Peer-to-Peer Learning-to-Rank

Marcel Gregoriadis

I. Introduction

Decentralization facilitates transparent, censorship-resistant, and democratic systems. The shift towards decentralized technologies like cryptocurrencies and Web3 is motivated by the desire to redistribute ownership and control away from centralized authorities, empowering individual users with greater autonomy and participation in the digital ecosystem.

In the same way that search engines like Google laid the foundation for the usability and popularity of the World Wide Web (WWW), effective search is also paramount for the success of decentralized applications. P2P file sharing services like BitTorrent and IPFS are flourishing. Yet, search tasks are usually outsourced to centralized indices (e.g., The PirateBay [**?**] and IPFS Search [1]). The reliance on centralized search indices presents a critical vulnerability. Providers of these services have the power to manipulate search results, and hide or suppress content. They can also succumb to external pressure [**?**].

II. Problem Description

Although various algorithms for decentralized content retrieval have been proposed [2], [9], [14], [15], centralized search indices remain popular. While decentralized content retrieval is notoriously challenging [12], we argue that one of the main reasons for its limited adoption is the **lack of descriptive metadata**. In the WWW, files are addressed by their location (the hosting website) rather than their content. Websites, being text-based, naturally provide a lot of auxiliary information on top of the content. Moreover, the intricate hyperlink structure of the web aids content discovery and relevance ranking, as has been exploited in Google's PageRank [6] algorithm. Website hosts furthermore have economic incentives (e.g., through ad revenue) to direct users to ther website, and thus to curate good quality metadata.

Content providers in P2P file sharing systems **lack these incentives**. Furthermore, files in these systems are usually non-textual (e.g., video and music) and, in the case of torrents, merely described by their *name*. For search, relying on the name alone is not only insufficient from an information perspective but it is also not trustworthy. The file name could be misleading or simply of inferior quality to a similar torrent. This is where **ranking of retrieved search result candidates becomes paramount**. While PageRank offered an elegant solution for the traditional web, the same principle cannot be applied to systems like BitTorrent. P2P file sharing, however, has another very powerful property, which is the number of seeders that provide a file: the more nodes that keep the data, the more popular it is assumed to be. Ranking according to this metric has been employed in The PirateBay, as well as for Tribler [**?**] and other decentralized search engines [22]. Retrieving this metric on-the-fly, however, is very slow. While researchers proposed to maintain this information in a DHT [22], security threats like those posed by Sybils have so far been left ignored.

III. Proposed Solution

We propose a novel strategy for ranking in P2P systems, for which we borrow a technique from traditional information retrieval. Learning-to-Rank (LTR) [18] describes a set of machine learning techniques that has been developed and applied to web search for decades. Based on historical user behavior (search queries, clickthroughs, etc.), a model is trained to learn the rules and patterns that guide relevance ranking. In that way, instead of asking users to provide metadata explicitly (e.g., through keywords or star-ratings), LTR leverages implicit cues (i.e., query content, selected result, and other user signals). Not only does learning from implicit signals **remove the need for incentivization**, but it is also expected with **higher accuracy** []. When applying LTR to P2P systems, we consider two possible implementations: *local-only* and *collaborative*.

A. Local-Only Strategy

In the *local-only* implementation, each peer trains their own LTR model on their personal search history and interactions only. The strength of this solution lies in its security. As training and inference happen only locally, and training data is sourced from the peer's own interactions, ranking quality cannot be corrupted by adversarial peers. Specializing the ranking model on the user level also yields the benefit of personalization, as peers can have different preferences based on their region, language, culture, or taste [23], [16]. However, when peers learn only from their own history, they miss out on collective knowledge, limiting their models ability to generalize to new or infrequent queries, especially when its own training data is scarce.

B. Collaborative Strategy

Ideally, we want to use the knowledge gained from other peers to improve our model's performance while still optimize for personal taste. To this end, we also propose a *collaborative* strategy that employs LTR with multi-task learning, treating each peers ranking function as one distinct task. Specifically, we use gossip learning [4] to establish globally shared layers of the LTR model, while reserving the final layers for local training and personalization.

IV. Related Works

The idea of using MTL for personalized LTR is not new; it has been previously proposed for region- or language-specific adaptations in centralized services like web search [3] and e-commerce platforms [16]. Ma et al. [19] proposed an architecture combining MTL with Mixture-of-Experts (MoE) for movie recommendations (fundamentally a ranking problem). In their work, the authors use MTL with tasks representing different objectives, e.g., watching a movie, purchasing a movie, and liking a movie.

While not with the application of LTR in mind, more recent work has examined MTL in P2P systems. Bouchra et al. [5] propose

RELATED WORK in Multi-task peer-to-peer learning using an encoder-only transformer: - In a peer-to-peer environment, Mohammadi et al. [21] implemented a system where individual agents retained specific expertise through local skills while simultaneously sharing general knowledge with nearby agents. This collaborative approach allowed agents to benefit from knowledge exchange while maintaining their proficiency in solving localized data, improving overall performance - Zantedeschi et al. [6] introduced a method to enable the dynamic formation of peer-to-peer connections by exploiting the resemblance among agents local linear models using empirical loss on the agents local dataset. Building on this idea, several other studies have further applied this approach to neural networks, yielding better model performance in scenarios with heterogeneous data distribution [29-33].

V. Background

A. Learning-to-Rank (LTR)

[7], [18]

B. Multi-Task Learning (MTL)

MTL [8] is a strategy for learning multiple related tasks. The idea is that related tasks differ in some attributes and are similar in others. It works by sharing a common set of features or representations among tasks (the shared layers), while allowing task-specific adaptations (task-specific layers). This shared learning helps the model generalize better by leveraging information across tasks. As a result, MTL achieves better prediction accuracy compared to training related tasks separately or training them together without differentiation [20], [17], [13].

C. Gossip Learning

Decentralized learning [4], [11], [10], also called *gossip learning*, provides a framework for collaborative training on edge devices. Peers train on their locally generated data and disseminate their updated model parameters to other peers. Incoming model parameters are aggregated and merged with the local model. Ultimately, models converge network-wide.

References

[1] ipfs-search.com. ipfs-search.com. [Accessed 21-08-2024].

[2] Presearch — presearch.com. https://presearch.com/. [Accessed 16-08-2024].

[3] Jing Bai, Ke Zhou, Guirong Xue, Hongyuan Zha, Gordon Sun, Belle Tseng, Zhaohui Zheng, and Yi Chang. Multi-task learning for learning to rank in web search. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1549–1552, 2009.

[4] Michael Blot, David Picard, Matthieu Cord, and Nicolas Thome. Gossip training for deep learning. *arXiv preprint arXiv:1611.09726*, 2016.

[5] Amaury Bouchra Pilet, Davide Frey, and François Taïani. Simple, efficient and convenient decentralized multi-task learning for neural networks. In *Advances in Intelligent Data Analysis XIX: 19th International Symposium on Intelligent Data Analysis, IDA 2021, Porto, Portugal, April 26–28, 2021, Proceedings 19*, pages 37–49. Springer, 2021.

[6] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.

[7] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136, 2007.

[8] Rich Caruana. Multitask learning. *Machine learning*, 28:41–75, 1997.

[9] Michael Christen. Home - YaCy — yacy.net. https://yacy.net/. [Accessed 16-08-2024].

[10] Martijn De Vos, Sadegh Farhadkhani, Rachid Guerraoui, Anne-Marie Kermarrec, Rafael Pires, and Rishi Sharma. Epidemic learning: Boosting decentralized learning with randomized communication. *Advances in Neural Information Processing Systems*, 36, 2024.

[11] István Hegedűs, Gábor Danner, and Márk Jelasity. Decentralized learning works: An empirical comparison of gossip learning and federated learning. *Journal of Parallel and Distributed Computing*, 148:109–124, 2021.

[12] Navin Keizer, Onur Ascigil, Michal Król, Dirk Kutscher, and George Pavlou. A survey on content retrieval on the decentralised web. *ACM Computing Surveys*, 56(8):1–39, 2024.

[13] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018.

[14] Nawras Khudhur and Satoshi Fujita. Siva-the ipfs search engine. In *2019 Seventh International Symposium on Computing and Networking (CANDAR)*, pages 150–156. IEEE, 2019.

[15] Mingyu Li, Jinhao Zhu, Tianxu Zhang, Cheng Tan, Yubin Xia, Sebastian Angel, and Haibo Chen. Bringing decentralized search to decentralized services. In *15th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 21)*, pages 331–347, 2021.

[16] Pengcheng Li, Runze Li, Qing Da, An-Xiang Zeng, and Lijun Zhang. Improving multi-scenario learning to rank in e-commerce by exploiting task relationships in the label space. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2605–2612, 2020.

[17] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Adversarial multi-task learning for text classification. *arXiv preprint arXiv:1704.05742*, 2017.

[18] Tie-Yan Liu et al. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331, 2009.

[19] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1930–1939, 2018.

[20] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3994–4003, 2016.

[21] Javad Mohammadi and Soheil Kolouri. Collaborative learning through shared collective knowledge and local expertise. In *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2019.

[22] Feng Wang and Yanjun Wu. Keyword search technology in content addressable storage system. In *2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pages 728–735. IEEE, 2020.

[23] Hema Yoganarasimhan. Search personalization using machine learning. *Management Science*, 66(3):1045–1070, 2020.