УДК 004.62

Реализация системы с веб-интерфейсом для просмотра статей конференции «Ломоносовские чтения на Алтае»

Н.С. Фелькер, О.Н. Половикова
АлтГУ, г. Барнаул

В наше время, когда количество информации увеличивается на десятки гигабайт за долю секунды, невозможно столь же быстро осуществлять поиск и анализ во всех этих неструктурированных сведениях. К тому же, если вручную осуществлять анализ этой информации, то на это уйдет достаточно большое количество времени. Поэтому эту задачу поручили специальным алгоритмам. Именно машинное обучение способно работать сразу с большим количеством данных в автоматическом режиме. Классификация документов – задача информационного поиска. Ее идея в том, что нужно составить классы (группы) документов так, чтобы в каждом классе находился близкий по смыслу документ, а в разных классах – различные. В информационном поиске зачастую прибегают к латентно-семантическому подходу, когда все слова из документа рассматриваются как принадлежащие к одной из нескольких тем. Одним из используемых методов, реализующих эту идею, является Латентное размещение Дирихле (LDA), который строит модель языка коллекции. Наше работа будет направлена на структурирование и доступ к информации статей конференции «Ломоносовские чтения на Алтае: фундаментальные проблемы науки и техники» Ежегодно в конференции принимает участие более 700 человек, по итогам конференции формируется сборник материалов. На данный момент не существует доступа к сборнику статей с различных платформ.

Целью данной работы является разработка системы с функцией публикации и хранения статей, с возможностью поиска по документам. А также, реализовать алгоритм подбора тематически близких по теме статей используя машинное обучение.

Задача. Проектирование и реализация веб-приложения, а также создание базы данных для хранения научных статей. Одна из задач – интеграция машинного обучения, а именно алгоритма Латентного размещения Дирихле, чтобы выявить темы документов, на примере статей конференции «Ломоносовские чтения на Алтае» и выдать прогноз близких по теме документов, так как это задача вероятностного тематиче-

ского моделирования, то один документ может относиться сразу к нескольким темам. Для этого необходимо реализовать алгоритм классификации текстов статей.

Для создания веб-приложения использовался фреймворк *Django*, именно в нем есть нативная поддержка выполнения Python приложений. Для работы с базой данных *Django* использует собственный ORM (объектно-реляционное отображение), в котором модель данных описывается классами Python, и по ней генерируется схема базы данных. Это избавляет от необходимости писать SQL-код. По умолчанию использовал *SQLlite3*. Алгоритм классификация (Латентное размещение Дирихле) реализован с помощью языка Python. Латентное размещение Дирихле (LDA) – это порождающая модель, объясняющая результаты наблюдений с помощью неявных групп, что позволяет получить объяснение, почему некоторые части данных схожи. Например, если наблюдениями являются слова, собранные в тексты, утверждается, что каждый текст представляет собой смесь небольшого количества тем и что появление каждого слова связано с одной из тем документа. LDA впервые был представлен в качестве графической модели для обнаружения тем Дэвидом Блеем, Эндрю Ыном и Майклом Джорданом в 2003 году [6]. Данный метод является доработкой PLSA, он основан на той же вероятностной модели, но с некоторыми дополнениями:

$$p(d,w) = \sum_{t \epsilon T} p(d)p(w|t)p(t|d)$$

- вектора $\theta_d = (p(t|d): t \epsilon T$ документов порождаются одним и тем же вероятностным распределением на нормированных $|T|$-мерных векторах; это распределение удобно взять из параметрического семейства распределений Дирихле $Dir(\theta, \alpha)$, $\alpha \epsilon R^{|T|}$;
- вектора тем $\varphi_t = (p(w|\ t) : w \epsilon W)$ порождаются одним и тем же вероятностным распределением на нормированных векторах размерности $|W|$; это распределение удобно взять из параметрического семейства распределений Дирихле $Dir(\theta, \beta)$, $\beta \epsilon R^{|W|}$ [7].

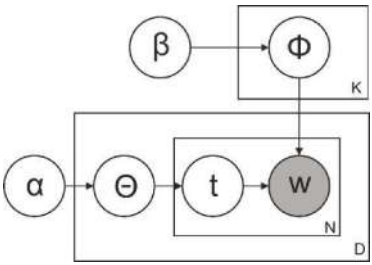

Рисунок 1 – Графическое представление LDA

Ниже представлен пример кода функции для сравнения документов:

```
unseen_document = 'дома'
bow_vector=dictionary.doc2bow(preprocess(unseen_document))
for index, score in sorted(lda_model_tfidf[bow_vector], key=lambda tup: -1*tup[1]):
    print("Score: {}\t Topic: {}".format(score, lda_model_tfidf.print_topic(index, 3)))
```

Таблица 1 – Результат работы алгоритма

Score: 0.662380039691925	Topic: 0.023*"дома" + 0.023*"умного" + 0.022*"безопасности"
Score: 0.1688511073589325	Topic: 0.019*"оценки" + 0.019*"кавитация" + 0.019*"исследования"
Score: 0.16876886785030365	Topic: 0.027*"ieee" + 0.027*"разработка" + 0.027*"стандарта"

Реализована основная архитектура веб-приложения, в котором есть функционал для кластеризации по тематическим разделам. Автоматизирован процесс добавления и статьи. Осуществляться хранение и доступ к статьям.

Библиографический список

1. Воронцов К.В. Лекции по вероятностными тематическим моделям. [Электронный ресурс] // MachineLearning.ru, информационно-аналитический ресурс. – Режим доступа: http://www.machinelearning.ru/wiki/images/2/22/Voron-2013-ptm.pdf, свободный.

2. Векторная модель. [Электронный ресурс] // Википедия, свободная энциклопедия. – Режим доступа: https://ru.wikipedia.org/ wiki/Векторная_модель, свободный.

3. TF-IDF. [Электронный ресурс] //Википедия, свободная энциклопедия. – Режим доступа: https://ru.wikipedia.org/wiki/TF-IDF, свободный.

4. Латентно-семантический анализ. [Электронный ресурс] / /Википедия, свободная энциклопедия. – Режим доступа: https://ru.wikipedia.org/wiki/Латентно-семантический_анализ/, свобдный.

5. Тематическое моделирование текстов на естественном языке. [Электронный ресурс] // КиберЛенинка, научная электронная библиотека. – Режим доступа: https://cyberleninka.ru/article/v/tematicheskoe-modelirovanie-tekstov-na-estestvennom-yazyke/, свободный.

6. Латентное размещение Дирихле. [Электронный ресурс] // Википедия, свободная энциклопедия. – Режим доступа: https://ru.wikipedia.org/wiki/Латентное_размещение_Дирихле/, свободный.

7. Тематическое моделирование. [Электронный ресурс] // Википедия, свободная энциклопедия. – Режим доступа: https://ru.wikipedia.org/wiki/Тематическое_моделирование/, свободный.

8. Оценка классификатора. [Электронный ресурс] // Личный блог. – Режим доступа: http://bazhenov.me/blog/2012/07/21/classification-performance-evaluation.html, свободный.

УДК 004.4

Использование алгоритма k-means в обработке изображений

А.Р. Сыздыкпаева[1], С.А. Шаймерден[1,2]
[1]ВКГУ им. С. Аманжолова, г.Усть-Каменогорск;
[2]АлтГУ, г. Барнаул

Спутниковые снимки, снятые в различных спектральных диапазонах, содержат очень полезную информацию и хранятся в цифровом виде. Использование космических снимков в оперативном обновлении карт среднего и мелкомасштабного экономически выгодно. На основе трех спектрального канала дистанционного зондирования цветные изображения переносят больше информации, чем наземные или аэрофотоснимки, а стереопары изображений позволяют проводить трехмерный анализ пространственных объектов [1,2].

k-means-самый распространенный вид из методов кластеризации. В связи с тем, что алгоритм имеет простоту и высокую скорость выполнения, он имеет большое значение [3]. Алгоритм k-means выполняется итеративно, который разбивает заданный набор пикселов на точки кластера k, приближающиеся к их центрам, и в результате перемещения места этих центров, выполняется кластеризация.

Алгоритм классирования k-means, последовательный в языке C++. Алгоритм SK means (последовательный k-средний). Обнаружен в алгоритме SK means в виде функции $J = \sum_{n=1}^{N}\sum_{k=1}^{k}\left\|x_n - c_k\right\|^2$, чтобы минимизировать функцию SK means назначения, то есть сделать погрешность функции квадратом. $J = \sum_{n=1}^{N}\sum_{k=1}^{k}\left\|x_n - c_k\right\|^2$, N – расстояние между данными, соответствующими центру J кластера, $x_n\ (1 \le n \le N)$ обозначает заданные точки и $c_k\ (1 \le k \le K)$ определяет тяжесть кластера $\left\|x_n - c_k\right\|^2$ – x_n и c_k – определение расстояния между ними. В