ОПРЕДЕЛЕНИЕ АКТУАЛЬНОСТИ ПУБЛИКАЦИЙ С ИСПОЛЬЗОВАНИЕМ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ

МОСАЛОВ ОЛЕГ ПЕТРОВИЧ

Государственное бюджетное образовательное учреждение высшего образования Московской области «Технологический университет»,

кандидат физико-математических наук

Аннотация. Описывается информационная система, спроектированная с использованием методов машинного обучения, которая позволяет по тексту научной статьи определить актуальность данной стать и сформировать список рекомендаций для автора, направленных на повышение актуальности. Даётся описание отдельных методов и использующих их инструментов, на базе которых построены отдельные блоки рассматриваемой информационной системы.

Ключевые слова: рекомендательные системы; машинное обучение; обработка естественного языка; статистический анализ; кластеризация; автоматизация образовательных и научно-исследовательских процессов.

EVALUATION OF THE RELEVANCE OF PUBLICATIONS USING MACHINE LEARNING METHODS

Annotation. The article describes an information system designed using machine learning methods that allows to evaluate the relevance of a scientific article by its text and form a list of recommendations for the author aimed to the relevance increase. The description of individual methods and tools, on the basis of which separate blocks of the considered information system are built, is given.

Keywords: recommendation systems; machine learning; natural language processing; statistical analysis; clustering; automation of educational and research processes.

1. ВВЕДЕНИЕ

В настоящее время любое высшее учебное заведение сталкивается с огромным потоком публикаций: студенты, аспиранты, научные работники и преподаватели в рамках процессов учёбы и научной-исследовательский работы публикуют свои результаты. Общее количество публикаций на русском языке, зарегистрированных в Web of Science и Scopus в период с 2007 года по 2016 год не изменялось, хотя распределение по темам и претерпевало изменения [3]. При этом, взаимная цитируемость научных статей остаётся на низком уровне.

Безусловно, русский язык публикации накладывает ограничения на количество её потенциальных читателей и, как следствие, потенциальных цитирований, но достаточно часто появляются публикации, которые активно ссылаются на англоязычные источники по той причине, что аналогичные материалы отсутствуют на русском.

Ранее уже делались успешные попытки применения методов машинного обучения для работы с текстами на русском языке и их классификации [1, 2].

В настоящей работе предлагается подход к проектированию и описывается реализация информационной системы, которая позволяет по тексту научной статьи определить её актуальность на данный момент времени и предложить действия, направленные на увеличение актуальности статьи.

Описываемый подход является реализацией системы поддержки принятия решений на основе статистического анализа текста статьи, алгоритмов машинного обучения с использованием корпуса текстов ранее изданных научных статей и анализа популярности отдельных тем в сети Интернет.

Статья имеет следующую структуру: в разделе 2 описывается формальная постановка задачи, раздел 3 посвящён описанию используемых методов и инструментов, раздел 4 содержит подробное описание информационной системы, решающей поставленную задачу. В разделе 5 приведены полученные на данный момент результаты, сформулированы выводы по работе и описаны направления дальнейших исследований.

1. ПОСТАНОВКА ЗАДАЧИ

Задача описываемой в данной статье информационной системы состоит в том, чтобы, получив от пользователя текст научной статьи на русском языке, вернуть рекомендации по изменению статьи для повышения её актуальности. При этом в процессе своей работы, при необходимости, информационная система может осуществлять взаимодействие с пользователем.

Все входные и выходные данные должны сохраняться для возможности последующего анализа. Все взаимодействия пользователя и информационной системы должны производиться посредством графического интерфейса.

Далее по тексту статьи указанные ниже термины употребляются в следующем смысле: «ключевые слова» – набор словосочетаний и отдельных слов, описывающих основное, высокоуровневое содержание текста; «кандидат в ключевые слова» – слово или словосочетание, целесообразность включения которого в ключевые слова рассматривается в процессе работы информационной системы; «ключевое словосочетание» – отдельная целостная компонента ключевых слов, может быть сочетанием или отдельным словом. Таким образом, «ключевые слова» – это набор «ключевых словосочетаний».

1. ИНСТРУМЕНТЫ

В рамках отдельных функциональных блоков описываемой информационной системы используются нижеперечисленные инструменты и методы.

*Первый метод* – это извлечение из текста кандидатов в ключевые слова с помощью статистического анализа и вычисление относительной важности входящих в них слов [4]. Для определения величины важности отдельного слова в тексте используются следующие правила:

1. Если слово начинается с заглавной буквы, но при этом не стоит в начале предложения, либо если слово полностью состоит из заглавных букв – величина его важности увеличивается.
2. Частота нахождения слова ближе к началу текста приводит к увеличению величины его важности.
3. Высокая частота употребления слова в тексте приводит к увеличению его важности.
4. Большая величина взаимосвязи слова с другими словами в тексте, а именно – количество различных слов, встречающихся в некоторой окрестности данного слова, приводит к увеличению его важности.
5. Высокая частота употребления слова в различных предложениях приводит к увеличению его важности.

Затем на основании важности отдельных слов формируются кандидаты в ключевые слова.

В качестве инструмента, реализующего первый метод, используется написанный на языке программирования Python сервис Yet Another Keyword Extractor [8]. Сильной стороной данного инструмента является то, что реализованный алгоритм не зависит от конкретного языка, таким образом, данный инструмент может быть использован для анализа статей, написанных на русском языке, без значительных изменений. Однако, инструмент требует настройки набора гиперпараметров, так как значения по умолчанию больше подходят для текстов на английском языке.

*Второй метод* – корректировка списка кандидатов в ключевые слова с помощью алгоритма машинного обучения. В данном подходе используется алгоритм обучения с учителем для задачи бинарной классификации. Для этого формируется обучающая выборка *D*, каждая запись которой *di* представляет собой кортеж (*A*, *w*, *e*), где *A* – текст статьи, *w* – ключевое словосочетание, *e* – оценка пользователя. На этой выборке обучается нейросетевая модель, которая в дальнейшем используется как функция аппроксимации *f*, относящая кортежи (*A*, *w*, *e*) к одному из двух классов: *X* – «принято пользователем» или *Y* – «отвергнуто пользователем», то есть *f*((*A*, *w*, *e*))∈{*X*, *Y*}.

В качестве инструмента, реализующего второй метод, используется искусственная нейронная сеть, реализованная на языке программирования Python с помощью библиотеки Keras [6]. Для построения цифрового представления текстовой информации возможно использовать заранее настроенные представления русскоязычных слов, такие как Navec от проекта Natasha [5], либо один из вариантов, подготовленных проектом DeepPavlov.ai [7].

*Третий метод* – построение списка тем текста на основании ключевых слов текста. Множество ключевых словосочетаний *W*, используемых в научных публикациях, очень велико, причём с появлением новых текстов мощность данного множества |*W*| только увеличивается. Множество же тем *T*, которое формируется в рамках данного метода, ограничено и их количество контролируется в рамках работы рассматриваемой информационной системы. Таким образом, метод заключается в реализации функции *g*, преобразующей подмножество потенциально бесконечного множества в подмножество другого, конечного множества: *g*: *W* → *T*.

В качестве инструмента, реализующего третий метод, используется алгоритм, реализованный на языке программирования Java, который использует таблицу соответствия между ключевыми словосочетаниями и темами текста, реализованную на базе данных MS SQL. Каждому ключевому словосочетанию *w* соответствует строка, каждой теме *t* – столбец данной таблицы. Используя значения в ячейках таблицы *rij* как веса: чем выше значение, тем больше соответствие между ключевым словосочетанием *w* и темой *t*, алгоритм формирует список тем {*t*1,…, *tM*} из ключевых слов {*w*1,…, *wN*}, при этом для каждой темы *ti* в списке вычисляется величина её соответствия тексту *c*, как сумма *rij* для *j* = 1,…, *N*.

Для определения списка используемых в таблице тем *t* и значений *r* в ячейках таблицы используется алгоритм кластеризации *k*-средних на множестве цифровых представления ключевых словосочетаний *w*, полученных с помощью одного из вышеописанных источников [5,7]. Центр каждого найденного кластера соотносится с темой, название которой определяется с использованием экспертного мнения. Соответствие *r* между ключевым словосочетанием *w* и темой *t* определяется как величина, обратная расстоянию между их цифровыми представлениями.

*Четвёртый метод* – определение актуальности темы по её названию. Актуальность темы *p* трактуется как величина, прямо пропорциональная количеству документов, хранимых в сети Интернет, в которых упоминается данная тема.

Инструментом, реализующим четвёртый метод, является написанный на языке программирования Java с использованием базы данных MS SQL алгоритм, который по названию темы *t* получает количество документов в сети Интернет, связанных с данной темой, и сохраняет это количество в качестве значения актуальности *p* с привязкой к теме *t* в базу данных.

*Пятый метод* – алгоритм формирования рекомендаций для автора научной статьи. Рекомендации состоят из двух частей: первая часть – указание тем, которые наиболее точно соответствуют тексту статьи, и соответствующих им значений актуальности *p*; вторая часть – указания тем документа, актуальность *p* которых высока, а соответствие тексту *c* низкое, чтобы автор мог принять решение о необходимости доработки текста статьи.

В качестве инструмента, реализующего пятый метод, используется алгоритм, написанный на языке программирования Java. В качестве входных данных используется список кортежей (*t*, *c*, *p*), где *t* – тема, *c* – соответствие темы *t* тексту, *p* – актуальность темы *t*. Выходные данные для обеих частей рекомендаций представляют отсортированные списки переданных на вход кортежей, для первой части сортировка производится в порядке убывания значения параметра *c*; для второй части – в порядке убывания значения выражения *αc*-1+*βp*, где *α* и *β* – нормировочные коэффициенты.

1. ОПИСАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Рассматриваемая информационная система использует все перечисленные в разделе 3 методы и инструменты для решения задачи, описанной в разделе 2. Описание работы информационной системы разбито на две части: первая часть содержит подготовительную часть, которая должна быть выполнена хотя бы один раз до начала эксплуатации системы; вторая часть содержит непосредственно процесс эксплуатации, то есть взаимодействия пользователя с информационной системой и действий, происходящих в системе при этом.

Первая, подготовительная, часть описания работы информационной системы состоит из следующих шагов: настройка гиперпараметров сервиса Yet Another Keyword Extractor (первый метод), обучение искусственной нейронной сети для коррекции списка ключевых словосочетаний (второй метод), формирования списка тем и их соответствия ключевым словосочетаниям, включая подборку гиперепараметров для кластеризации ключевых словосочетаний (третий метод).

В базовом варианте сервиса Yet Another Keyword Extractor для настройки доступно несколько параметров. Первый параметр – lan, отвечающий за язык текста, от которого зависит, какой список стоп-слов, т.е. слов, считающихся служебными и исключающихся из кандидатов в ключевые слова, будет использован. В текущей версии информационной системы всегда используется значение «ru», т.е. предполагается работа с русскоязычными текстами. Второй параметр – maxNgramSize, определяет максимальное количество отдельных слов в одном ключевом словосочетании. Рекомендуемое значение по умолчанию для английского языка – 3, для русского же языка в результате проведённых экспериментов было выбрано значение 5. Третий параметр – top, определяет количество потенциальных ключевых слов, которые вернёт сервис. В настоящее время рассматриваемая информационная система использует значение 10 для данного параметра.

Для обучения искусственной нейронной сети необходима размеченная обучающая выборка, т.е. набор текстов научных статей и ключевых слов для каждой их них. При этом размер такой выборки должен быть не менее нескольких тысяч, чтобы статистические закономерности, которые зафиксирует нейронная сеть, были устойчивыми. Для эффективной работы данного блока информационной системы нужно провести обучение на выборке размером не менее, чем в несколько десятков тысяч. В рамках данного шага подготовки системы к эксплуатации используются два подхода по формированию выборки: первый – сбор и сохранение данных о статьях, которые загружены пользователями в систему, второй – ручная доразметка статей, имеющихся в общем доступе. При появлении новой выборки размеченных текстов достаточного размера возможно дообучение используемой искусственной нейронной сети.

Чтобы использовать алгоритм кластеризации *k*-средних для формирования списка тем, требуется список ключевых слов размеров минимум в несколько сотен, причём ключевые слова должны быть взяты из научных статей различной тематики. Безусловно, список ключевых слов большего размера приведет к более надёжным результатам. Значение величины *k* выбирается не более 10% от размера списка ключевых слов, при этом максимальное значение, используемое в текущей версии информационной системы, равно 200.

Вторая, основная, часть описания работы информационной системы приведена ниже.

Пользователь с помощью графического интерфейса загружает текст *A* научной статьи на русском языке в информационную систему, которая, используя первый описанный в разделе 3 метод – статистический анализ текста, извлекает 10 фраз длинной не более 5 слов, которые трактуются как кандидаты в ключевые слова данной статьи.

Данный список подергается корректировке с применением второго из описанных в разделе 3 методов – искусственной нейронной сети, в результате чего для каждого из кандидатов в ключевые слова определяется его класс. Те пункты списка, которые отнесены к классу «отвергнуто пользователем», исключаются из списка.

Полученный список кандидатов в ключевые слова выводится в графический интерфейс, после чего пользователь может внести следующие изменения в список: удалить те пункты, которые, на его взгляд, не соответствуют тексту статьи или являются слишком общими, или добавить дополнительные ключевые словосочетания длиной не более 5 слов. Закончив редактирование списка пользователь подтверждает его, после чего текст статьи и финальный список ключевых словосочетаний сохраняются в базу данных, в том числе, для дальнейшего дообучения искусственной нейронной сети.

Затем информационная система с использованием третьего метода, описанного в разделе 3, – таблицы весов соответствия между ключевыми словосочетаниями *w* и темами *t*, для каждого из ключевых словосочетаний в списке формирует вектор весов *r* соответствия темам. Если ключевое словосочетание *w* явно присутствует в таблице соответствия, то в качестве компонент вектора весов используются значения ячеек из строки, соответствующей ему. Если ключевое словосочетание отсутствует в таблице, то, используя цифровое представление ключевых словосочетаний, оно раскладывается в линейную комбинацию ключевых словосочетаний, имеющихся в таблице, и компоненты вектора весов вычисляются как линейные комбинации соответствующих весов имеющихся в таблице ключевых словосочетаний. Также отсутствующее ключевое словосочетание и рассчитанные таким образом веса сохраняются в базу данных, для дальнейшего использования при очередной кластеризации ключевых словосочетаний для определения тем.

Матрица, полученная из векторов весов *r* соответствия темам, транспонируется и для каждой темы *t* веса суммируются, таким образом формируется список (*t*, *c*), где *t* – тема, *c* – соответствие темы *t* анализируемому тексту.

Для каждой темы, используя четвёртый метод, описанный в разделе 3 – определение актуальности темы по количеству её упоминаний в сети Интернет, из списка кортежей (*t*, *c*) формируется список кортежей (*t*, *c*, *p*), где *p* – актуальность темы *t*.

В итоге информационная система, используя пятый описанный в разделе 3 метод – сортировку полученного списка кортежей, выводит в графический интерфейс рекомендации: первая часть в виде наиболее актуальных тем анализируемого текста в виде пар (*t*, *p*), отсортированных так, что первыми выводятся наиболее актуальные темы; вторая часть в виде актуальных тем, которые недостаточно раскрыты в анализируемом тексте, в виде кортежей (*t*, *c*, *p*), отсортированных так, что первыми выводятся темы с высоким значением актуальности, но с низким, однако ненулевым, значением соответствия анализируемому тексту *A*.

1. ЗАКЛЮЧЕНИЕ

Описываемая в данной работе информационная система относится к классу рекомендательных систем, позволяющих на основе сбора большого количества данных и их статистического анализа создавать информационную поддержку пользователей в процессе принятия ими решений. В данном случае основными пользователями системы являются авторы научных статей на русском языке, который ставят перед собой задачу повысить актуальность собственных работ и таким образом увеличить их цитируемость.

Рассматриваемая информационная система на данный момент находится в процессе опытной эксплуатации в рамках функциональности внутреннего информационного портала ГБОУ ВО МО «Технологический университет». В рамках опытной эксплуатации показано, что разработанная информационная система решает поставленную задачу.

После перевода системы в промышленную эксплуатацию, накопления достаточного объёма обратной связи и сбора дополнительных данных возможно внесение изменений в логику работы информационной системы и в список используемых методов и инструментов. В рамках уже запланированного развития системы будут проводиться эксперименты, использующие многовариантное тестирование отдельных блоков, с целью повышения качества, скорости и надёжности работы как отдельных блоков, так и информационной системы в целом.

Логика работы отдельных блоков системы, таких как излечение кандидатов в ключевые слова из текста статьи, коррекция списка кандидатов в ключевые слова, формирование списка тем текста из списка ключевых словосочетаний и определение актуальности отдельных тем, не зависит друг от друга. Это позволяет рассматривать эти блоки информационной системы по-отдельности, в том числе изменять, настраивать, оптимизировать данные блоки и реализовывать их в виде отдельных микросервисов, которые могут быть использованы как составные части других информационных систем, решающих свои задачи.

СПИСОК ЛИТЕРАТУРЫ

1. *Агеев М. С., Добров Б. В., Лукашевич Н. В.* Автоматическая рубрикация текстов: методы и проблемы // *Ученые записки Казанского университета. Серия Физико-математические науки.* — 2008. — Т. 150, № 4. — С. 25–40.
2. *Краснянский М.Н., Обухов А.Д., Соломатина Е.М., Воякина А.А.* Сравнительный анализ методов машинного обучения для решения задачи классификации документов научно-образовательного учреждения // *Вестник Воронежского государственного университета. Серия Системный анализ и информационные технологии.* — 2018. — № 3. — С. 173–182.
3. *Москалёва О.В., Акоев М.А.* Публикации на разных языках в индексах цитирования, или Есть ли шанс у русского языка в науке? // *Университетская книга.* — 2018. — № 3. — С. 42–45.
4. *Campos R., Mangaravite V., Pasquali A., Jorge A.M., Nunes C., and Jatowt A.* A Text Feature Based Automatic Keyword Extraction Method for Single Documents. // *In: Pasi G., Piwowarski B., Azzopardi L., Hanbury A. (eds). Advances in Information Retrieval. ECIR 2018 (Grenoble, France. March 26 – 29). Lecture Notes in Computer Science*. — 2018. — Vol. 10772. — PP. 684–691.
5. Compact high quality word embeddings for Russian language [Электронный ресурс] // URL: https://github.com/natasha/navec (дата обращения: 03.09.2020).
6. Keras: Deep Learning for humans [Электронный ресурс] // URL: https://github.com/keras‑team/keras (дата обращения: 10.02.2020).
7. Pre-trained embeddings. DeepPavlov. [Электронный ресурс] // URL: http://docs.deeppavlov.ai/en/master/features/pretrained\_vectors.html (дата обращения: 04.03.2020).
8. Yet Another Keyword Extractor (Yake). Unsupervised Approach for Automatic Keyword Extraction using Text Features. [Электронный ресурс] // URL: https://github.com/LIAAD/yake (дата обращения: 09.06.2020).