиеся к личностям (например, Джек Лондон);

– Organization recognition, извлекает сущности, относящиеся к организациям (например, UNICEF);

– Event recognition, извлекает сущности, относящиеся к событиям (например, фестиваль, конференция).

Модуль Attribute recognition позволяет определять свойства, принадлежащие объекту (например, размер экрана, диагональ).

Модуль Gender recognition определяет пол личности на основе анализа его имени.

В **модуле Entity oriented sentiment analysis** определяется класс мнения для выбранного объекта, при этом дополнительно анализируются его синонимы и варианты написания.

API для обработки естественного языка от Google

Google Cloud Natural Language API [14] использует REST API. Текстовые массивы могут быть переданы через запрос или загружены через Google Cloud Storage. Процесс NLP-обработки в Google Cloud Natural Language API представлен на рис. X.10.

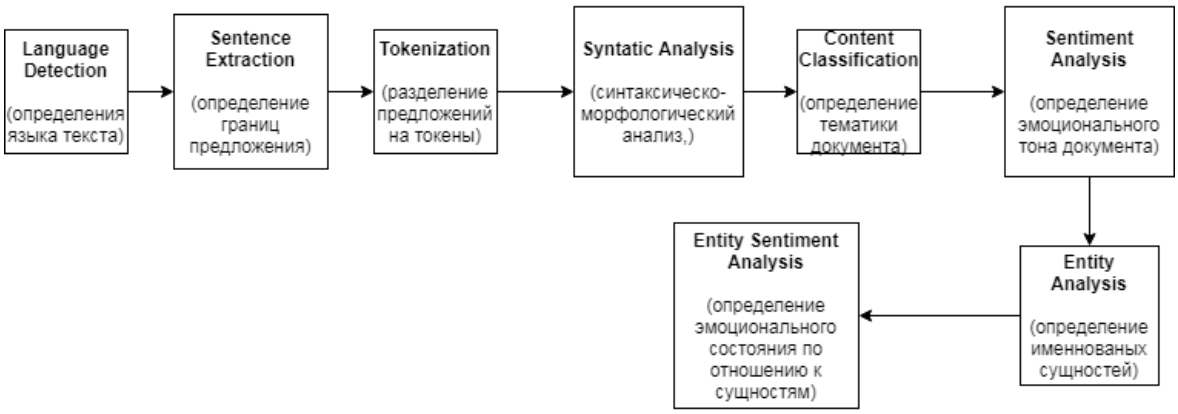

Рисунок 8 – Последовательность NLP-обработки в Google Cloud Natural Language API

Модуль Language Detection позволяет определять язык входного текста для более чем 50 языков.

Модуль Sentence Extraction определяет границы предложения.

Модуль Tokenization проводит разделение предложений на отдельные токены (слова, части слов, символы).

Модуль Syntatic Analysis использует информацию о токенах и их контекстном окружении, а также позиции в предложении и определяет морфологическую и синтаксическую информацию. Дополнительно в данном модуле может быть произведена лемматизация. Для каждого языка используется своя модель для определения морфологической и синтаксической информации.

Модуль Content classification анализирует текстовый контент и возвращает его тематику. Стоит отметить, что для выдачи результата текст должен содержать более 20 токенов.

Модуль Sentiment analysis просматривает входной текст и определяет преобладающее эмоциональное мнение в нём (позитивное, негативное или нейтральное). В результате данный модуль выдаёт числовое значение настроения и его магнитуду. Значение настроения указывает на общую эмоцию документа. Величина магнитуды настроения в документе указывает, сколько эмоционального содержимого присутствует в документе, и это значение часто пропорционально длине документа. Документ с нейтральной оценкой (около 0.0) может указывать на документ с низким уровнем эмоций или на смешанные эмоции с положительными, так и отрицательными эмоциями. Как правило, пользователь может масштабировать значения величин для устранения неоднозначности в этих случаях, поскольку у действительно нейтральных документов будет низкое значение, а у смешанных документов будут более высокие значения. Т.е. «явно положительное» и «явно отрицательное» настроение варьируется для разных вариантов сценариев. Поэтому пользователь должен определить порог, который ему подходит, а затем отрегулировать порог после тестирования и проверки результатов. Например, можно определить порог любой оценки свыше 0.25 как «явно положительный», а затем изменить порог оценки до 0,15 после просмотра данных и выдачи результатов.

Модуль Entity analysis извлекает тип сущности из входного текста, а также позволяет разрешать проблему анафор. Результат возвращается в виде набора обнаруженных сущностей и параметров, связанных с этими сущностями, такие как тип сущности, важность сущности для текущего документа (от 0.0 до 1.0) и положение в тексте, которые ссылаются на одну и ту же сущность. Объекты возвращаются в порядке (от наивысшего к низшему) их значимости, которые отражают их отношение к общему тексту. Дополнительно может быть возвращена ссылка на статью в Википедии для определённой сущности, а также может быть возвращено MID-значение соответствующее вхождению сущности в Google Knowlegde Graph. Стоит обозначить, что MID-значения уникальны для разных языков, поэтому дополнительно можно использовать такие значения для связывания сущностей из разных языков.

Модуль Entity sentiment analysis позволяет проверять входной текст на предмет известных сущностей, возвращает информацию об этих сущностях и выявляет преобладающее эмоциональное мнение о сущности в тексте. Настроение субъекта представлено числовой оценкой и значениями величины и определяется для каждого упоминания субъекта. Эти оценки затем агрегируются в общую оценку настроения и величину для субъекта. В результате выводятся все записи с содержанием сущности, которые были найдены в документе; упоминание записи для каждого вхождения сущности, а также числовые значения оценки и величина магнитуды для каждого упоминания сущности. Общие значения оценки и величины для объекта представляют собой совокупность значений конкретного показателя и величины для каждого упоминания объекта.

Также кроме Google Cloud Natural Language API компания Google реализовала систему Google AutoML Natural Language [15], обладающую таким же функционалом и структурой как вышеописанная платформа, но с одним ключевым отличием – Google AutoML Natural Language позволяет пользователям дополнительно обучить собственные модели на основе загружаемых в систему аннотированных текстовых массивов.

Фреймворк Saga Natural Language Understanding

Saga Natural Language Understanding (NLU) [16-18] позволяет пользователям создавать и поддерживать гибкие и масштабируемые модели для обработки текстовых массивов. Saga NLU объединяет множество методов моделирования языка и машинного обучения в единую, удобную для пользователя семантическую среду, позволяющую обрабатывать различные варианты использования естественного языка. Данная платформа поддерживает работу для 60 языков. Архитектура Saga NLU построена на Apache Spark и Spark MLlib. Процесс NLP-обработки в Saga NLU представлен на рис. 9.

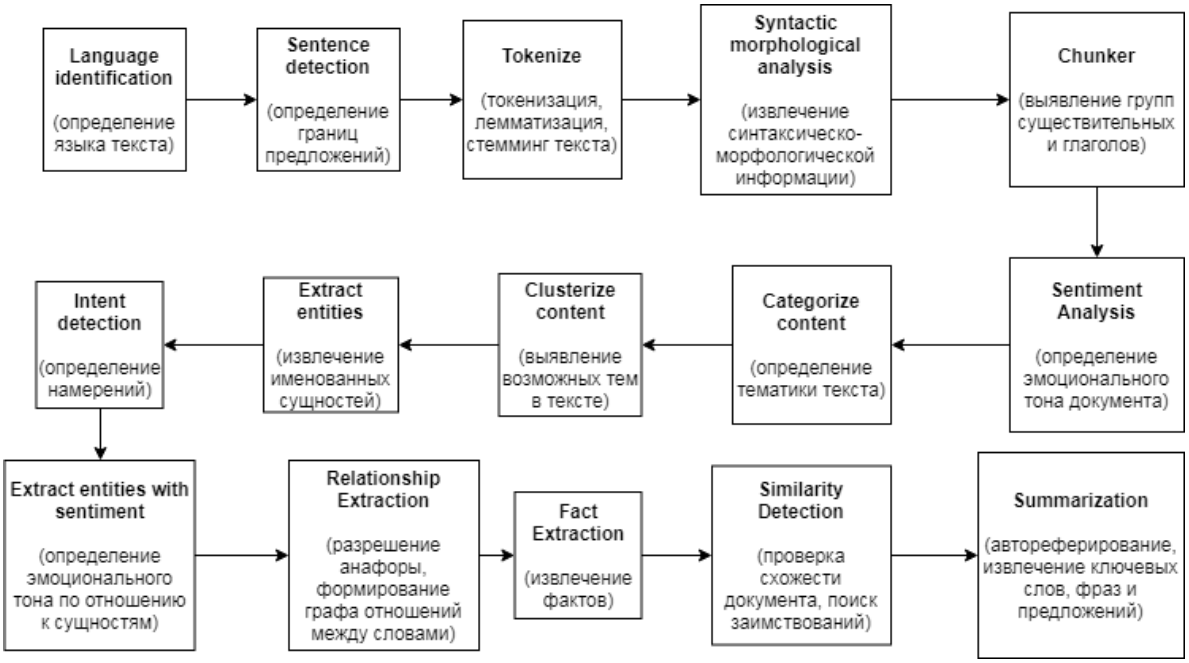

Рисунок 9 – Последовательность NLP-обработки в Saga NLU

Модуль Language identification определяет языка для всего документа, для каждого абзаца или предложения. В зависимости от определенного языка система затем определяет какие лингвистические алгоритмы и словари должны быть применены далее.

Модуль Sentence detection позволяет определять границы предложения для входного текста.

Модуль Tokenize позволяет разделять текстовый поток на токены, которые можно использовать для дальнейшей обработки и понимания. Токенами могут быть слова, цифры, идентификаторы или знаки пунктуации (в зависимости от варианта использования). Для ряда языков используется разделение на составные части слов. Дополнительно в этом модуле производится:

– нормализация аббревиатур;

– лемматизация и стемминг, которые позволяют уменьшить количество слов, что в ряде случаев позволяет получить более качественный результат в таких задачах как классификация текста. Для лемматизации используется словарь для конкретного языка, чтобы выполнить точное сокращение до корневых слов. Для стемминга используется простое сопоставление с образцом, чтобы удалить суффиксы токенов (например, удалить «-s», «-ing» и т. п.).

Модуль Syntactic morphological analysis позволяет получить морфологическую (части речи и т. п.), а также синтаксическую информацию (роли слов и их отношение).

Модуль Chunker группирует токены в группы существительных, группы глаголов.

Модуль Sentiment Analysis позволяет определять общий эмоциональный тон документа.

Модуль Intent detection позволяет определять намерения субъекта в тексте, т.е. действия, которые субъект собирается совершить с интересующим его объектом.

Модуль Categorize content позволяет определять тематику входного документа.

Модуль Clusterize content определяет основные темы в текстовом документе, в случае если они отсутствуют в модуле Categorizing content, т.е. определяет новые темы.

Модуль Extract entities извлекает именованные сущности из текста, используя несколько методов:

– регулярные выражения для извлечения таких сущностей как телефонные номера, ID, e-mail, URL и т.п.;

– на основе словарей для таких объектов как цвета, единицы измерения, размеры, сотрудники, бизнес-группы, названия лекарств и т. д.;

– на основе шаблонов для извлечения имен, фамилий и отчеств людей, а также названий компаний;

– статистический анализ для извлечения имен людей, компаний, географических объектов, которые ранее не были известны, а также для извлечения сущностей из хорошо структурированного текста (например, академического или публицистического текста).

Модуль Extract entities with sentiment обеспечивает возможность определять класс мнения по отношению к сущности, а также выводить эти вхождения из текстового документа.

Модуль Relationship extraction позволяет разрешить проблему связей между словами (анафору), а также определить граф для изучения отношений между словами в документе или документах.

Модуль Fact extraction извлекает структурированную информацию в виде факта посредством извлечения из текста именованных сущностей, которые связаны между собой какой-либо логической и заранее определенной связью для дальнейшего анализа и визуализации.

Модуль Similarity detection осуществляет поиск сходства между различными документами, а также обнаруживает плагиат.

Модуль Summarization осуществляет автореферирование текстового документа. Стоит отметить, что в модуле предусмотрена возможность извлекать ключевые предложения, а также ключевые слова / фразы из текстового документа.

Выводы

Бурное развитие методов глубокого обучения обусловило революцию в технологиях обработки текста. Если раньше самым разработанным компонентом лингвистического процессора считался морфологический анализ, то на сегодняшний день предложены архитектуры глубокий сетей, эффективно решающие такие задачи как: классификация и тематическая рубрикация текста, определение языка, тональной и эмоциональной окраски, выделение сущностей, извлечение мнений, синтез ответов на запросы пользователя и т.п. Соответствующие модули реализованы в рассмотренных выше NLP-системах, использующих машинное обучение. При этом для таких задач как классификация или кластеризация текста, определение эмоционального тона текста и т.п. NLP-системы после токенизации слов использовать лемматизацию или стемминг. Данная процедура, как правило, позволяет добиться лучшего результата для вышеуказанных задач.

Стоит отметить, что преобладающее число NLP-систем имеют в наличии модуль для извлечения ключевых слов, фраз и предложений для входного текстового документа, которые используются для классификации текста. Кроме того, большинство NLP-систем имеют модуль для извлечения связей между объектами. В данном модуле строится граф, позволяющий находить связи между объектами (например, обнаружить владельцев фирмы, а также их связи с другими компаниями).

При реализации некоторых модулей применяется гибридный подход, т.е. наряду с моделями машинного обучения используются словари, регулярные выражения, системы правил.

Проблема снятия омонимии и разрешения анафор решена далеко не во всех современных NLP-системах. Платформы NLP-обработки, в которых частично решена проблема омонимии, используют не только анализируемый токен, но и его контекст. Для языков подобных английскому необходимо использовать функционал развёртывания фраз (например, глагольных конструкций).

Практически все рассмотренные платформы предоставляют API или имеют открытый исходный код, что предоставляет разработчикам NLP-систем возможность реализовывать функции интеллектуальной обработки ЕЯ-текстов для своих задач.

Список литературы

1. СПЕЦИАЛЬНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ «3i NLP Platform». Описание применения [Электронный ресурс]. URL: http://dss-lab.ru/Content/docs/3i_NLP_Platform.pdf (дата обращения: 27.12.2020).
2. Apache OpenNLP Developer Documentation [Электронный ресурс]. URL: https://opennlp.apache.org/docs/1.9.1/manual/opennlp.html (дата обращения 27.12.2020).
3. Apache OpenNLP - The Apache Software Foundation! [Электронный ресурс]. URL: https://opennlp.apache.org (дата обращения: 27.12.2020).
4. What is the Text Analytics API? [Электронный ресурс]. URL: https://docs.microsoft.com/en-us/azure/cognitive-services/text-analytics/overview (дата обращения: 27.12.2020).
5. Introducing Text Analytics in the Azure ML Marketplace? [Электронный ресурс]. URL: https://blogs.technet.microsoft.com/machinelearning/2015/04/08/introducing-text-analytics-in-the-azure-ml-marketplace/ (дата обращения: 27.12.2020).
6. IRELA – платформа анализа данных [Электронный ресурс]. URL: https://irela.ru/ (дата обращения: 27.12.2020).
7. IDOL Data Sheet [Электронный ресурс]. URL: https://www.microfocus.com/media/data-sheet/idol_ds.pdf (дата обращения: 27.12.2020).
8. Closing the Gaps in Natural Language Processing [Электронный ресурс]. URL:https://www.microfocus.com/media/flyer/closing_the_gaps_in_natural_language_processing_flyer.pdf (дата обращения: 27.12.2020).
9. IDOL. Unified text analytics, speech analytics and video analytics [Электронный ресурс]. URL: http://anovamr.com/ru-ru/products/information-data-analytics-idol/overview (дата обращения: 27.12.2020).
10. Data Analytics with NLP & Text Analytics | Lexalytics [Электронный ресурс]. URL: https://www.lexalytics.com (дата обращения: 27.12.2020).
11. Semantria API documentation [Электронный ресурс]. URL: https://semantria-docs.lexalytics.com/docs (дата обращения: 27.12.2020).
12. MonkeyLearn. Create new value from your data [Электронный ресурс]. URL: https://monkeylearn.com (дата обращения: 27.12.2020).
13. Functions of Neticle text analysis [Электронный ресурс]. URL: https://api.neticle.com (дата обращения: 27.12.2020).
14. Natural Language API Basics [Электронный ресурс]. URL: https://cloud.google.com/natural-language/docs/basics (дата обращения: 27.12.2020).
15. Google Cloud AutoML Natural Language [Электронный ресурс]. URL: https://cloud.google.com/natural-language/automl/docs/ (дата обращения: 27.08.2019).
16. Technology Assets to Support Search and Big Data Analytics Projects | Search Technologies [Электронный ресурс]. URL: https://www.searchtechnologies.com/technology (дата обращения: 27.12.2020).

17. Saga Natural Language Understanding (NLU) Framework [Электронный ресурс]. URL: https://www.accenture.com/us-en/services/applied-intelligence/saga-natural-language-understanding (дата обращения: 27.12.2020).
18. Saga Natural Language Understanding (NLU) Framework | Search Technologies [Электронный ресурс]. URL: https://www.searchtechnologies.com/saga-natural-language-understanding-framework (дата обращения: 27.12.2020).
19. Пикалёв Я. С. Система автоматической генерации транскрипций русскоязычных слов-исключений на основе глубокого обучения [Текст] / Я. С. Пикалёв, Т. В. Ермоленко // Проблемы искусственного интеллекта. – 2019. – № 4(15). – С. 35–50.

References

1. SPECZIAL`NOE PROGRAMMNOE OBESPEChENIE «3i NLP Platform». Opisanie primeneniya [E`lektronny`j resurs]. URL: http://dss-lab.ru/Content/docs/3i_NLP_Platform.pdf (data obrashheniya: 27.12.2020).
2. Apache OpenNLP Developer Documentation [E`lektronny`j resurs]. URL: https://opennlp.apache.org/docs/1.9.1/manual/opennlp.html (data obrashheniya 27.08.2019).
3. Apache OpenNLP - The Apache Software Foundation! [E`lektronny`j resurs]. URL: https://opennlp.apache.org (data obrashheniya: 27.12.2020).
4. What is the Text Analytics API? [E`lektronny`j resurs]. URL: https://docs.microsoft.com/en-us/azure/cognitive-services/text-analytics/overview (data obrashheniya: 27.12.2020).
5. Introducing Text Analytics in the Azure ML Marketplace ? [E`lektronny`j resurs]. URL: https://blogs.technet.microsoft.com/machinelearning/2015/04/08/introducing-text-analytics-in-the-azure-ml-marketplace/ (data obrashheniya: 27.12.2020).
6. IRELA – platforma analiza danny`kh [E`lektronny`j resurs]. URL: https://irela.ru/ (data obrashheniya: 27.12.2020).
7. IDOL Data Sheet [E`lektronny`j resurs]. URL: https://www.microfocus.com/media/data-sheet/idol_ds.pdf (data obrashheniya: 27.12.2020).
8. Closing the Gaps in Natural Language Processing [E`lektronny`j resurs]. URL: https://www.microfocus.com/media/flyer/closing_the_gaps_in_natural_language_processing_flyer.pdf (data obrashheniya: 27.12.2020).
9. IDOL. Unified text analytics, speech analytics and video analytics [E`lektronny`j resurs]. URL: http://anovamr.com/ru-ru/products/information-data-analytics-idol/overview (data obrashheniya: 27.12.2020).
10. Data Analytics with NLP & Text Analytics | Lexalytics [E`lektronny`j resurs]. URL: https://www.lexalytics.com (data obrashheniya: 27.12.2020).
11. Semantria API documentation [E`lektronny`j resurs]. URL: https://semantria-docs.lexalytics.com/docs (data obrashheniya: 27.08.2019).
12. MonkeyLearn. Create new value from your data [E`lektronny`j resurs]. URL: https://monkeylearn.com (data obrashheniya: 27.12.2020).
13. Functions of Neticle text analysis [E`lektronny`j resurs]. URL: https://api.neticle.com (data obrashheniya: 27.12.2020).
14. Natural Language API Basics [E`lektronny`j resurs]. URL: https://cloud.google.com/natural-language/docs/basics (data obrashheniya: 27.12.2020).
15. Google Cloud AutoML Natural Language [E`lektronny`j resurs]. URL: https://cloud.google.com/natural-language/automl/docs/ (data obrashheniya: 27.12.2020).
16. Technology Assets to Support Search and Big Data Analytics Projects | Search Technologies [E`lektronny`j resurs]. URL: https://www.searchtechnologies.com/technology (data obrashheniya: 27.12.2020).
17. Saga Natural Language Understanding (NLU) Framework [E`lektronny`j resurs]. URL: https://www.accenture.com/us-en/services/applied-intelligence/saga-natural-language-understanding (data obrashheniya: 27.12.2020).
18. Saga Natural Language Understanding (NLU) Framework | Search Technologies [E`lektronny`j resurs]. URL: https://www.searchtechnologies.com/saga-natural-language-understanding-framework (data obrashheniya: 27.12.2020).
19. Pikalyov Ya. S., Yermolenko T. V. Sistema avtomaticheskoy generatsii transkriptsiy russkoyazychnykh slov-isklyucheniy na osnove glubokogo obucheniya [System of automatic transcription generation of Russian-language words exceptions on the basis of deep learning] *Problemy iskusstvennogo intellekta* [Problems of Artificial Intelligence], 2019, No 4 (15), S. 35–50.

RESUME

Ya. S. Pikalyov

Overview of architectures of systems for intelligent processing of natural language texts

NLP systems are widely used in everyday business processes related to speech recognition and synthesis, machine translation, annotation and classification of documents, determining the sentiment and emotional coloring of text, organizing a dialogue with a user in natural language, etc.

Today, automatic text processing technologies are experiencing a rise associated with the rapid development of deep learning methods, on the basis of which modern NLP systems are implemented.

The article discusses the most popular platforms for intelligent processing of natural language texts, lists their architectures and describes the capabilities that they provide to the user.

In the systems considered, the problem of determining the language of the text, lemmatization and tokenization, determining the boundaries of a sentence, morphological analysis, highlighting named entities, classifying documents or categorizing them by keywords is solved. Most NLP systems have a module for extracting connections between objects, determining the emotional tone of the text, and auto-referencing.

Along with machine learning models, when implementing some modules, NLP systems use dictionaries, regular expressions, rule systems, i.e. hybrid approach.

Almost all the platforms considered provide an API or have an open source code, which provides developers of NLP systems with the ability to implement functions for intelligent processing of NL texts for their tasks.

РЕЗЮМЕ

Я. С. Пикалёв

Обзор архитектур систем интеллектуальной обработки естественно-языковых текстов

NLP-системы получили широкое распространение в повседневных бизнес-процессах, связанных с задачами распознавания и синтеза речи, машинного перевода, аннотирования и классификации документов, определения тональности и эмоциональной окраски текста, организации диалога с пользователем на естественном языке и т.п.

На сегодняшний день технологии автоматической обработки текста переживают подъем, связанный с бурным развитием методов глубокого обучения, на основе которых реализованы современные NLP-системы.

В статье рассмотрены наиболее популярные платформы для интеллектуальной обработки естественно-языковых текстов, приведены их архитектуры и описаны возможности, которые они предоставляют пользователю.

В рассмотренных системах решена задача определения языка текста, лемматизации и токенизации, определения границ предложения, морфологического анализа, выделения именованных сущностей, классификации документов или их рубрикации по ключевым словам. Большинство NLP-систем имеют модуль для извлечения связей между объектами, определения эмоционального тона текста, автореферирования.

Наряду с моделями машинного обучения при реализации некоторых модулей NLP-системы используют словари, регулярные выражения, системы правил, т.е. гибридный подход.

Практически все рассмотренные платформы предоставляют API или имеют открытый исходный код, что предоставляет разработчикам NLP-систем возможность реализовывать функции интеллектуальной обработки ЕЯ-текстов для своих задач.

Статья поступила в редакцию 19.10.2020.