УДК 004.891

Применение методов кластеризации и машинного обучения для построения рекомендательной системы для определения актуальности научных публикаций

О.П. Мосалов, кандидат физико-математических наук, заведующий учебно-научной лабораторией интеллектуальных систем управления, Государственное бюджетное образовательное учреждение высшего образования Московской области «Технологический университет имени дважды Героя Советского Союза, лётчика-космонавта А.А. Леонова», г. Королёв, Московская область,

И.А. Иванов, бакалавр по специальности «Прикладная информатика», разработчик программного обеспечения,

Публичное акционерное общество «Сбербанк России», г. Москва,

 М.А. Першин, бакалавр по специальности «Прикладная информатика», студент группы ИО-17,

Государственное бюджетное образовательное учреждение высшего образования Московской области «Технологический университет имени дважды Героя Советского Союза, лётчика-космонавта А.А. Леонова», г. Королёв, Московская область

*В статье представлено решение задачи построения рекомендательной системы для определения актуальности научных публикаций. Рассматривается алгоритм определения актуальностей, а также применяемые методы машинного обучения для обработки текстовой информации. Рассмотрено и обосновано применение векторных представлений слов и кластеризации.*

Рекомендательная система, векторные представления слов, ключевые слова, кластеризация, определение актуальности.

**Application of clustering and machine learning for building of a recommendation system for determining of scientific articles relevance**

**O.P. Mosalov**, Ph.D., head of the educational and scientific laboratory of intellectual control systems,

State educational institution of higher education Moscow region “Leonov’s University of technology”, Korolev, Moscow region,

**I.A. Ivanov**, bachelor in applied informatics, software developer,

Public Joint-Stock Company Sberbank of Russia, Moscow,

**M.A. Pershin**, bachelor in applied informatics, student of a group IO-17,

State educational institution of higher education Moscow region “Leonov’s University of technology”, Korolev, Moscow region.

*In the article a solution of a task of developing a recommendation system for determining of scientific article relevance is presented. An algorithm for calculation of relevance and applied methods of machine learning for text data processing are considered. Usage of word embeddings and clustering are justified.*

Recommendation system,word embeddings, key words, clustering, relevance determining

**Введение.** Обработка текстовой информации с применением методов машинного обучения является одной из крупнейших областей в сфере искусственного интеллекта. На основе современных, постоянно совершенствующихся инструментов для решения подобного рода задач создаются различные решения. Одним из немаловажных направлений разработки таких решений является создание рекомендательных систем [6].

Одной из самых молодых и активно развивающихся областей разработок является использование рекомендательных систем в сфере образования [8], например, такие сервисы, как MathGarden [20]. Данные сервисы направлены на персонализированное обучение пользователей.

Важно отметить, что в данной области существует большая потребность в анализе текстовой информации для определения её качества и актуальности.

В данной статье рассматриваются методы машинного обучения, а также технологии, применяемые для построения рекомендательной системы, позволяющей оценивать актуальность текстов научных публикаций.

Основная цель разработки данной рекомендательной системы заключается в том, чтобы получить решение, которое способно не только дать экспертную оценку актуальности текущему состоянию текста публикации на основе его статистических характеристик, но и сформировать рекомендации по увеличению этого показателя. Другая цель – за счёт внедрения данной системы дать пользователям возможность сократить время на анализ и редактирование их статей.

**Назначение разрабатываемой системы, основные термины и решения****.** Есть основания полагать, что на основе текста научной статьи можно определить ее актуальность, для этого необходимо проанализировать текст и выделить ключевые слова и фразы, векторизовать и изучить их расположение в векторном пространстве.

Главная задача разрабатываемой информационной системы заключается в том, чтобы, получив от пользователя публикацию, определить величину актуальности и дать рекомендации по изменению содержания статьи с целью повышения этого показателя.

Для работы основного алгоритма необходимо обработать и сформировать набор тем состоящий из ключевых слов, полученных из Yet Another Keyword Extractor [13]. Данная задача была решена с помощью кластеризации и векторных представлений библиотеки Navec [22].

Также необходимо пояснить, что для анализа расположения точек в векторном пространстве в нем заранее выделяются центры областей (классов), соответствующих общепринятым темам научных публикаций. Эти центры строятся с помощью рубрикации множества ключевых слов, вручную извлечённых из массива научных статей.

**Основной алгоритм работы системы.**

1. На вход поступает публикация с выделенными в ней ключевыми словами. Важно учитывать, что статьи должны включать разделы:
* Ключевые слова;
* Основной текст.
1. После загрузки статьи через интерфейс, происходит подготовка её текста к отправке в API Yet Another Keyword Extractor, развёрнутому в Docker Container-е. Данная процедура помогает нам определить предполагаемые ключевые слова, на основе текста статьи. На сервере данная обработка начинается с этапа фильтрации текста за счёт использования инструмента языка Java для программной обработки документов Microsoft, носящего название Apache POI [11]. Также, если на вход поступает публикация в формате файла PDF, будет использоваться такой инструмент, как IText [17], позволяющий получить текст публикации из файла формата PDF. После выгрузки текста из файла, он проходит несколько этапов обработки:
2. Замена всех переносов параграфов в тексте на «.» для лучшего распознавания текста Yet Another Keyword Extractor-ом;
3. Удаление англоязычных блоков текста;
4. Составление из найденных частей готового к анализу текста научной публикации, который будет содержать только необходимые нам блоки текста, загруженной научной публикации, что в свою очередь позволит нам, используя Yet Another Keyword Extractor, более точно определить ключевые слова анализируемой публикации.
5. На основании ключевых слов определяется принадлежность публикации к классам.
6. Сохраняется выборка значений актуальности по классам.
7. Формируются рекомендации на основании полученных значений актуальности.

**Цифровые представления слов.** Для получения цифровых представлений использовался упомянутый выше инструмент Navec. Данный инструмент применяется в отдельном модуле системы, направленным на проведения в нём всех математических расчётов. Для построения этого модуля применялись следующие инструменты:

1. Navec;
2. Keras [14];
3. NumPy [23];
4. PyPlot [16];
5. Sklearn [24];
6. Pymorphy2 [18].

**Navec**. Библиотека Navec – часть проекта Natasha [21], коллекция предобученных векторных представлений слов для русского языка. В процессе анализа было выяснено, что около 80-90% веса модели занимает таблица векторных представлений. Небольшой размер – это преимущественное отличие Navec от других библиотек. Работа с компактными моделями очень удобно, так как размер модели векторных представлений небольшой, соответственно потребление объема памяти значительно снижается, в то время как скорость работы повышается, чем выгодно отличает Navec.

Для наглядности векторные представления Navec сравнены с другими аналогами (RusVectores) [19].

|  |  |  |  |
| --- | --- | --- | --- |
|  | Качество | Размер модели, МБ | Размер словаря, x103 |
| Navec | 0,719 | 50,6 | 500 |
| RusVectores | 0,638-0,726 | 220,6-290,7 | 189-249 |

Таблица 1 – Сравнение характеристик векторных представлений.

Из таблицы 1 очевидно, что размер архива Navec в 5-6 раз меньше, чем у RusVectores, в то время как размер словаря больше в 2-3 раза (500 тысяч слов).

Для оценки качества векторных представлений слов была рассмотрена семантическая близость на шести датасетах (SimLex965, LRWC, HJ, RT, AE, AE2). Модель, использующая векторные представления Navec, показала наилучшие результаты сред рассмотренных вариантов.

Для разрабатываемого сервиса был выбран архив hudlit\_12B\_500K\_300d\_100q, обученный на 145 ГБ художественной литературе. Данный архив позволяет формировать векторные представления слов длинной 300.

**Разработанные модули системы.** Разрабатываемая система имеет следующую структуру компонентов (Рисунок 1):

1. Ядро системы – содержит основные алгоритмы и базу данных MySQL.
2. Yet Another Keyword Extractor – используется для анализа текста статьи и выделения из неё ключевых словосочетаний и слов.
3. Модуль NLP – содержит нейросетевой фильтр [2, 3, 15] для фильтрации не пригодных для дальнейшей обработки ключевых слов полученных из Yet Another Keyword Extractor, а также функционал для различных операций над цифровым представлением текстовой информации таких как:
	1. Вычисления расстояний между кластерами и ключевыми словами
	2. Формирование векторных представлений слов.



Рисунок 1 – Диаграмма развёртывания.

Подробное описание устройства и вычислительных экспериментов с модулем NLP приведены в работе [5].

**Алгоритмы кластеризации.** Для задачи рубрикации ключевых слов были выбраны и реализованы алгоритмы жесткой кластеризации k‑means и нечеткой fuzzy c‑means [1, 4, 9, 10]. При выполнении кластеризации были использованы методы оценки качества кластеризации: метод силуэта и инерции [7].

В качестве множества объектов для кластеризации были использованы векторные представления около тысячи ключевых слов, извлечённых из около трёхсот научных статей.

Для определения того, что называется величиной актуальности – вводится понятие «классы». «Классы» формируются на основе, полученных при кластеризации кластерах. Это набор отдельных тем, к которым с некоторой степенью принадлежности *A* – можно отнести ключевые слова извлечённые из Yet Another Keyword Extractor. Данный набор классов необходимо представлять отдельными точками в многомерном пространстве для обеспечения возможности вычисления степени принадлежности классов к ключевым словам.

**Алгоритм k-means.** Один из самых надёжных и наиболее часто используемых алгоритмов кластеризации – это k-means. Этот алгоритм часто применяется как базовый вариант для сравнения и оценки эффективности работы других, более сложных алгоритмов.

В начале случайным образом в пространстве кластеризуемых объектов выбирается *k* точек, которые являются начальным приближением к центрам будущих кластеров и называются центроидами. Далее для каждого объекта определяется ближайший к нему центроид и, таким образом, формируется первая версия кластеров. Затем для получившихся кластеров с использованием выбранной меры близости вычисляются новые центроиды. Далее эти два шага – распределение объектов по кластерам и обновление центроидов – повторяются до тех пор, пока изменение центроидов не станет пренебрежимо малым. Также возможны и другие критерии остановки алгоритма, например – заранее заданное количество итераций обновления центроидов и т.п. Визуально этот процесс показана на рисунке 2.



Рисунок 2 – Общий алгоритм работы k-means.

В алгоритме k-means можно использовать широкий спектр мер близости при вычислении ближайшего центроида. В данной работе была выбрана самая популярная мера близости – евклидово расстояние.

Для определения количества кластеров были применены методы оценки качества кластеризации. На рисунке 3 представлен график коэффициента силуэта для кластеризации данным методом



Рисунок 3 – Коэффициент силуэта для k-means в зависимости от количества кластеров.

Также были рассчитаны значения коэффициента инерции, график которого представлен на рисунке 4.



Рисунок 4 – Коэффициент инерции для k-means в зависимости от количества кластеров.

После анализа полученных графиков оценки кластеризации было определенно количество кластеров – 13, и были кластеризованы векторные представления ключевых слов. Результаты кластеризации методом k-means представлены на рисунке 5.



Рисунок 5 – Результаты кластеризации: количество ключевых слов, попавших в каждый кластер.

**Метод нечеткого fuzzy c-means.** Одной и главных проблем предыдущего алгоритма кластеризации (k-means), является его жесткость, что снижает эффективность рубрикации. K-means относит ключевую фразу лишь к одному кластеру, что не всегда интуитивно корректно, поэтому было принято решение использовать алгоритм нечеткой кластеризации fuzzy c-means.

Не всегда возможно однозначно распределить точки по кластерам, особенно в сложных наборах данных, где, в соответствии с предметной областью, кластеры перекрываются. Для эффективного выполнения кластеризации в таких случаях вместо отнесения точек к кластерам говорят о наборе вероятностей вхождения каждой точки в каждый кластер.

Алгоритм fuzzy c-means требует больше системных ресурсов, однако, особенности нечеткой кластеризации, дают более правильные и легче интерпретируемые результаты кластеризации.

Также, как и для алгоритма k-means, была проведена оценка количества кластеров методом силуэта. Результаты представлены на рисунке 6.



Рисунок 6 – Коэффициент силуэта для fuzzy c-means в зависимости от количества кластеров.

Видно, что коэффициент силуэта не применим как критерий выбора количества кластеров в случае нечёткой кластеризации, поэтому дополнительно были использованы коэффициенты разделения и энтропии разделения [12], результаты представлены на рисунках 7 и 8.



Рисунок 7 – Коэффициент разделения в зависимости от количества кластеров.



Рисунок 8 – Коэффициент энтропии разделения в зависимости от количества кластеров.

Результаты нечеткой кластеризации не имеет смысла интерпретировать в виде диаграммы, аналогичной рисунку 5, так как каждый объект относится ко всем кластерам с некоторой вероятностью.

После анализа полученных данных, в качестве количества кластеров для алгоритма fuzzy c-means было выбрано значение 20.

Таким образом, была построена рубрикация ключевых слов и, как следствие некоторая разметка пространства векторных представлений для дальнейшего использования при вычислении актуальностей.

**Алгоритм вычисления актуальности ключевого слова или фразы.** На вход алгоритма подаётся набор классов $C\_{i}$, где $i=1,…,N$, для каждого из которых путём запроса в поисковй API определена актуальность $A\_{i}$, а также собственно список ключевых слов $w\_{i}$, где $i=1,…,K$, для которых необходимо определить значения актуальности.

Отметим, что значения актуальности классов $A\_{i}$ никак не ограничены, однаков процессе вычисления актуальностей ключевых слов производится приведение этих значений к отрезку [0,1].

Для определения актуальности набора ключевых слов $w\_{i}$, где $i=1,…,K$, применяется следующий алгоритм:

1. Актуальности классов $A\_{i}$ нормируются, с использованием функции softmax:

|  |  |
| --- | --- |
| $$\hat{A}\_{i}=\frac{e^{A\_{i}}}{\sum\_{j}^{}e^{A\_{j}}}$$ | (1) |

1. Для каждого ключевого слова $w\_{i}$ вычисляются расстояния от него до каждого из классов:

|  |  |
| --- | --- |
| $$d\_{ij}=ρ\left(w\_{i},C\_{j}\right)$$ | (2) |

1. Нормируются расстояния для каждого ключевого слова $w\_{i}$:

|  |  |
| --- | --- |
| $$\hat{d}\_{ij}=\frac{e^{d\_{ij}}}{\sum\_{k}^{}e^{d\_{ik}}}$$ | (3) |

1. Актуальность ключевого слова $w\_{i}$ вычисляется следующим образом:

|  |  |
| --- | --- |
| $$a\_{i}=\frac{1}{N}\sum\_{j=1}^{N}\hat{A}\_{j}\left(1-\hat{d}\_{ij}\right)$$ | (4) |

Итого, значения нормированных актуальностей классов принадлежат отрезку [0,1], значения нормированных расстояний до классов для каждого ключевого слова – аналогично.

Так как суммы нормированных значений равны 1, их можно трактовать как вероятности.

Актуальности ключевых слов также принадлежат отрезку [0,1], они больше, если больше актуальности классов, расстояние до которых меньше.

При этом классы и ключевые слова являются объектами одного и того же пространства (300-мерных векторных представлений). На рисунке 9 показан пример взаимного расположения классов и ключевых слов.



Рисунок 9 – Визуализация степеней отношения отдельных слов к классам.

**Алгоритм вычисления актуальности статьи.** Для определения общей величины актуальности всей статьи используется ранее полученная информация об актуальности ключевых слов, извлечённых из текста статьи.

Вычисление общей актуальности статьи осуществляется по следующей формуле:

|  |  |
| --- | --- |
| $$a=\sum\_{i=1}^{\left‖\left\{w\right\}\right‖}a\_{i}$$ | (5) |

где $\left\{w\right\}$ – множество всех ключевых слов, а $a\_{i}$ – актуальность ключевого слова $w\_{i}$, определённая по формуле (4).

Таким образом, общая актуальность текста статьи считается суммой актуальностей отдельных ключевых слов.

**Эксперименты с трендами классов.** В ходе проведённых экспериментов отслеживания трендов актуальностей, установлено что в зависимости от текущего значения величины актуальности на текущий момент времени меняется величина актуальности текста статьи после анализа.

Данная зависимость видна на рисунках (рис. 10-12). Где на рисунке (рис. 10) приведён график изменения актуальностей, а на рисунках (рис. 11 и 12) – можно увидеть рост итогового процента актуальности у одной и той-же статьи.

Был рассмотрен временной промежуток с 03.05.2021 по 04.05.2021.

На графике видно, что в данный период наблюдался рост актуальности у класса «Планирование».



Рисунок 10 – График изменения трендов актуальности.



Рисунок 11 – Актуальность публикации 17.6%.



Рисунок 12 – Актуальность публикации 25,1%.

**Выводы.** Информационная система, описанная в рамках данной работы, относится к классу рекомендательных систем и является готовым инструментом для статистического анализа текста публикаций, определения актуальности текста и формирования рекомендаций для повышения актуальности. Данные рекомендации способны помочь пользователям на их основе принимать необходимые решения по редактированию содержания их публикаций.

При проектировании и разработки системы применён широкий спектр методов машинного обучения, а также более классические математические методы.

Полученные результаты будут применены как при доработке описанной системы, так и при реализации других систем, использующих аналогичные логические блоки, а также представляют академический интерес.

В дальнейшем планируется провести анализ трендов актуальностей классов для предсказания актуальностей на долгосрочный период и использование этих предсказаний для прогноза значения цитируемости статей. Также планируется применять получаемую от пользователей обратную связь для дальнейшей оптимизации отдельных блоков и системы в целом.

*Литература*

1. Воронцов К.В. Алгоритмы кластеризации и многомерного шкалирования. Курс лекций // МГУ, 2007.
2. Воронцов К.В. Машинное обучение (курс лекций), «Линейный классификатор и стохастический градиент» [Электронный ресурс] Режим доступа: www.machinelearning.ru/wiki/index.php?title=

Машинное\_обучение\_(курс\_лекций%2C\_К.В.Воронцов) (дата обращения: 03.11.2020).

1. Дударов С.П. История возникновения нейронных сетей [Электронный ресурс] Режим доступа: https://neuronus.com/history/5-istoriya-nejronnykh-setej.html (дата обращения: 23.03.2021).
2. Заде Л. А. Кластеризация и кластер // Л.А. Заде, С. Рао и др. Москва: Мир, 1980. 383 с.
3. Мосалов О.П., Иванов И.А. Использование автокодировщика для фильтрации в задаче извлечения ключевых слов из текста // XXIII Международная научно-техническая конференция «Нейроинформатика-2021» – М.: МИФИ – 2021 (в печати).
4. Рекомендательные системы: как помочь пользователю найти то, что ему нужно? [Электронный ресурс] Режим доступа: https://vc.ru/marketing/152926-rekomendatelnye-sistemy-kak-pomoch-polzovatelyu-nayti-to-chto-emu-nuzhno (дата обращения: 21.11.2020).
5. Руссеу П.Дж. Силуэты: графическое средство для интерпретации и проверки кластерного анализа // Вычислительная и прикладная математика. 20, 1987, С. 53-65.
6. Статистика российского образования [Электронный ресурс] Режим доступа: http://stat.edu.ru/stat/vis.shtml (дата обращения: 12.06.2020).
7. Часовских A. Обзор алгоритмов кластеризации данных [Электронный ресурс] Режим доступа: https://habr.com/ru/post/101338/ (дата обращения: 20.04.21).
8. Чубукова И.А. Курс лекций «Data Mining Методы кластерного анализа // Интернет-университет информационных технологий. [Электронный ресурс] Режим доступа: https://www.intuit.ru/studies/courses/6/6/lecture/182?page=2/ (дата обращения: 05.05.21).
9. Apache Software Foundation, Apache POI - the Java API for Microsoft Documents [Электронный ресурс] Режим доступа: https://poi.apache.org (дата обращения: 09.07.2020).
10. Askerova B.G. Optimum entropy clustering in information systems // International research and practice journal “Software & Systems”, № 4, 2017, pp. 643-646.
11. Campos R., Mangaravite V., Pasquali A., Jatowt A., Jorge A., Nunes C., Jatowt A. YAKE! Keyword Extraction from Single Documents using Multiple Local Features // In Information Sciences Journal. Elsevier, Vol. 509, 2020, pp. 257-289.
12. Deep learning for humans. Keras [Электронный ресурс] Режим доступа: https://keras.io/ (дата обращения: 14.01.2021).
13. GoProg.ru Нейронные сети: правило Хебба [Электронный ресурс] Режим доступа: https://goprog.ru/posts/nn-hebb (дата обращения: 09.10.2020).
14. Hunter J. D. Matplotlib: A 2D Graphics Environment // Computing in Science & Engineering, Vol. 9, No. 3, 2007, pp. 90-95.
15. IText, iText 7 core [Электронный ресурс] Режим доступа: https://itextpdf.com/ru/products/itext-7/itext-7-core (дата обращения: 21.07.2020).
16. Korobov M. Morphological Analyzer and Generator for Russian and

Ukrainian Languages // Analysis of Images, Social Networks and Texts, 2015, pp. 320-332.

1. Kutuzov A., Kuzmenko E. WebVectors: A Toolkit for Building Web Interfaces for Vector Semantic Models // In: Ignatov D. et al. (eds) Analysis of Images, Social Networks and Texts. AIST 2016. Communications in Computer and Information Science, Vol. 661, 2017.
2. MathGarden [Электронный ресурс] Режим доступа: https://play.google.com/store/apps/details?id=com.nummolt.kinder.garden.math\_serialized&hl=en\_US&gl=US (дата обращения: 20.12.2020).
3. Natasha, Tools for Russian NLP: segmentation, embeddings, morphology, lemmatization, syntax, NER, fact extraction [Электронный ресурс] Режим доступа: https://github.com/natasha/natasha (дата обращения: 27.11.2021).
4. Navec, Compact high quality word embeddings for Russian language [Электронный ресурс] Режим доступа: https://habr.com/ru/post/516098/#navec (дата обращения: 17.01.2021).
5. NumPy, The fundamental package for scientific computing with Python [Электронный ресурс] Режим доступа: https://numpy.org/ (дата обращения: 19.01.2021).
6. Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., Vanderplas J., Passos A., Cournapeau D., Brucher M., Perrot M., Duchesnay E. Scikit-learn: Machine Learning in Python // JMLR 12, 2011, pp. 2825-2830.