ФИЗИКА ЭЛЕМЕНТАРНЫХ ЧАСТИЦ И АТОМНОГО ЯДРА
2024. Т. 55, вып. 3. С. 660–664

АЛГОРИТМ ПОИСКА НАУЧНЫХ ПУБЛИКАЦИЙ НА ОСНОВЕ ЦИТИРОВАНИЙ С ПРИМЕНЕНИЕМ НЕЙРОСЕТЕВЫХ МОДЕЛЕЙ

Д. О. Доровских [1, 2, *], *А. Б. Теслюк* [1, **], *С. А. Бобков* [2, ***]

[1] Московский физико-технический институт
(национальный исследовательский университет), Долгопрудный, Россия
[2] Национальный исследовательский центр «Курчатовский институт», Москва

Базы данных научных публикаций насчитывают миллионы статей. Для доступа к релевантной информации необходимы эффективные методы поиска, в которых используются факторы, связанные с ключевой сутью публикаций. Представлен подход к извлечению сути статей из кратких описаний, используемых при цитировании, с применением нейросетевых моделей. В результате создан прототип сервиса, который позволяет находить научные публикации по коротким описаниям.

Databases of scientific publications contain millions of articles. To access relevant information, effective search methods are needed that use factors related to the key essence of publications. An approach to extracting the essence of articles from brief descriptions used in references with the help of neural network models is presented. As a result, a prototype of a search service was created that allows you to find scientific publications by short descriptions.

PACS: 89.20.Ff

ВВЕДЕНИЕ

Методы поиска по базам научных публикаций непрерывно развиваются. Помимо традиционного текстового поиска и систем, в которых учитываются индексы цитирований, применяется семантический поиск, нейросетевые модели и др. В популярных системах поиска по научной литературе Google Scholar и Scopus алгоритмы ранжирования не только выполняют полнотекстовый поиск, но и учитывают данные о цитированиях одних статей другими [1–3].

Системы CoCites [4] и Connected Papers [5] предоставляют возможности анализа частоты совместного цитирования и представления результатов в виде графа близких по смыслу статей. Повышение точности

* E-mail: dariana0803@mail.ru
** E-mail: anthony.teslyuk@gmail.com
*** E-mail: s.bobkov@grid.kiae.ru

поиска требует использования дополнительных факторов, связанных с сутью публикаций. В рамках исследования новых методов поиска по научной литературе была разработана система семантического поиска научных публикаций на основе информации о внешнем цитировании с использованием нейросетевых моделей по большим базам научных публикаций.

В качестве источника данных был выбран полнотекстовый архив научных публикаций по биомедицине PubMed Central (PMC) объемом 7,6 млн статей (9,1 ТБ) [6]. Для извлечения сути публикаций использовалась информация о цитировании одних статей другими, а именно текст авторского упоминания ключевых результатов другой работы и ссылка на нее. Для поиска по смыслу была применена современная нейросетевая модель BERT [7] на основе алгоритмов трансформеров [8].

В базе статей отбирались краткие описания со ссылками на другие статьи, в итоге было собрано более 550 000 упоминаний работ. С помощью дополнительно обученной модели BERT построено дерево векторов упоминаний статей. В результате был создан прототип сервиса поиска. Он принимает запросы пользователя и выполняет поиск по ключевым словам в созданном дереве. Для отображения результатов реализован веб-интерфейс сервиса поиска на основе библиотек Flask Python [9] и React [10].

ОБУЧАЮЩИЕ ДАННЫЕ

Метаинформация статей в архиве PubMed Central представлена в файлах формата XML, с помощью библиотеки Python lxml была извлечена такая метаинформация, как идентификаторы (pmid, doi), название публикации, авторы и текст аннотации. Она была объединена в набор данных в формате JSON, где ключами стали идентификаторы статей.

Отдельно был подготовлен набор данных с информацией о цитированиях. С помощью библиотеки Python Nltk размечены тексты статей и выделены предложения, содержащие упоминания ключевых результатов других работ и ссылки на них. Информация об упоминаниях статей помещена в файл формата JSONL. Файлы такого формата можно обрабатывать построчно, что крайне полезно для больших данных. Всего было выделено 530 530 упоминаний, каждая строка файла содержит идентификаторы источника и субъекта цитаты, текст упоминания.

РЕАЛИЗАЦИЯ

Данные из набора JSONL загружались с помощью библиотеки Pandas, которая создавала объект DataFrame, связывающий идентификаторы статей с данными о цитированиях. В дальнейшем загруженные данные используются для обучения и формирования результатов поиска.

Далее выполнялась предобработка набора упоминаний для подачи на вход нейросетевой модели. В работе использовалась нейросетевая модель BERT версии bert-base-uncased (12 слоев, 768 скрытых, 12 голов, 110 млн параметров), которая была предварительно обучена на корпусах данных на английском языке. Упоминания из набора подавались на вход токенизатора модели BERT, который разбивает текст на токены: слова, числа, знаки препинания и др. Затем токены конвертируются в соответствующие им числовые индексы.

Обработанные данные подаются на вход нейросетевой модели BertForSequenceClassification. Это модель BERT с добавленным линейным слоем сверху и возможностью решения задач многоклассовой классификации. В качестве выходных данных использовались векторы состояния последнего слоя модели — векторные представления для текстового упоминания. Пространство векторных представлений было поделено на части для эффективного поиска ближайших соседей, использовался алгоритм K-D-деревьев в библиотеке Scikit-learn.

Для увеличения точности поиска было реализовано дополнительное обучение нейросетевой модели BertForSequenceClassification. При дообучении модели стояла задача сделать векторные представления разных упоминаний в тексте одной и той же статьи ближе друг к другу.

Для этой цели было отобрано 10 000 текстовых упоминаний. В качестве меток для дообучения использовались идентификаторы статей, которые являются источниками упоминаний.

Набор данных с текстовыми упоминаниями статей был разделен на тренировочные и тестовые данные. При обучении входной набор тренировочных данных разбивается на пакеты установленного размера. Данные каждого пакета копируются на GPU, разбираются на числовые представления, выполняется прямой проход обучения. На выходе получаем потери, которые накапливаются по пакетам для вычисления среднего значения. Затем выполняется обратный проход для вычисления градиентов, обновляются параметры, вычисляются средние потери для пакетов данных.

Дообучение проводилось на кластере НИЦ «Курчатовский институт» HPC5 со следующими характеристиками: операционная система CentOS 7, 19 узлов с 2 CPU Intel Xeon E5-2650 v2 и 3 GPU Nvidia Tesla K80.

После обучения модель и соответствующий ей токенизатор сохраняются для использования в системе поиска.

СХЕМА ТЕСТИРОВАНИЯ

Прототип системы поиска по научным публикациям состоит из двух компонентов: поискового сервиса, программный код которого написан с применением библиотеки Flask Python, и веб-интерфейса на основе библиотеки React.

Поисковый сервис принимает на вход запросы от веб-интерфейса, содержащие список ключевых слов. Далее запросы токенизируются и передаются в нейросетевую модель. Для векторного представления запроса находятся ближайшие соседи в K-D-дереве векторных представлений, далее найденные упоминания соотносятся по идентификаторам с соответствующими статьями. Метаданные релевантных научных работ отправляются в веб-интерфейс в формате JSON.

Для разработанной системы поиска была проведена оценка качества результатов. Для выбранной статьи, библиографический список которой насчитывает N позиций, формировался текст запроса для поиска. Выполнялся поиск, и определялось количество публикаций K, которые найдены при поиске и действительно упоминаются в выбранной статье. Отношение K/N является метрикой точности поиска.

РЕЗУЛЬТАТЫ

Было проведено два вида тестирований: в первом случае запрос для поиска составлялся вручную, во втором — запросом являлось случайным образом выбранное упоминание из текста статьи. Разработанная система поиска сравнивалась классическим алгоритмом поиска на основе метрики Tf-Idf [11], где вместо нейросетевой модели BERT используется фиксированный алгоритм векторизации текста.

Вручную было подготовлено 30 запросов для поиска. Средняя точность поиска разработанного алгоритма с применением модели BERT составила 61 %. Алгоритм на основе метрики Tf-Idf показал точность в 41 %. Во втором случае было выполнено 100 автоматически выбранных запросов. В результате можно оценить среднюю точность поиска 58 % для алгоритма с применением BERT и 39 % для поиска на основе метрики Tf-Idf.

ЗАКЛЮЧЕНИЕ

В ходе работы получен алгоритм поиска по научным публикациям на основе набора данных текстовых упоминаний статей и нейросетевой модели BERT. Модель BERT была дополнительно обучена для улучшения качества поиска на наборе из 10 000 упоминаний.

Для интерактивного поиска в корпусе данных научных статей разработан поисковый веб-сервис на основе библиотек Python Flask и JavaScript React.

Тестирование поисковой системы для различных запросов показывает, что предложенный алгоритм на основе нейросетевой модели BERT имеет лучшую точность поиска по сравнению с системой на основе метрики Tf-Idf.

СПИСОК ЛИТЕРАТУРЫ

1. *Suzuki H.* Google Scholar Metrics for Publications. Google Scholar Blog. 2012.
2. *Burnham J. F.* Scopus Database: A Review // Biomed. Digit. Libr. 2006. V. 3, No. 1. P. 1–8.
3. *Falagas M. E., Pitsouni E. I., Malietzis G. A., Pappas G.* Comparison of PubMed, Scopus, Web of Science, and Google Scholar: Strengths and Weaknesses // FASEB J. 2008. V. 22, No. 2. P. 338–342.
4. *Small H.* Co-Citation in the Scientific Literature: A New Measure of the Relationship between Two Documents // J. Amer. Soc. Inform. Sci. 1973. V. 24, No. 4. P. 265–269.
5. *Eitan A., Smolyansky E., Harpaz I., Perets S.* Connected Papers: Find and Explore Academic Papers. 2020.
6. *Roberts R. J.* PubMed Central: The GenBank of the Published Literature. 2001.
7. *Devlin J., Chang M. W., Lee K., Toutanova K.* Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805. 2018.
8. *Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser Ł, Polosukhin I.* Attention Is All You Need // Adv. Neur. Inform. Proces. Syst. 2017. V. 30. P. 5998–6008.
9. *Grinberg M.* Flask Web Development: Developing Web Applications with Python. O'Reilly Media, Inc., 2018.
10. *Fedosejev A.* React. js Essentials. Packt Publ. Ltd, 2015.
11. *Bafna P., Pramod D., Vaidya A.* Document Clustering: TF-IDF Approach // 2016 Intern. Conf. on Electrical, Electronics, and Optimization Techniques (ICEEOT). IEEE. 2016. P. 61–66.