Neural Web Crawling*

Extended Abstract

Francesca Pezzuti[1], Sean MacAvaney[2] and Nicola Tonellotto[1]

[1]University of Pisa, Pisa, Italy
[2]University of Glasgow, Glasgow, UK

Abstract

Given the vast scale of the Web, crawling prioritisation techniques based on graph traversal, popularity, link analysis, and textual content are frequently applied to surface documents that are most likely to be valuable. While these techniques have proven effective for keyword-based search, retrieval methods and user search behaviours are shifting from keyword-based matching to natural language semantic matching. Semantic matching and quality signals have been applied during ranking with great success, and recently, researchers have proposed to exploit them also to prioritise the frontier of Web crawlers. To investigate more on this, we propose two novel neural policies with the goal of surfacing content that is semantically rich and valuable for modern search needs, ultimately aligning the crawler behaviour with the recent shift towards natural language search. Our experiments on the English subset of ClueWeb22-B and the MS MARCO Web Search and Researchy Questions query sets show that, compared to existing crawling techniques, neural crawling policies significantly improve harvest rate during the early stages of crawling.

Keywords

Crawling, Web search, Quality estimation

1. Introduction

The effectiveness of search engines heavily depends on the indexed corpus: if the corpus is incomplete or filled with low-quality pages, search results could be irrelevant [10]. A *crawler* is a program that systematically traverses the Web and downloads Web pages to build and keep up-to-date such a search corpus. Its ability in prioritising high-quality pages is crucial for providing accurate and relevant results to user queries [7]. Crawlers maintain a priority queue of URLs of pages to visit, called *frontier*, and continuously download pages, extract their outgoing links, and prioritise them in the frontier to select the next pages to crawl. Traditional graph traversal algorithms like *Breadth-First Search* (BFS) can be used for traversal, without utilising any heuristic to prioritise URLs [13, 6]. In contrast, *Best-First* policies (BF) are designed to prioritize the frontier leveraging some heuristic like click-through rate [14], PageRank [16], or textual content [13, 14], although each of them comes with limitations. For instance, the most notable BF policy, PageRank, assigns higher priority to well-linked pages, but requires storing the full Web graph, involves resource-intensive computations [12], and is inaccurate on sub-graphs [1, 9]. On the other hand, content-based quality estimation using keyword matching or term frequency has been mainly used in focused crawlers [24] which are query-driven, and outside the scope of our work.

Although all these prioritisation policies work well for keyword-based queries, they either (i) ignore the textual content of pages, (ii) use it for keyword matching w.r.t. topic keywords, or (iii) use it for query-driven focused-crawling, relying on relevance signals based on term frequency and inverse document frequency, ignoring the semantics of texts. Recent advancements in contextualised Large Language Models (LLMs), have shifted search on the Web toward conversational and complex, question-based queries rather than simple keywords [23, 8]. Consequently, many search applications, such as

IIR2025: 15th Italian Information Retrieval Workshop, 3th - 5th September 2025, Cagliari, Italy

*This is an extended abstract of [18].

francesca.pezzuti@phd.unipi.it (F. Pezzuti); sean.macavaney@glasgow.ac.uk (S. MacAvaney); nicola.tonellotto@unipi.it (N. Tonellotto)
0009-0005-2364-2043 (F. Pezzuti); 0000-0002-8914-2659 (S. MacAvaney); 0000-0002-7427-1001 (N. Tonellotto)

question answering systems, LLM-based assistants, and mobile voice search, shifted their focus from short keyword queries to natural language ones [25, 22, 8, 4, 26, 21].

In this work, we argue that crawlers should also adapt to this shift. We hypothesise that, by using LLMs to estimate quality during crawling, we can improve the ability of crawlers to surface documents valuable for search tasks, particularly for natural language search. Building on the approach by Pezzuti et al. [19], we propose to prioritise the crawling frontier based on the quality scores generated by the neural quality estimators introduced in [3]. We call this approach *neural crawling*. A neural quality estimator is an LLM fine-tuned to assess the likelihood of a textual document being relevant to *any* query. Given its semantic understanding of text, the quality scores it produces capture the semantic quality of the input text. By relying that Web pages with similar quality are likely to link to each other, we propose two neural policies that exploit this property to propagate quality from the inlinking neighbourhood of each Web page. While existing studies have shown the potential of neural crawlers for targeting high-quality LLM pre-training data and for generic search tasks [19], the alignment of neural and traditional crawlers with the recent trend towards natural language search remains unexplored. To investigate this, we evaluate the crawling effectiveness on keyword queries and natural language queries, using search corpora crawled by a traditional BFS crawler, and by our proposed neural crawlers.

Our experiments on the English subset of ClueWeb22-B [15] show that, w.r.t. to BFS, our neural policies can significantly improve crawling effectiveness for natural language queries from Researchy Questions [20] (up to $+149\%$ in HR), while remaining competitive for keyword queries from MS MARCO Web Search [5] (up to $+20\%$ in HR).

2. Neural Crawling Policies

Let p denote a Web page, and $\mathcal{O}_p$ denote the set of outlinks from p, i.e., the pages that p links to. Let $\mathcal{F}$ denote the *frontier*, which stores the URLs of the pages yet to be crawled. The frontier is composed of (u, P_u) pairs, where u is a URL and P_u is its priority. A *crawling policy* is composed by: (i) a *priority-assignment function* $P : u \mapsto P_u$, typically based on heuristics such as PageRank, (ii) an *update policy*, that defines how priorities are updated when discovering a new link to a page whose URL is already in the frontier, and (iii) a *selection policy*, that decides which page to crawl next, according to its priority. For instance, the well-known Breadth-First Search (BFS) crawling policy uses a constant priority-assignment function, a First-In-First-Out selection policy, and an update policy that does not change priorities upon rediscovery. In contrast, the Best-First (BF) policy employs a priority assignment function based on a link-based quality estimation heuristic such as PageRank, a maximum priority selection policy, and advanced update policies. Inspired by the link-based nature of PageRank, we propose two neural BF crawling policies leveraging an LLM-based quality estimation heuristic to prioritise Web pages with high semantic quality during the crawling process, called QFirst and QMin.

At the core of our proposed neural BF crawling policies is the use of a LLM-based heuristic function $M_\theta : p \mapsto \mathbb{R}$, parametrised by θ and optimised to distinguish high semantic quality pages from low-quality ones. In particular, we aim at exploiting this neural heuristic in the priority-estimation function, ideally using $P : u \mapsto M_\theta(p)$ to prioritise a page p whose URL is u. However, in real-world crawling settings we cannot access the textual content of a Web page before its download. Indeed, using the ideal semantic quality as the priority when enqueueing pages into $\mathcal{F}$ is only feasible in a theoretical scenario where an oracle function has access to a page text prior its download. We refer to this oracle-based crawling policy as QOracle [19], and we use it as an upper-bound on the performance achievable by our practical neural crawling policies.

In the absence of an oracle, we can reasonably assume that the quality of a page is related to the quality of the pages it is connected to. Indeed, existing literature shows that the quality of a page is positively correlated with that of its linking neighbours. If this relationship holds, we can effectively propagate quality via link structure by using as a proxy estimate of the quality of a page that of one of its ancestors, i.e., the page that linked to it.

In our first neural crawling policy, referred to as QFirst, when processing a page p and encountering the outgoing URL $\tilde{u}$ of a new page $\tilde{p} \in \mathcal{O}_p$ for the first time, we insert $\tilde{u}$ into $\mathcal{F}$ with priority $P_{\tilde{u}} = M_\theta(p)$ and we never update it.

In our second neural crawling policy, referred to as QMin, we additionally assume that if a page is linked to a low-quality page, it is highly unlikely to be of high-quality. If this holds, we could postpone the crawling of low-quality pages by decreasing their priority whenever a link from a low-quality ancestor is discovered. In doing so, we aim to boost the prioritisation high-quality Web pages while deprioritising low-quality ones. To implement the QMin policy, when processing a page p and re-encountering an already enqueued URL $\tilde{u} \in \mathcal{F}$ of a page $\tilde{p} \in \mathcal{O}_p$, we update its priority as the minimum between the current priority and the quality of the new ancestor p, i.e., $P_{\tilde{u}} \leftarrow \min\{P_{\tilde{u}}, M_\theta(p)\}$.

We do not to propose a QMax policy, as our preliminary experiments, consistent with prior research [19, 3], suggest that neural estimators better identify low-quality than high-quality content.

3. Experimental Setup

We perform a single-threaded simulation of the crawling process on the English subset of ClueWeb22-B (CW22B-eng), which contains $87M$ head Web pages from ClueWeb22 (CW22) [15], a recently released corpus crawled by a commercial search engine. In our simulations, Web pages are crawled sequentially, and we assume a constant per-page crawling time. In real-world scenarios, crawlers iteratively download Web pages and stop after periods of duration T to allow retrievers to update their index. To simulate this, we measure the crawling effectiveness every $T = 2.5M$ crawled pages. We start to crawl from $100k$ randomly selected seed URLs, and we reach a total of $29M$ pages. The source code to reproduce our experiments is publicly available on Github[1].

Queries. We use MS MARCO Web Search (MSM-WS) [5] and Researchy Questions (RQ) [20] query sets, both generated from the logs of commercial search engines. The former contains queries reflecting a real query distribution and relevance labels over CW22 extracted from a real click-log, with explicit relevance assessments. The latter contains multi-perspective, non-factoid English queries, and a click distribution over CW22. For each query in RQ, we consider the most clicked page to be relevant. Since we work with a subset of CW22B, but both query sets are related to CW22, we excluded queries without relevant results in CW22B-eng. MSM-WS queries are generally shorter compared to RQ, with a narrow scope and mainly keyword-based. Moreover, MSM-WS queries contain fewer interrogatives such as "how" and "why", while RQ queries have a broader scope and are similar to natural language.

Neural Quality Estimation. In our experiments, we use the *QT5-Small*[2] neural quality estimator [3], fine-tuned on the MSM-WS training set by Pezzuti et al. [19].

Effectiveness. There exist several approaches to compute the effectiveness of crawling policies [11, 7]. In this work, we use the Harvest Rate (HR), one of the most widely used metrics that can be used for this [2, 17]. Let $\mathcal{R}^{\mathcal{Q}}$ denote the set of all pages relevant to at least a query q in a query set $\mathcal{Q}$. At time t, for the query set $\mathcal{Q}$, the harvest rate $HR(\mathcal{Q}, t)$ is defined as:

$$HR(\mathcal{Q}, t) = |\mathcal{R}^{\mathcal{Q}}_t| / t,$$

where $\mathcal{R}^{\mathcal{Q}}_t$ is the subset of relevant pages crawled up to time t. As noted before, the page crawl time is our time unit, and t corresponds to the crawling of t pages. This metric measures the crawl ability to maximise the number of crawled relevant pages while minimising that of irrelevant ones.

Baseline. We compare our policies against BFS, the simplest yet effective policy. We do not compare with PageRank since prior research showed that on small graphs PageRank is not accurate and BFS is stronger [1, 9, 13]. For significance testing, we use a two-tailed Z-test for proportions with $p = 0.01$.

[1] https://github.com/fpezzuti/neural_crawling

[2] https://huggingface.co/macavaney/qt5-small-msw

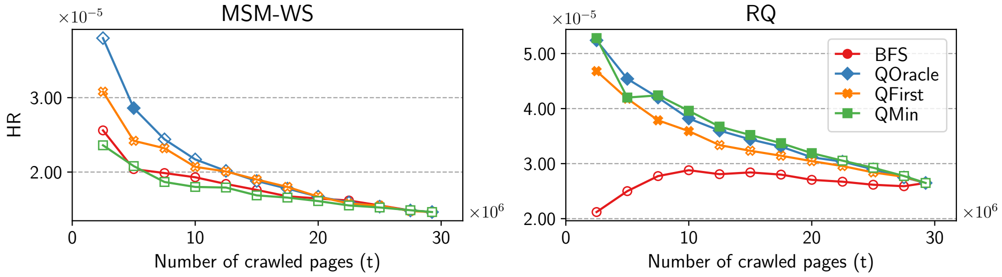

Figure 1: Assessment of neural policies and BFS in terms of Harvest Rate. MSM-WS (left) and RQ (right). Statistically significant differences w.r.t. BFS are denoted with *filled circles* ($p = 0.01$).

4. Experimental Results

To investigate if using neural policies during crawling helps a crawler find relevant pages earlier than BFS, Figure 1 shows the HR over time for our oracle neural policy QOracle, our two practical neural policies QFirst and QMin, and BFS, on both the MSM-WS and RQ datasets.

From the figure, we note that the QOracle policy initially exhibits superior performance w.r.t. all other policies in surfacing pages relevant to keyword queries in MSM-WS, and is immediately followed by the QFirst policy. On natural language queries, all our neural crawlers substantially outperform BFS, and the QMin policy attains almost the same HR as QOracle.

The fact that QMin exhibits comparable performance to QOracle on RQ, while being unexpectedly on par with BFS on MSM-WS, suggests that most pages valuable for natural language search can be easily reached by postponing the exploration of low-quality links, while some of the pages valuable for keyword-oriented search may only be reachable through low-quality links. Thus, reluctance in following these links may hamper the discovery of valuable Web pages located deeper in the Web graph.

Meanwhile, QFirst, our simplest policy, significantly outperforms BFS on both query sets, and achieves competitive performance w.r.t. the other methods without introducing excessive overhead. Unlike QOracle and QMin, which rely on a greedier prioritisation and favour exploitation, QFirst is more exploration-oriented as it relies on noisier estimates. As a result, it has higher chances of discovering valuable pages only reachable throughout local minima.

To conclude, our experiments show that, w.r.t. to BFS, our QMin policy can significantly improve crawling effectiveness for natural language queries from RQ by up to $+149\%$ in early HR, while remaining competitive for keyword queries from MSM-WS, with up to $+20\%$ in early HR.

5. Conclusion and Future Work

In this paper, we proposed two neural policies for neural crawling, both leveraging neural quality estimators to prioritise the early crawl of semantically high-quality pages.

We compared our proposed policies with an oracle policy, and with the well-established BFS baseline in terms of crawling effectiveness. Our findings reveal that, especially for natural language queries, we can markedly improve the early effectiveness of the crawler by using a neural policy in place of a traditional one. While our results show the promise of our approach, we recognise several limitations of this work that open up meaningful directions for future research. First, our experiments were conducted in a controlled, simulated setting; the effectiveness of our approach has not been validated in real-world, multi-threaded environments, which are typically subject to practical constraints like politeness policies, host reachability issues, and others. Second, further investigation is needed to better understand the potential biases introduced by neural quality estimators, particularly in terms of fairness and transparency. Third, we our proposed policies may be vulnerable to adversarial manipulation, and their robustness to such attacks should has yet to be explored.

We leave for future work experiments on other Web corpora and query sets, as well as experiments with other policies and other baseline comparisons.

Declaration on Generative AI

During the preparation of this work, the author did not use any AI tool.

References

[1] Paolo Boldi, Massimo Santini, and Sebastiano Vigna. Do Your Worst to Make the Best: Paradoxical Effects in PageRank Incremental Computations. In *Proc. WAW*, pages 168–180, 2004.

[2] Soumen Chakrabarti, Martin van den Berg, and Byron Dom. Focused crawling: a new approach to topic-specific Web resource discovery. *Computer Networks*, 31(11-16):1623–1640, 1999.

[3] Xuejun Chang, Debabrata Mishra, Craig Macdonald, and Sean MacAvaney. Neural Passage Quality Estimation for Static Pruning. In *Proc. SIGIR*, pages 174–185, 2024.

[4] Jia Chen, Jiaxin Mao, Yiqun Liu, Fan Zhang, Min Zhang, and Shaoping Ma. Towards a Better Understanding of Query Reformulation Behavior in Web Search. In *Proc. WWW*, pages 743–755, 2021.

[5] Qi Chen, Xiubo Geng, Corby Rosset, Carolyn Buractaon, Jingwen Lu, Tao Shen, Kun Zhou, Chenyan Xiong, Yeyun Gong, Paul Bennett, Nick Craswell, Xing Xie, Fan Yang, Bryan Tower, Nikhil Rao, Anlei Dong, Wenqi Jiang, Zheng Liu, Mingqin Li, Chuanjie Liu, Zengzhong Li, Rangan Majumder, Jennifer Neville, Andy Oakley, Knut Magne Risvik, Harsha Vardhan Simhadri, Manik Varma, Yujing Wang, Linjun Yang, Mao Yang, and Ce Zhang. MS MARCO Web Search: A Large-scale Information-rich Web Dataset with Millions of Real Click Labels. In *Proc. WWW*, pages 292–301, 2024.

[6] Paul M.E. De Bra and Reinier D.J. Post. Information retrieval in the World-Wide Web: Making client-based searching feasible. *Computer Networks*, 27(2):183–192, 1994.

[7] Dennis Fetterly, Nick Craswell, and Vishwa Vinay. The impact of crawl policy on web search effectiveness. In *Proc. SIGIR*, pages 580–587, 2009.

[8] Ido Guy. Searching by Talking: Analysis of Voice Queries on Mobile Web Search. In *Proc. SIGIR*, pages 35–44, 2016.

[9] Holzmann Helge, Anand Avishek, and Khosla Megha. Estimating pagerank deviations in crawled graphs. *Applied Network Science*, 4(86), 2019.

[10] Dirk Lewandowski and Nadine Höchstötter. *Web Searching: A Quality Measurement Perspective*, pages 309–340. Springer, 2008.

[11] Filippo Menczer, Gautam Pant, Padmini Srinivasan, and Miguel E. Ruiz. Evaluating topic-driven web crawlers. In *Proc. SIGIR*, pages 241–249, 2001.

[12] Filippo Menczer, Gautam Pant, and Padmini Srinivasan. Topical web crawlers: Evaluating adaptive algorithms. *ACM TOIT*, 4(4):378–419, 2004.

[13] Marc Najork and Janet L. Wiener. Breadth-first crawling yields high-quality pages. In *Proc. WWW*, pages 114–118, 2001.

[14] Liudmila Ostroumova, Ivan Bogatyy, Arseniy Chelnokov, Alexey Tikhonov, and Gleb Gusev. Crawling Policies Based on Web Page Popularity Prediction. In *Proc. ECIR*, pages 100–111, 2014.

[15] Arnold Overwijk, Chenyan Xiong, and Jamie Callan. ClueWeb22: 10 Billion Web Documents with Rich Information. In *Proc. SIGIR*, pages 3360–3362, 2022.

[16] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank Citation Ranking: Bringing Order to the Web. In *Proc. WWW*, 1999.

[17] Gautam Pant and Padmini Srinivasan. Link contexts in classifier-guided topical crawlers. *IEEE TKDE*, 18(1):107–122, 2006.

[18] Francesca Pezzuti, Sean MacAvaney, and Nicola Tonellotto. Neural Prioritisation for Web Crawling. In *Proc. ICTIR*, page 8, 2025.

[19] Francesca Pezzuti, Ariane Mueller, Sean MacAvaney, and Nicola Tonellotto. Document Quality Scoring for Web Crawling. arXiv:2504.11011, 2025.

[20] Corby Rosset, Ho-Lam Chung, Guanghui Qin, Ethan C. Chau, Zhuo Feng, Ahmed Awadallah,

Jennifer Neville, and Nikhil Rao. Researchy Questions: A Dataset of Multi-Perspective, Decompositional Questions for LLM Web Agents. arXiv:2402.17896, 2024.

[21] Artsiom Sauchuk, James Thorne, Alon Y. Halevy, Nicola Tonellotto, and Fabrizio Silvestri. On the Role of Relevance in Natural Language Processing Tasks. In *Proc. SIGIR*, pages 1785–1789, 2022.

[22] Khan Tajmir, Rashid Umer, and Rehman Abdur. End-to-end pseudo relevance feedback based vertical web search queries recommendation. *Multimedia Tools and Applications*, 83(31):75995–76033, 2024.

[23] Johanne R. Trippas, Sara Fahad Dawood Al Lawati, Joel Mackenzie, and Luke Gallagher. What do Users Really Ask Large Language Models? An Initial Log Analysis of Google Bard Interactions in the Wild. In *Proc. SIGIR*, pages 2703–2707, 2024.

[24] Lalit Kumar Tyagi, Anish Gupta, and Vibhash Singh Sisodia. A New Era of Web Mining: Innovative Approaches in Focused Web Crawling for Domain-Specific Information. In *Proc. ICTACS*, pages 1–6, 2023.

[25] Ryen W. White. Advancing the Search Frontier with AI Agents. *Commun. ACM*, 67(9):54–65, 2024.

[26] Yutong Xie, Zhaoying Pan, Jinge Ma, Luo Jie, and Qiaozhu Mei. A Prompt Log Analysis of Text-to-Image Generation Systems. In *Proc. WWW*, pages 3892–3902, 2023.