Петрова А. Н., Фролов Д. О.
A. N. Petrova, D. O. Frolov

ОПТИМИЗАЦИЯ ПОИСКА В НАУЧНЫХ ПУБЛИКАЦИЯХ С ПОМОЩЬЮ ГЛУБОКОГО ОБУЧЕНИЯ

OPTIMIZING SEARCH IN SCIENTIFIC PUBLICATIONS USING DEEP LEARNING

Петрова Анна Николаевна – кандидат технических наук, заведующая кафедрой «Проектирование, управление и развитие информационных систем» Комсомольского-на-Амуре государственного университета (Россия, Комсомольск-на-Амуре). E-mail: PetrovaAN2006@yandex.ru.
Anna N. Petrova – PhD in Technical Sciences, Head of the Department «Design, Management and Development of Information Systems», Komsomolsk-na-Amure State University (Russia, Komsomolsk-on-Amur). E-mail: PetrovaAN2006@yandex.ru.

Фролов Дмитрий Олегович – аспирант Комсомольского-на-Амуре государственного университета (Россия, Комсомольск-на-Амуре). E-mail: optcompanys@mail.ru.
Dmitriy O. Frolov – Postgraduate Student, Komsomolsk-na-Amure State University (Russia, Komsomolsk-on-Amur). E-mail: optcompanys@mail.ru.

Аннотация. В статье представлен новый подход к автоматическому поиску в научных данных, основанный на использовании модели рекуррентной свёрточной нейронной сети. С учётом увеличения объёма научных публикаций и сложности выбора алгоритмов данный метод направлен на улучшение автоматического обнаружения и анализа алгоритмов, предложенных в текстах научных статей. В рамках работы описаны два ключевых модуля: первый из них направлен на обнаружение псевдокода с применением Multi-Layer Perceptron, а второй – на извлечение целевых текстовых строк, содержащих информацию об эффективности алгоритмов, с использованием Region-based Convolutional Neural Network. Оба модуля реализованы в фреймворке PyTorch, а для обработки данных используется метод встраивания слов. Проанализированы методы автоматического извлечения элементов документа, таких как псевдокод, таблицы, рисунки и графики, а также предложен способ создания краткого текста для улучшения понимания их содержания. Описана система поиска алгоритмов, которая включает создание синопсиса, индексирование документов и использование современного метода поиска для ранжирования результатов. Экспериментальная часть демонстрирует, что улучшенная модель машинного обучения существенно превосходит базовую модель по всем ключевым метрикам: точность возросла с 75,95 до 98,8 %, запоминаемость увеличилась с 71,02 до 97,9 %. Результаты подчёркивают значительный вклад предложенного метода в автоматизацию поиска и анализа научных данных, что может значительно облегчить процесс выбора и оценки алгоритмов в научных исследованиях.

Summary. The paper presents a new approach to automatic search in scientific data based on the use of a recurrent convolutional neural network model. Taking into account the increase in the volume of scientific publications and the complexity of choosing algorithms, this method is aimed at improving the automatic detection and analysis of algorithms proposed in the texts of scientific articles. The work describes 2 key modules: the 1st of them is aimed at detecting pseudocode using Multi-Layer Perceptron, and the 2nd is aimed at extracting target text strings containing information about the efficiency of algorithms using Region-based Convolutional Neural Network. Both modules are implemented in the PyTorch framework, and the word embedding method is used for data processing. The methods for automatic extraction of document elements such as pseudocode, tables, figures, and graphs are analyzed, and a method for creating a short text for improving the understanding of their content is proposed. The algorithm search system is described, which includes synopsis creation, document indexing, and the use of a modern search method for ranking results. The experimental part demonstrates that the improved machine learning model significantly outperforms the baseline model in all key metrics: accuracy increased from 75.95% to 98.8%, memorability increased from 71.02% to 97.9%. The results highlight the significant contribution of the proposed method to the automation of scientific data search and analysis, which can significantly facilitate the process of selecting and evaluating algorithms in scientific research.

Ключевые слова: автоматический поиск, научные данные, рекуррентная свёрточная нейронная сеть, машинное обучение, обнаружение псевдокода, семантические метаданные, встраивание слов.

Key words: automatic search, scientific data, recurrent convolutional neural network, machine learning, pseudocode detection, semantic metadata, word embedding.

УДК 517.95

Введение. В последние десятилетия учёные предлагали алгоритмы для решения вычислительных задач, которые затем применялись в других областях. Например, алгоритмы для выравнивания биопоследовательностей и обнаружения частых закономерностей изначально создавались для других задач. В связи с ростом объёма научных данных выбор подходящего алгоритма становится сложной задачей.

Научные данные содержат сотни тысяч статей, многие из которых предлагают новые алгоритмы. Существуют методы поиска алгоритмов на основе метаданных, но они не учитывают специфические функции, такие как временная сложность. Поиск подходящего алгоритма часто основывается на опыте человека, а не на систематическом анализе данных.

Автоматический поиск алгоритмов в научных данных – непростая задача, требующая анализа как текста, так и элементов документа, таких как таблицы и графики. Традиционные методы не справляются с задачей извлечения информации о результатах и эффективности алгоритмов. В статье представлен улучшенный подход на основе машинного обучения для обнаружения алгоритмов, который достигает 96,5 % точности.

Обнаружение элементов документа. Извлечение элементов документа, таких как псевдокод, таблицы, рисунки и графики, из цифровых статей способствует обобщению результатов, описанию инструкций и визуализации идей. Автоматическое извлечение и индексирование этих элементов открывают новые возможности для применения в интеллектуальном анализе данных. Методы оптического распознавания символов и компьютерного зрения, такие как системы для автоматического извлечения научных изображений, позволяют извлекать результаты из графических объектов и эффективно индексировать их для поиска.

Научные библиотеки, такие как «Многопрофильный научный журнал», публикуемый Public Library of Science, и «Цифровая библиотека и поисковая система для научных публикаций», поддерживают поиск по таблицам и рисункам, но не предоставляют текстовое обобщение содержимого этих элементов. Для решения этой задачи был предложен метод автоматического создания краткого текста, описывающего элемент документа, что помогает пользователям лучше понять его релевантность их информационным запросам.

Система поиска алгоритмов использует машинное обучение для автоматического извлечения и индексирования псевдокода в текстовых документах, создавая метаданные для поиска по запросам пользователей. Хотя точность обнаружения псевдокода в такой системе достаточна для практического использования, она может быть улучшена. Для этого была предложена улучшенная техника машинного обучения на основе нейронных сетей Multi-Layer Perceptron.

Нейронные сети для анализа текста. Развитие глубоких нейронных сетей и методов обучения представлению способствовало решению проблемы разреженности данных и улучшению понимания словарных представлений. Встраивание слов позволяет оценивать семантическую связанность, и предварительно обученные встраивания вместе с глубокими нейронными сетями доказали свою высокую эффективность в задачах обработки естественного языка и текстовой классификации. Для определения семантической связанности в научных текстах модели на основе Region-based Convolutional Neural Network (RCNN) особенно важны для текстовой классификации и обнаружения перефразирования. RCNN используется для выделения строк текста, где обсуждаются алгоритмы с точки зрения их эффективности (точности и полноты).

Извлечение семантических данных. На рис. 1 описывается подход к извлечению семантических метаданных. Он состоит из двух подмодулей. Первый модуль улучшает существующий

алгоритм обнаружения псевдокода с использованием нейронной сети, обученной на 15 функциях документа, и обеспечивает более высокую точность. Второй модуль – обнаружение оценочных метрик (определение метрик оценки) – извлекает строки текста, содержащие обсуждение алгоритмов и экспериментов, основанных на оценочных метриках. Второй модуль использует сегментацию документов для извлечения соответствующих разделов и определяет целевые строки с помощью Region-based Convolutional Neural Network, обученного на вручную размеченном наборе данных.

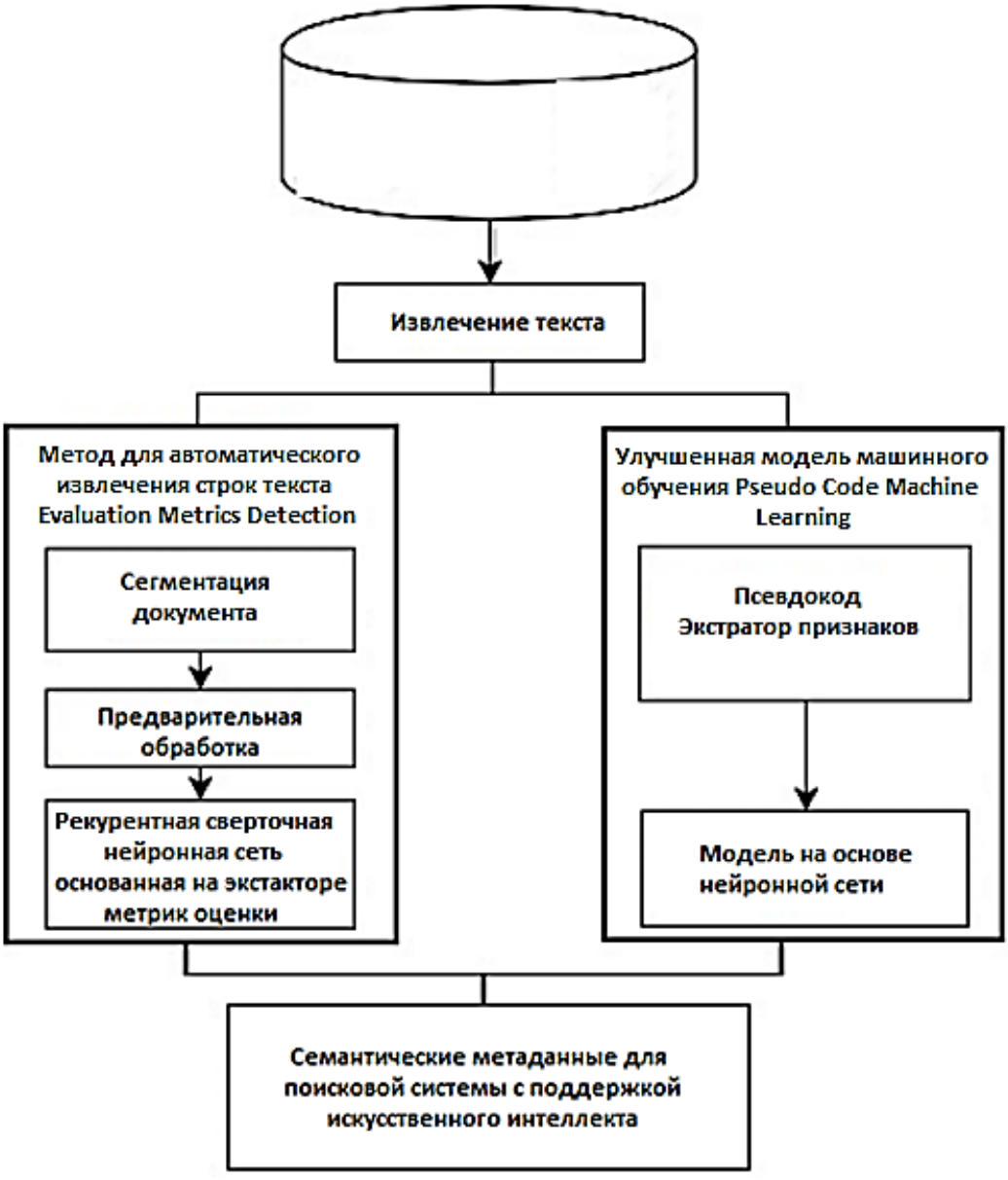

Рис. 1. Подход к извлечению семантических метаданных

Модель нейронной сети на основе глубокого обучения. На рис. 2 представлена структура Region-based Convolutional Neural Network. Эта модель использует слова и контекст в качестве представлений слов. Двунаправленная Region-based Convolutional Neural Network применяется для захвата контекста слов.

Для левого и правого контекста слова w_i вычисляются векторы $c_r(w_i)$ и $c_l(w_i)$ с использованием специального уравнения

$$c_l(w_i) = f(\left(w^{(l)}\right)c_l(w_{i-1}) + \left(w^{(sl)}\right)e(w_{i-1})),$$

где $w^{(l)}$ служит для преобразования контекста между скрытыми слоями, а $w^{(sl)}$ объединяет контекст левого слова с текущим словом. Вектор $e(w_{i-1})$ представляет собой встраивание предыдущего слова w_{i-1} с элементами вещественного значения, в то время как $c_l(w_{i-1})$ обозначает левый контекст этого слова.

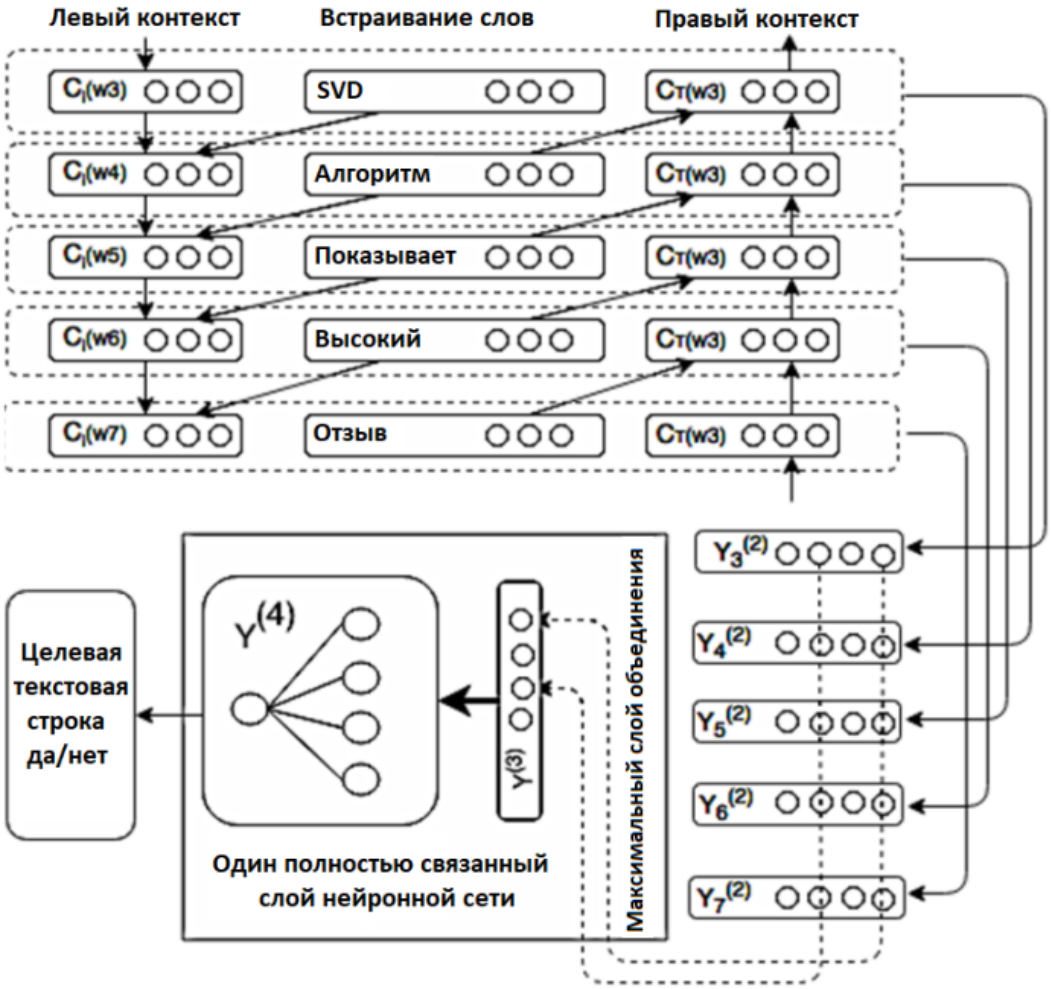

Рис. 2. Структура сети Region-based Convolutional Neural Network, включающая индексы, которые обозначают положение каждого слова в предложении

Аналогичным образом вычисляется $c_r(w_i)$:

$$c_r(w_i) = f(\left(w^{(r)}\right)c_r(w_{i-1}) + \left(w^{(sr)}\right)e(w_{i-1})).$$

На рис. 2 вектор $c_l(w_7)$ включает контекст и семантику всех левых слов в форме кодировок. Левый контекст сети – алгоритм разложения сингулярных значений (Singular Value Decomposition) – демонстрирует высокий уровень взаимодействия со всеми предыдущими словами предложения. Аналогично $c_r(w_7)$ содержит правый контекст.

Представление слова w_i формируется путём объединения левого и правого контекстов:

$$x_i = [c_l(w_i); e(w_i); c_r(w_i)].$$

Модель получает все значения c_l и c_r при прямом и обратном проходах соответственно. После изучения представления слова в x_i было применено линейное преобразование с функцией активации *tanh*, чтобы добавить некоторую нелинейность:

$$y_i^{(2)} = \tanh(W^{(2)}x_i + b^{(2)}),$$

где $y_i^{(2)}$ представляет скрытый семантический вектор, который включает в себя наиболее значимые и мощные факторы для представления текста, анализируя каждый семантический компонент.

Модель также включает слой максимального пула, как показано в сетевой архитектуре на рис. 2. Слой объединения преобразует текст разной длины в вектор фиксированной длины, что помогает выделить наиболее важную информацию во всём тексте.

Таким образом, после изучения представления всех слов был использован слой максимального пула:

$$y^{(3)} = \max_{i=2}^{n} y_{(i)}^{(2)},$$

где функция max применяется поэлементно и максимум k-го элемента $y_{(i)}^{(2)}$ находится в k-й позиции $y^{(3)}$.

Модель включает один полностью связанный скрытый слой, который служит выходным слоем:

$$y^{(4)} = W^{(4)} y^{(3)} + b^{(4)}.$$

Наконец, была применена сигмовидная функция активации, чтобы получить число вероятности:

$$p_{(i)} = \frac{\exp(y_k^4)}{1 + exp^{(y_k^4)}}.$$

Обучение модели RCNN. Для обучения модели RCNN первостепенно были определены все параметры модели для обучения θ:

$$\theta = \{E, b^{(2)}, b^{(4)}, c_l(w_1), c_r(w_n), W^{(2)}, W^{(4)}, W^{(l)}, W^{(r)}, W^{(sl)}, W^{(sr)}\},$$

где векторы $b^{(2)}$, $b^{(4)}$ являются значимыми векторами, а E – вещественные вложения слов, тогда как $W^{(2)}$, $W^{(4)}$, $W^{(l)}$, $W^{(r)}$ и $W^{(sl)}$, $W^{(sr)}$ представляют собой преобразования матрицы и $c_l(w_1), c_r(w_n)$ – начальные вектора вещественного значения левого и правого контекста. Целью было максимизировать вероятность (логарифм правдоподобия) относительно θ.

Далее для оптимизации процесса обучения использовался стохастический градиентный спуск:

$$\theta \rightarrow \sum_{c \in D} \log p(class_c | D, \theta),$$

где D представляет собой набор документов, а $class_c$ обозначает положительный класс текстовых данных.

В качестве представления слов было применено встраивание на основе модели для обработки естественного языка (NLP). Поскольку набор данных страдал от проблемы дисбаланса классов, были использованы следующие методы для балансировки модели:

– случайная избыточная выборка, при которой примеры класса меньшинства случайным образом дублируются до достижения равенства классов;

– случайная недостаточная выборка, при которой примеры большинства класса случайным образом удаляются до тех пор, пока оба класса не станут равными.

Для обучения использовались 68 % данных, а для тестирования – 32 %, как и в модели Pseudo Code Machine Learning. После применения методов балансировки у нас оказалось 4337 положительных образцов (целевые строки) и 4770 отрицательных образцов (строки, не содержащие информации об эффективности алгоритма). Наконец, были скорректированы гиперпараметры сети, такие как размер скрытого слоя H (до 100), скорость обучения (до 0,001), размер словаря V (до 3000) и количество эпох обучения (до 100).

Поисковая система с автоматическим поиском в научных данных. Разработанная поисковая система также включает в себя поддержку искусственного интеллекта. В поисковой системе выполняются важные ключевые этапы:

– создание синопсиса для каждого документа с применением улучшенной модели машинного обучения и методики определения оценочных метрик;

– стандартное индексирование как синопсиса, так и полнотекстовых документов;

– использование современного метода поиска для ранжирования результатов на основе пользовательских запросов и проведения сравнительного анализа.

Эксперимент. Эксперименты состояли из двух ключевых модулей:

1. обнаружение псевдокода с использованием нейронной сети Multi-Layer Perceptron;

2. обнаружение целевых текстовых строк (содержащих информацию об эффективности алгоритма) с использованием Region-based Convolutional Neural Network. Для реализации Region-based Convolutional Neural Network был использован фреймворк PyTorch.

Набор данных включал 327 научных статей. Для обоих модулей использовался одинаковый набор данных, что позволило провести сравнение между базовой моделью и улучшенной моделью машинного обучения. В нашем наборе данных, содержащем около 41 240 текстовых строк, только 7,1 % представляет собой целевой текст, передающий информацию об эффективности соответствующего алгоритма.

Сравнение моделей машинного обучения. Индексы стандартной точности, полноты, *f*-меры и точности используются для оценки как обнаружения псевдокода, так и обнаружения целевых текстовых строк.

В табл. 1 представлено сравнение базовой версии модели машинного обучения и улучшенной модели для обнаружения псевдокода. Результаты показали, что улучшенная модель, использующая Multi-Layer Perceptron, превзошла базовую модель по всем показателям. С улучшенной моделью была достигнута точность 98,8 %, в то время как базовая модель показала результат 75,95 %, что на 27 % хуже. Также наблюдается значительное улучшение в других метриках: запоминаемость возросла с 71,02 до 97,9 %, точность – с 76,52 до 98,2 %, а общий показатель *f*1 – с 76,25 до 97,1 %.

Таблица 1

Показатели сравнения моделей машинного обучения

Метод	Запоминаемость	Точность	Общий показатель
Базовая модель машинного обучения	71,02	76,52	76,25
Улучшенная модель машинного обучения	97,9	98,2	97,1

Заключение. В данной статье представлен новый подход к автоматическому поиску алгоритмов в научных данных, использующий модель рекуррентной свёрточной нейронной сети. Предложенный метод эффективно решает проблему выбора подходящих алгоритмов из множества научных статей и учитывает как текстовые, так и визуальные элементы документа.

Разработанные модули для обнаружения псевдокода и целевых текстовых строк показали значительное улучшение по сравнению с базовыми моделями. Использование Multi-Layer Perceptron для обнаружения псевдокода и Region-based Convolutional Neural Network для извлечения информации об эффективности алгоритмов продемонстрировало высокую точность и эффективность. Применение встраивания слов и методов балансировки классов обеспечило надёжность и стабильность модели.

Результаты экспериментов подтверждают, что улучшенная модель машинного обучения значительно превосходит базовые подходы: увеличение точности на 27 % и существенное улучшение других метрик. Эти достижения подчёркивают потенциал предложенной системы в улучшении автоматизации поиска и анализа алгоритмов, что может значительно облегчить работу исследователей, занимающихся обработкой больших объёмов научной информации.

Данный подход открывает новые возможности для интеграции интеллектуального анализа данных в поисковые системы, способствуя более точным и эффективным извлечению и оценке информации из научных публикаций. В будущем предполагается дальнейшее развитие методов и

технологий, что может привести к улучшению качества автоматического поиска и анализа данных в различных областях науки и техники.

ЛИТЕРАТУРА

1. Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. Journal of Machine Learning Research, 3, 993-1022.
2. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of NAACL-HLT 2019.
3. Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.
4. Joulin, A., Mikolov, T., Grave, E., Bojanowski, P., Mikolov, T., & Tsvetkov, A. (2017). Bag of Tricks for Efficient Text Classification. Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics.
5. Chen, J., & Zhou, Y. (2016). A Survey on Deep Learning for Data Mining. IEEE Transactions on Knowledge and Data Engineering.
6. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation.
7. Liu, X., & Zhang, J. (2020). Deep Learning for Text Classification: A Comprehensive Review. Journal of Computer Science and Technology, 35 (5), 923-950.
8. Kumar, V., & Ailawadhi, A. (2020). Deep Learning-Based Text Representation for Document Retrieval. Journal of Computational Information Systems, 16 (7), 129-141.
9. Liu, Y., & Zhang, X. (2019). A Survey of Text Mining Techniques and Applications. Information Processing & Management, 56 (5), 1035-1060.
10. Hochreiter, S., & Schmidhuber, J. (2015). Learning to Learn Using Gradient Descent. Proceedings of the 28th International Conference on Machine Learning (ICML).
11. Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. Proceedings of the 27th International Conference on Neural Information Processing Systems (NeurIPS).
12. Ghani, R., & Le, M. T. (2020). Improving Search Performance with Deep Learning Models. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval.
13. Yin, P., & Yao, X. (2021). Transfer Learning for Natural Language Processing: A Survey. ACM Computing Surveys (CSUR), 54 (2), 1-35.
14. Xia, Y., & Li, X. (2018). Deep Neural Networks for Document Classification and Search. IEEE Transactions on Knowledge and Data Engineering, 30 (10), 1892-1905.
15. Zhang, C., & Li, W. (2020). A Comprehensive Review of Deep Learning-Based Text Mining Techniques. ACM Transactions on Intelligent Systems and Technology (TIST), 11 (4), 1-30.