International Journal of Scientific Research in Science and Technology
Available online at : www.ijsrst.com
Print ISSN: 2395-6011 | Online ISSN: 2395-602X
doi : https://doi.org/10.32628/IJSRST52411240

Deep Learning-Based Web Crawler Page Rank Algorithm for Enhanced Search Relevance

Praveenkumar G D*[1], Sadhanayaki S[2]

[1]Assistant Professor, Department of CT – UG, Kongu Engineering College –Erode, Tamilnadu, India

[2]Assistant Professor, Department of Computer Technology and Information Technology, Kongu Arts and Science College (Autonomous), Erode, Tamilnadu, India

ARTICLEINFO

Article History:

Accepted: 10 March 2024

Published: 20 March 2024

Publication Issue :

Volume 11, Issue 2

March-April-2024

Page Number :

289-295

ABSTRACT

A web crawler page rank algorithm employing deep learning techniques aims to revolutionize the process of indexing and ranking web pages by leveraging neural networks. By analyzing content, context, and user behavior patterns, the algorithm improves relevance, adapts dynamically to evolving web content, enhances user experience, scales efficiently, and remains robust against manipulation. This approach promises to deliver more enhance the accuracy and efficiency of web crawlers in ranking web pages.

Keywords: Web Page, Web Crawler, Feature Extraction, RNN, Page Rank Algorithm.

I. INTRODUCTION

Web page created with xml and html. The World Wide Web has become a very popular medium publishing. The web is rich information on gathering and making a sense of this data is difficult because publication on the web is unorganized. Using a web browser or search engine user needed data to find specific information on the web. The user needed data is convert to query format in web browser or search engine or retrieve a big data from data base these big data may be structure or semi structure[8]. In the world of web crawling and page ranking algorithms, the advent of deep learning methodologies has revolutionized the way we approach analyzing and understanding vast amounts of data. Deep neural networks, a subset of artificial intelligence, have shown great promise in improving the efficiency and accuracy of web crawlers and page ranking systems. Deep neural networks are a type of machine learning algorithm inspired by the structure and function of the human brain. They consist of multiple layers of interconnected nodes, or artificial neurons, that work together to process and analyze data. These networks are capable of learning complex patterns and relationships in data, making them ideal for tasks like image recognition, natural language processing, and, in our case, web crawling and page ranking. Web

crawling, the process of systematically browsing the internet to index and collect information from websites, can benefit greatly from deep learning methodologies[1]. Traditional web crawlers rely on predefined rules and heuristics to navigate and extract data from web pages. However, deep learning algorithms can adapt and learn from the data they encounter, improving their performance over time. By training deep neural networks on large datasets of web pages, web crawlers can better understand the structure and content of websites, making them more effective at discovering and indexing relevant information.

PageRank algorithm, originally developed by Larry Page and Sergey Brin at Google, revolutionized web search by ranking web pages based on their importance and relevance. Integrating Recurrent Neural Networks (RNNs) into the PageRank algorithm can offer a novel approach to web page ranking that leverages the sequential nature of web interactions. In this enhanced algorithm, RNNs can be employed to model the dynamic relationships between web pages over time. By treating the web crawling process as a sequence of interactions between pages, RNNs can capture the temporal dependencies and evolving link structures inherent in the web graph. This enables the algorithm to adaptively update page rankings based on the changing connectivity patterns observed during crawling [2]. Deep learning can enhance these algorithms by analyzing more complex features and patterns in web data, leading to more accurate and personalized page rankings. By incorporating deep neural networks into page ranking systems, search engines can provide users with more relevant and reliable search results, improving the overall user experience.

II. Methodology

A web crawler page rank algorithm utilizing deep learning aims to enhance the accuracy and efficiency of traditional page ranking methods by leveraging the power of neural networks. The primary objectives of such an algorithm include,

Improved Relevance: Deep learning models can analyze the content and context of web pages more comprehensively, allowing for a finer understanding of relevance. By considering various factors such as textual content, semantic meaning, and user behavior patterns, the algorithm aims to prioritize pages that are most relevant to a given query or topic[3].

Dynamic Adaptation: The algorithm should be capable of adapting to evolving web content and user preferences. Deep learning techniques enable the model to continuously learn from new data and adjust its ranking criteria accordingly, ensuring that search results remain up-to-date and reflective of current trends.

Enhanced User Experience: By accurately ranking pages based on relevance and quality, the algorithm seeks to improve the overall user experience. This includes presenting users with more informative and engaging search results, reducing the need for manual refinement or sifting through irrelevant content.

Scalability and Efficiency: Deep learning algorithms can efficiently process large volumes of web data, making them suitable for scalable web crawling and ranking tasks. By optimizing resource utilization and computational efficiency, the algorithm aims to deliver timely and reliable search results even in the face of increasing data volumes and user demands.

Robustness to Manipulation: With the proliferation of spam and manipulation techniques on the web, the algorithm should be robust against attempts to artificially inflate page rankings. Deep learning models can detect patterns indicative of manipulative behavior and adjust ranking scores accordingly, ensuring fair and trustworthy search results for users.

1.1 PREPROCESSING

Remove HTML Tags: Use a HTML parser library such as BeautifulSoup or lxml in Python to parse the HTML content of each web page. Extract only the text content within HTML tags while discarding any markup, scripts, or styling information.

Remove Special Characters: Remove special characters such as punctuation marks, symbols, and numerical digits from the extracted text. Regular expressions or built-in string manipulation functions can be used to achieve this.

Convert to Lowercase: Convert all text to lowercase to ensure consistency in word representations and avoid duplicate entries caused by case differences[4].

Tokenization: Tokenize the preprocessed text into individual words or tokens. This involves splitting the text into a list of words based on whitespace or punctuation boundaries.
Libraries such as NLTK (Natural Language Toolkit) or spaCy can be used for tokenization.

Example :
["Natural", "language", "processing", "NLP", "is", "a", "subfield", "of", "artificial", "intelligence", "AI", "that", "focuses", "on", "the", "interaction", "between", "computers", "and", "humans", "through", "natural", "language", ".", "NLP", "techniques", "enable", "computers", "to", "understand", ",", "interpret", ",", "and", "generate", "human", "language", ",", "facilitating", "tasks", "such", "as", "language", "translation", ",", "sentiment", "analysis", ",", "and", "information", "retrieval", "."]

Remove Stopwords: Remove stopwords, which are commonly occurring words that typically do not carry much semantic meaning (e.g., "the", "is", "and"). Use predefined lists of stopwords available in libraries like NLTK or spaCy, or custom stopword lists tailored to specific domains or languages.

Stemming or Lemmatization: Apply stemming or lemmatization to normalize words by reducing them to their base or root form. Stemming algorithms like Porter or Snowball stemmer and lemmatization tools such as WordNet Lemmatizer can be employed for this purpose[5].

Example :
["natur", "languag", "process", "nlp", "is", "a", "subfield", "of", "artifici", "intellig", "ai", "that", "focus", "on", "the", "interact", "between", "comput", "and", "human", "through", "natur", "languag", ".", "nlp", "techniqu", "enabl", "comput", "to", "understand", ",", "interpret", ",", "and", "gener", "human", "languag", ",", "facilit", "task", "such", "as", "languag", "translat", ",", "sentiment", "analysi", ",", "and", "inform", "retriev", "."]

Filter Short or Rare Words: Remove very short words (e.g., single characters) or words that occur rarely in the corpus, as they are unlikely to contribute meaningfully to the ranking process.

Concatenate Tokens: Concatenate the tokens back into a single string or sequence of words, ready for further processing or feature extraction. This concatenated representation provides a standardized and normalized version of the original text, ready for further processing such as feature extraction, modeling, or indexing within the web crawler page rank algorithm.

A web crawler page rank algorithm using deep learning involves leveraging neural networks to evaluate and prioritize web pages based on their relevance and importance within a given domain. Unlike traditional page rank algorithms that rely on link analysis and graph theory, deep learning approaches employ sophisticated models to

understand the semantic context of web pages and their relationships. In this algorithm, the web crawler collects data from various web pages, including textual content, meta-information, and hyperlink structure. This data is then fed into a deep learning model, typically a convolutional neural network (CNN) or recurrent neural network (RNN)[6,7], which learns to extract meaningful features and patterns from the web page content. The deep learning model is trained using labeled data, where the relevance or importance of each web page is provided as a target variable. This could be based on factors such as user engagement metrics, inbound links, or domain authority.

To calculate ranking scores for each web page based on the learned model and extracted features, the web crawler page rank algorithm employs a combination of traditional ranking methodologies and deep learning techniques. After preprocessing the textual data and extracting relevant features from each web page, including traditional signals like TF-IDF and link analysis metrics, as well as deep learning-derived features capturing semantic meaning and user interaction patterns, the algorithm proceeds to compute ranking scores. The accuracy and effectiveness of the ranking process, the web crawler page rank algorithm combines traditional ranking scores with deep learning-derived scores using weighted averaging or ensemble methods. This fusion strategy leverages the complementary strengths of both approaches to better capture the multidimensional aspects of web page relevance and quality. Traditional ranking scores, such as TF-IDF, link analysis metrics, and content freshness, provide valuable signals reflecting factors like keyword relevance, authority, and recency. On the other hand, deep learning-derived scores capture nuanced semantic meanings, user interaction patterns, and contextually relevant features extracted from textual content and user behavior.

In the process of constructing feature vectors to represent each web page, the algorithm first extracts relevant features from the preprocessed text data. After cleansing and standardizing the textual content by removing noise, HTML tags, and stopwords, the algorithm identifies key signals indicative of web page relevance, quality, and user engagement. These features encompass a range of traditional metrics like TF-IDF scores, term frequencies, and link analysis metrics, alongside deep learning-derived features such as semantic embeddings and sentiment analysis scores. Once these features are computed for each web page, they are organized into feature vectors. Each feature vector serves as a numerical representation of a web page, encapsulating its characteristics based on the extracted features. By consolidating diverse information from the text data and other relevant sources, these feature vectors provide a holistic view of each web page's content, structure, and significance within the web ecosystem. This structured representation facilitates subsequent analysis and modeling tasks, enabling the algorithm to accurately rank and present search results to users based on their relevance and quality. After extracting features from the preprocessed text data, the next crucial step is to combine these features into feature vectors representing each web page. Each feature vector consists of numerical values that represent the selected features for a particular web page. It's imperative to ensure that these feature vectors maintain the same order or indexing as the corresponding web pages for alignment purposes. To achieve this, the algorithm organizes the extracted features into structured vectors, where each dimension corresponds to a specific feature. For instance, if a web page has features such as TF-IDF scores, term frequencies, and link analysis metrics, the feature vector for that page would contain numerical values representing these metrics in a consistent order.

By maintaining alignment between the feature vectors and the corresponding web pages, the

algorithm ensures that each vector accurately represents the characteristics of its associated web page. This structured representation facilitates further analysis, modeling, and ranking tasks within the algorithm, enabling it to effectively assess and prioritize search results based on their relevance and quality. Ultimately, by constructing feature vectors in this manner, the algorithm enhances its ability to deliver accurate and meaningful search experiences to users. The ranking capabilities of the web crawler page rank algorithm, a deep learning model such as a convolutional neural network (CNN) or recurrent neural network (RNN) is designed and trained. This model is trained using labeled data, where the labels represent relevance scores or user feedback on search results. The training process involves fine-tuning the model parameters using optimization algorithms like stochastic gradient descent (SGD) or Adam.

In the design phase, the architecture of the deep learning model is determined based on the nature of the data and the complexity of the ranking task. For instance, a CNN may be suitable for extracting features from textual content or images, while an RNN may be more adept at capturing sequential dependencies in user behavior data. The model architecture typically consists of multiple layers, including input, hidden, and output layers, with activation functions to introduce non-linearity. Once the model architecture is defined, it is trained using labeled data, where each sample is associated with a relevance score or user feedback. The training data is split into training and validation sets to assess the model's performance during training and prevent overfitting. During training, the model learns to map input features to relevant output scores through iterative adjustments of its parameters.

The optimization process involves updating the model parameters to minimize the discrepancy between predicted relevance scores and ground truth labels. Optimization algorithms like stochastic gradient descent (SGD) or Adam are commonly used to adjust the parameters in the direction of steepest descent of the loss function. These algorithms employ techniques such as gradient descent and momentum to efficiently navigate the parameter space and converge towards optimal solutions. Through iterative training and optimization, the deep learning model learns to accurately rank web pages based on their relevance and quality. Fine-tuning the model parameters ensures that it captures intricate patterns and dependencies in the data, leading to improved performance in ranking search results. By leveraging deep learning techniques in this manner, the web crawler page rank algorithm enhances its ability to deliver highly relevant and personalized search experiences to users.

Once the web pages have been assigned ranking scores through the algorithm, the next crucial step is to sort these pages based on their scores in descending order to generate search results. This ensures that the most relevant and high-quality pages appear at the top of the search engine result pages (SERPs), providing users with immediate access to the most pertinent information. Sorting the web pages based on their ranking scores involves arranging them in a ranked list, where pages with higher scores are positioned towards the top of the list, while those with lower scores are placed further down. This ranking process ensures that users are presented with the most relevant and valuable content first, optimizing their search experience and minimizing the effort required finding relevant information. In addition to sorting the pages, the algorithm also generates relevant snippets or summaries for each page to provide users with a quick overview of the content. These snippets may include excerpts from the page's text, metadata information such as the page title and URL, or other relevant details extracted from the page's content.

Once the search results are sorted and snippets are generated, they are presented to users in the search

engine result pages (SERPs). Users can then browse through the results, click on the links to access the full content of the pages, and find the information they are looking for. By sorting the web pages based on their ranking scores and providing relevant snippets, the algorithm ensures that users are presented with highly relevant and informative search results that meet their information needs effectively. This enhances the overall search experience, increasing user satisfaction and engagement with the search engine platform. Continuous gathering of user feedback and engagement metrics, such as click-through rates (CTRs) and dwell time, plays a pivotal role in refining the web crawler page rank algorithm to enhance search relevance and user satisfaction. By incorporating user feedback into the ranking algorithm, the system can iteratively adapt and improve its performance based on real-world user interactions.

As users interact with the search results presented to them, their behavior provides valuable signals about the relevance and quality of the content. Metrics like click-through rates (CTRs), which measure the proportion of users who click on a search result after viewing it, and dwell time, which measures the duration users spend on a page after clicking through, offer insights into user satisfaction and engagement. Incorporating user feedback into the ranking algorithm involves analyzing these engagement metrics and adjusting the ranking scores accordingly. Pages that receive higher CTRs and longer dwell times are considered more relevant and valuable to users, and their ranking scores can be boosted accordingly. Conversely, pages with low engagement metrics may be downgraded in the ranking to reflect their lower relevance or quality. The algorithm iteratively learns from user feedback, continuously refining its ranking criteria and adapting to changing user preferences and search trends. This dynamic feedback loop enables the algorithm to stay responsive to user needs and preferences, ensuring that search results remain relevant and satisfying over time.

By leveraging user feedback to inform the ranking algorithm, the web crawler page rank algorithm can deliver increasingly personalized and relevant search experiences to users. This iterative improvement process enhances user satisfaction, engagement, and trust in the search engine platform, ultimately leading to a more effective and valuable search experience for all users. During the training process, the model adjusts its parameters to minimize the difference between predicted and actual page ranks, effectively learning to assign higher scores to pages that are more likely to be relevant or important within the given domain. Once trained, the deep learning model can then be used to evaluate and rank new web pages encountered by the crawler. By considering both the content of the page and its contextual relationships with other pages, the algorithm can generate more accurate and contextually relevant page rankings compared to traditional methods.

Overall, leveraging deep learning in web crawler page rank algorithms allows for more sophisticated analysis of web page content and relationships, leading to improved search engine performance and user experience. In the context of a web crawler, the RNN can be employed to parse and interpret the textual content of web pages as well as to analyze the sequential structure of HTML documents. By processing text sequentially, the RNN can capture contextual dependencies and semantic nuances that contribute to the understanding of page content. Additionally, RNNs can be utilized to model the traversal behavior of the web crawler itself. By maintaining an internal state that evolves as the crawler navigates through the web, the RNN can learn patterns in page traversal, enabling it to make more informed decisions about which links to follow and which pages to prioritize for indexing. Moreover,

RNN-based PageRank algorithms can offer improved resilience to manipulation and spam. By considering the sequential context of web interactions, the algorithm can better discern genuine user engagement from artificial or malicious behavior. This enhances the robustness and reliability of the ranking system, ensuring that high-quality and relevant content is prioritized in search results.

III.CONCLUSION

In conclusion, the integration of deep learning methodologies, specifically deep neural networks, has brought new possibilities and advancements to web crawling and page ranking algorithms. The proposed deep learning-based page ranking algorithm represents a promising direction for advancing the state-of-the-art in web search technology. By leveraging the capabilities of neural networks, we can address the challenges posed by the ever-growing web landscape and provide more relevant and personalized search results to users.

IV. References

[1]. Y. Su, "Research on Website Phishing Detection Based on LSTM RNN," *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Chongqing, China, 2020, pp. 284-288, doi: 10.1109/ITNEC48623.2020.9084799

[2]. N. M. Rezk, M. Purnaprajna, T. Nordström and Z. Ul-Abdin, "Recurrent Neural Networks: An Embedded Computing Perspective," in *IEEE Access*, vol. 8, pp. 57967-57996, 2020, doi: 10.1109/ACCESS.2020.2982416.

[3]. P. A. Naidu, K. D. K. Yadav, B. Meena and Y. V. N. Meesala, "Sentiment Analysis By Using Modified RNN And A Tree LSTM," *2022 International Conference on Computing, Communication and Power Technology (IC3P)*, Visakhapatnam, India, 2022, pp. 6-10, doi: 10.1109/IC3P52835.2022.00012.

[4]. S. Roopak and T. Thomas, "A Novel Phishing Page Detection Mechanism Using HTML Source Code Comparison and Cosine Similarity," in 2014 Fourth International Conference on Advances in Computing and Communications, 2014, pp. 167–170.

[5]. S. Marchal, J. Francois, R. State, and T. Engel, "PhishStorm: Detecting Phishing With Streaming Analytics," IEEE Transactions on Network and Service Management, vol. 11, no. 4, pp. 458–471, 2014.

[6]. A.K. Sangaiah, M. Sadeghilalimi, A.A.R. Hosseinabadi and W. Zhang, ―Energy consumption in point-coverage wireless sensor networks via bat algorithm‖. IEEE Access vol. 7, pp. 180258-180269, 2019.

[7]. R. Katarya and O.P. Verma, ―An effective web page recommender system with fuzzy c-mean clustering‖. Multimedia Tools and Applications vol. 76, no. 20, pp. 21481-21496, 2017.

[8]. Praveenkumar, G. D., and R. Gayathri. "A Process of Web Usage Mining and Its Tools." International Journal of Advanced Research in Science, Engineering and Technology 2.11 (2015).

Cite this article as :

Praveenkumar G D, Sadhanayaki S, "Deep Learning-Based Web Crawler Page Rank Algorithm for Enhanced Search Relevance", International Journal of Scientific Research in Science and Technology (IJSRST), Online ISSN : 2395-602X, Print ISSN : 2395-6011, Volume 11 Issue 2, pp. 289-295, March-April 2024. Available at doi : https://doi.org/10.32628/IJSRST52411240

Journal URL : https://ijsrst.com/IJSRST52411240