RESEARCH-ARTICLE

Leveraging Topic Specificity and Social Relationships for Expert Finding in Community Question Answering Platforms

MADDALENA AMENDOLA, Istituto Di Informatica E Telematica, Pisa, Pisa, PI, Italy

ANDREA PASSARELLA, Istituto Di Informatica E Telematica, Pisa, Pisa, PI, Italy

RAFFAELE PEREGO, Institute of Information Science and Technologies "Alessandro Faedo", Pisa, PI, Italy

Published: 07 February 2026
Accepted: 08 January 2026
Revised: 29 September 2025
Received: 09 July 2024

Leveraging Topic Specificity and Social Relationships for Expert Finding in Community Question Answering Platforms

MADDALENA AMENDOLA, IIT-CNR, Italy
ANDREA PASSARELLA*, IIT-CNR, Italy
RAFFAELE PEREGO*, ISTI-CNR, Italy

Online Community Question Answering (CQA) platforms have become indispensable tools for users seeking expert solutions to their technical queries. The effectiveness of these platforms relies on their ability to identify and direct questions to the most knowledgeable users within the community, a process known as Expert Finding (EF). EF accuracy is crucial for increasing user engagement and the reliability of provided answers. We present TUEF, a *Topic-oriented User-Interaction model for EF*, which aims to fully and transparently leverage the heterogeneous information available within online CQA platforms. TUEF integrates content and social data by constructing a multi-layer graph that maps user relationships based on their answering patterns on specific topics. By combining these sources of information, TUEF identifies the most relevant users for any given question and ranks them using learning-to-rank techniques. Our findings indicate that TUEF's topic-oriented model significantly enhances performance, particularly in large communities discussing well-defined topics. Additionally, we show that the interpretable learning-to-rank algorithm integrated into TUEF offers transparency and explainability with minimal performance trade-offs. The exhaustive experiments conducted across six CQA communities show that TUEF outperforms all competitors, achieving a minimum performance boost of 42.42% in P@1, 32.73% in NDCG@3, 21.76% in R@5, and 29.81% in MRR.

CCS Concepts: • **Information systems** → **Retrieval models and ranking**; **Evaluation of retrieval results**; *Users and interactive retrieval*.

Additional Key Words and Phrases: Do, Not, Use, This, Code, Put, the, Correct, Terms, for, Your, Paper

1 Introduction

Online Community Question-Answering (CQA) platforms, such as StackOverflow and AskUbuntu, have become indispensable tools for users seeking expert solutions to their technical queries. These platforms are built upon the collaborative efforts of users who pose questions and provide answers, thus creating an extensive repository of shared knowledge. The success of CQA platforms relies on their ability to effectively identify and direct questions to the most knowledgeable experts within the community, thus engaging the users. This crucial process, known as Expert Finding (EF), is vital for ensuring the accuracy and reliability of the answers provided.

The EF task focuses on identifying and recognizing users with a high level of expertise who can offer accurate and timely responses to posted questions. It significantly enhances user engagement, trust, and satisfaction by routing questions to the most knowledgeable community members. Recently, the rise of Large Language Models (LLMs) and generative Artificial Intelligence (AI) has reshaped how users seek information online, leading to a decline in participation in online CQA platforms. Nevertheless, we argue that EF remains highly relevant and may

Authors' Contact Information: Maddalena Amendola, maddalena.amendola@iit.cnr.it, IIT-CNR, Pisa, Italy; Andrea Passarella, andrea.passarella@iit.cnr.it, IIT-CNR, Pisa, Italy; Raffaele Perego, raffaele.perego@isti.cnr.it, ISTI-CNR, Pisa, Italy.

ACM 1558-2868/2026/2-ART
https://doi.org/10.1145/3793531

even become more important in the future. Currently, LLMs and generative AI are trained on large collections of online data, often including content from CQA sites, which constrains their knowledge to information that has already been documented. As a result, they struggle to address novel questions, particularly in rapidly evolving or niche domains where up-to-date, context-sensitive, and authoritative responses are required [17]. In such contexts, expert-driven knowledge-sharing platforms will continue to play a key role, and EF frameworks are essential to connect with the most suitable human experts to ensure accurate and reliable answers.

Despite advancements in EF methodologies, there are still challenges in effectively integrating the diverse information sources available on CQA platforms. Current proposals for addressing the EF task in CQA contexts rely on information derived from the textual content of questions and answers and from features and signals that model the users' engagement with this content and their interactions within the networked community. The complexity of such interactions and the dynamic nature of expertise necessitates a comprehensive approach that can adapt to varying contexts and user behaviors, which, to the best of our knowledge, still needs to be fully addressed in the literature.

In this paper, we comprehensively address the EF task by proposing TUEF, a *Topic-oriented User-Interaction model for Expert Finding*, which aims to fully and transparently utilize the vast amount of heterogeneous information available within online question-answering communities. TUEF integrates content and social data by constructing a topic-based Multi-Layer Graph (MLG) that maps out user relationships based on their topical answering patterns. By combining these sources of information, TUEF aims to identify the most relevant and knowledgeable users for any given question. TUEF operates by first generating the MLG, where each layer represents a major topic within the community. These topics are automatically identified through the analysis of tags from past questions. Within each layer, nodes represent active users in topic discussions, and edges model the similarities and relationships among these users. When a question is posed, TUEF uses the MLG and a ranking model to determine the relevant topics and corresponding graph layers. In each relevant layer, TUEF selects candidate experts from two perspectives: (i) *Network Perspective*: identifying central users who have significant influence within the community; (ii) *Content Perspective*: identifying users who have previously answered similar questions. Navigating the graph from seed nodes identified according to these criteria, TUEF explores and collects candidate experts through appropriate exploration policies. It then extracts static and query-dependent features based on text and graph relationships for these candidates. Finally, TUEF applies Learning-to-Rank techniques to score and rank the candidates by their expected relevance to the question.

Unlike previous studies that integrate both social and content-based information, TUEF uniquely leverages the MLG to enable fine-grained expert selection, ensuring that network influence and content relevance are evaluated within the appropriate topic layers. Additionally, it introduces a new approach to node representation that enhances user interaction modeling across different topics. Specifically, each node is represented by a knowledge vector, which encodes the user's contribution to various sub-topics within a layer (macro-topic), normalized by their total contribution across all layers. This representation allows for a more context-aware and balanced assessment of expertise.

Our work aims to address the following Research Questions (RQs):

- RQ1: How does an MLG approach that integrates topic-oriented and network-based user interactions improve expert finding in CQA platforms? Additionally, how do content and network components complement each other in enhancing accuracy?
- RQ2: How does the interpretability of the ranking model affect the trade-off between expert selection accuracy and model transparency in expert finding tasks?
- RQ3: How do different evaluation configurations influence the assessment of expert ranking performance in CQA platforms?
- RQ4: Is the proposed approach scalable?

This work builds upon a previous paper [1], extending it with the following main new contributions:

- We extend and conduct an ablation study of TUEF across six scientific communities: StackOverflow, Unix, AskUbuntu, Server Fault, Physics, and Mathematics. This analysis allows us to understand the robustness of the model and the contribution of the various CQA information sources and TUEF components to the end-to-end predictive performance (RQ1).
- We integrate an interpretable Learning-to-Rank algorithm [60] in the TUEF framework. This integration demonstrates that TUEF can provide full transparency while maintaining competitive accuracy, highlighting interpretability as an additional strength of our approach (RQ2).
- We introduce a new categorization of state-of-the-art EF models based on the evaluation approach they adopt: *Expert Ranking* and *Expert Subsample Ranking*. This categorization systematizes existing approaches and highlights their implications, contributing to the definition of good practices for fair and consistent EF model evaluation (RQ3).
- Based on this categorization, we compare TUEF with state-of-the-art competitor models within the *Expert Ranking* category, to which TUEF belongs. Moreover, we adapt TUEF for a fair comparison with state-of-the-art models in the *Expert Subsample Ranking* category, accommodating this different evaluation configuration. We thus highlight the overall advantage of TUEF concerning relevant alternatives in the state of the art (RQ3).
- We study the ability of TUEF to scale with larger datasets by considering four datasets corresponding to 1, 2, 3, and 4 months of StackOverflow data (RQ4).

Our findings indicate that the MLG significantly enhances performance when the topics discussed in the community involve a well-defined clustering of question tags, as in StackOverflow. In the cases of small communities characterized by a less broad distribution of discussion topics, TUEF using single- or multi-layer graphs exhibit comparable performance. As discussed in [1], content information consistently emerges as the most crucial component for EF in all communities. Additionally, we show that integrating an interpretable Learning-to-Rank solution into TUEF typically results in a slight performance decrease. However, particularly in some communities, the reduction in prediction performance is minimal, making the interpretable version of TUEF a valuable option for gaining insights into the decision-making process. Finally, the results of extensive experiments conducted on CQA communities with varying size and characteristics show that TUEF is scalable and surpasses all baseline models, excelling in both the *Expert Ranking* and *Expert Subsample Ranking* evaluation categories.

The article is structured as follows: in Section 2, we present state-of-the-art models divided based on two different points of view: (i) the information used (Text-, Feature-, and Network-based methods) and (ii) the evaluation approach adopted (Expert Ranking and Expert Subsample Ranking). Next, Section 3 details the TUEF framework. Following, in Section 4, we present the experimental setup and detail the communities examined, the settings of TUEF hyperparameters, and the metrics used to assess its performance. Finally, in Section 5, we present the results of the TUEF ablation study across all communities and the comparison between TUEF and the state-of-the-art baselines under the two evaluation-based categories.

2 Related Work

In this section, we categorize state-of-the-art EF models from two perspectives: the type of information the models use (textual, features, and network) and the evaluation methodology the authors adopt for their assessment (with or without subsampling). Then, we survey some relevant work in the related area of EF for team formation, and discuss the positioning of CQA platforms in the era of LLMs.

2.1 Expert Finding Information-based classification

Research proposals in the field of the EF task for CQA platforms can be grouped into three broad groups based on the information they explore to address the task: text-based, feature-based, and network-based methods.

Text-based methods. A variety of methods have been proposed to address the EF task by leveraging similarities between current and previously answered questions. Dehghan and Ansell [15] introduced a model that clusters question terms based on semantic similarity and co-occurrence. Similarly, Liang et al. [54] proposed a semantic unsupervised generative adversarial network to assess similarities between word representations and experts. Moreover, Dehghan et al. [16] modeled user expertise by utilizing the tree structure of various domains, tags, and the temporal dimension of user response behavior. In a related vein, Zhang et al. [85] considered the temporal dimension by proposing models with multi-shift and multi-resolution settings to capture temporal dynamics effectively. Fu et al. [26] introduced a novel approach using the Recurrent Memory Reasoning Network (RMRN), which employs different reasoning memory cells equipped with attention mechanisms to focus on various aspects of the question. Building on the concept of attention mechanisms, Peng et al. [56, 64–66] implemented these mechanisms in multi-view, multi-grained, semi-supervised pre-trained, and pre-trained models with personalized fine-tuning for expert finding. Moreover, Qian et al. [68] proposed a Multi-Hop Interactive Attention-based Classification Network (MIACN) that utilizes attention mechanisms to detect latent interactions among question subjects and bodies.

Feature-based methods. The second group of methods in EF relies on hand-crafted features to model the expertise of community members. Roy et al. [71] formalized a scoring function to capture various aspects of expertise, including the propensity to answer questions related to profile tags, the ability to provide accepted answers, and recent activity levels. Mumtaz et al. [62] proposed a framework integrating activity, community, and time-aware features, with the temporal aspect also considered in [24] to track and incorporate the evolution of user roles, as well as in [45]. Similarly, Tondulkar et al. [84] included features that favor experts providing high-quality answers to complex questions, utilizing a comprehensive set of features capturing user availability and knowledge, and applying Learning-to-Rank (LtR) methods for expert ranking. The quality and consistency of answers are further addressed by Faisal et al. [21], who proposed a model based on an adaptation of the bibliometric g-index. LtR techniques are further utilized by Sorkhani et al. [77], introducing 74 content-based and social-based features. An important aspect in addressing the EF task is considering the intimacy between the asker and answerer, studied by Fu et al. [25]. As in text-based approaches, Tan et al. [83] employed attention mechanisms by proposing a method using Hierarchical Attentional Factorization Machines (HAFM). This approach combines factorization machines with hierarchical attention mechanisms to effectively model user-expert interactions and to highlight the importance of various features in expert recommendation tasks. The hierarchical attention mechanism helps in prioritizing relevant information at both the user and expert levels, enabling more accurate and personalized recommendations. Contrary to many studies focusing primarily on identifying the most senior platform users as experts, Roy et al. [70] aim to predict promising expert users at an early stage in community question answering sites.

Network-based methods. *Network-based methods* integrate interactions and relationship information within networks. Kundu et al. [44] defined a framework that comprises a text-based component for assessing expert knowledge on specific topics, accompanied by a Competition Based Expertise Network (CBEN) [3]. This network uses link analysis techniques, such as AuthorRank [58] and Weighted HITS [51]. The CBEN approach is further utilized in [46], where connections between two users are established if they have responded to the same question. Additionally, this study incorporates both *intra-profile* and *inter-profile* preferences of community users. In [48], the same authors propose a topic-sensitive hybrid expertise retrieval system (TSHER) that integrates assessments

of knowledge, reputation, and authority. Le et al. [50] measure the similarity between answerer and asker using social network techniques, employing Random Walks with Restart to calculate the proximity between two nodes.

Contrarily, Sun et al. [80] propose a language-agnostic perspective of user expertise, constructing a competition graph with user and question nodes where edges represent increasing levels of question difficulty, thus depicting the hierarchical structure of questions. This concept of hierarchy is further explored in [78]. Addressing data sparsity, Sang et al. [73] introduced a Multi-modal Multi-view Semantic Embedding (MMSE) framework that learns semantic embeddings from both local and global perspectives, incorporating social structure information to enhance embedding quality. Ghasemi et al. [27] focus on user embedding, proposing a joint model for text and node similarity.

Moreover, Li et al. [52] represent the CQA platform as a Heterogeneous Information Network (HIN), aiming to learn embeddings for nodes representing question contents, raisers, and answerers. They employ an LSTM-equipped Metapath-based Embedding algorithm and a CNN scoring function to rank experts. The use of HIN and metapath-based algorithms is also noted in [67]. Similarly, in [82], to tackle the cold question routing problem, the authors model users, questions, and tags as distinct node types in an undirected CQA graph built from historical interactions, and use a Graph Convolutional Network [35] to learn neighbor-aware embeddings that integrate both structural and textual information. Liu et al. [59] apply advanced semantic analysis and interest drift modeling to evaluate experts based on the relevance and depth of their contributions in specific domains. The concept of interest drift is similarly addressed by Krishna et al. [43], who propose a simple graph diffusion-based expert recommendation model that accounts for semantic and temporal information. Temporal dynamics in expertise assessment are further examined by Costa et al. in [10, 11], who introduce temporally-discounted, tag-based models.

2.2 Expert Finding Evaluation-based classification

The EF task is commonly cast to an item recommendation task in which the items are community users to be ranked by a measure of likelihood they can effectively answer the new question. The users with the highest scores are the ones to whom the new question should be routed. When evaluating recommender systems, there are two important methods to consider: online evaluation and offline evaluation. Online, end-to-end evaluation is the primary and most reliable way to evaluate a recommender system [34, 53], but it is not applicable during the development of a new model [13]. On the other hand, offline evaluation is the main instrument available for academic research and allows for exploring hyperparameter settings [8, 75]. It is commonly used to test a new model's effectiveness before moving to online evaluation. Moreover, in item recommendation tasks, the catalog of items to retrieve from is usually large, and finding matching items from this large pool is challenging [41]. Considering these factors, recent studies have speeded up the evaluation process by calculating the performance metrics of interest on a target set only. This target set is usually a small sample of the items possibly matching the query that includes all relevant items and a defined number of negative (non-relevant) items. However, in recent years, researchers have analyzed and critically evaluated these sample metric strategies. Krichene et al. [41] thoroughly investigated sampled metrics, revealing that they are inconsistent with their exact version. This finding is significant as it means that these sampled metrics do not persist relative statements, such as recommender A is better than B, not even in expectation. Furthermore, they found that the smaller the sampling size, the less difference between metrics. Canamares et al. [8] found that comparative evaluation using reduced target sets contradicts, in many cases, the corresponding outcome using large targets. Finally, Dallmann et al. [13] explored two widely used sampling strategies, *sampling by popularity* and *uniform random sampling*, and found that both can produce inconsistent rankings compared with the full ranking of the models and consistently produce different ranking when compared over different sample sizes.

When considering the EF context, one effective strategy to simplify the task and reduce complexity is identifying expert users based on specific criteria, such as their consistent provision of high-quality answers. However, it's crucial to acknowledge that this approach can intensify the cold-start problem, potentially leading to the exclusion of users who could be highly relevant to new questions. Despite this, the method significantly accelerates metric computation and enhances system accuracy. So far, different strategies have been adopted to define a subset of users representing the experts: 10% of the most active users [44, 48, 52, 55, 56, 64–67, 70, 83], all users with the number of accepted answers greater than a threshold [24–26, 43, 46, 47, 68, 79, 81], or a two-stage approach that considers the acceptance ratio [14, 22, 37, 62, 63]. No specific heuristic can determine whether a user should be considered an expert. However, to conduct a fair comparison among different models for the EF task, it is important to distinguish studies according to the sampling approach adopted during the ranking phase, ensuring that the models are compared under the same assumptions. Specifically, we recognize two different methods:

- **Experts Ranking**: For a new question, this group of works essentially ranks all users labeled as *expert users* in the first stage. The limitation of this approach during the evaluation phase is that it requires the test set to consist of questions for which the best answerer is a user who was previously labeled as an expert. On the other hand, this evaluation methodology provides realistic figures of the end-to-end performance of the observed system. The following works belong to this group: [10, 11, 22, 24–26, 43, 44, 46–48, 59, 62, 63, 68, 70, 79, 81].
- **Experts Subsample Ranking**: Given the new questions along with the information of all the users who actually answered the question (ground truth), the studies in this group rank only a fixed-size subsample of users. This set of users is generally not large and always includes the ground truth users plus some other users who didn't answer the question (negative examples). In most of the works following this approach, the size of the subsample is 20 and negative examples chosen randomly among the 10% most active users are added to the subsample until it reaches the desired cardinality. While this approach overcomes one limitation of the *Experts Ranking* methodology by requiring only the knowledge of the answers of the questions in the training set, it relies on a sampling strategy that might produce inconsistent rankings if the sample size is varied and can result in unreliable system rankings[8, 13, 41]. The works belonging to this group include the following: [27, 52, 55, 56, 64–67, 73, 77, 83, 85].

Although both approaches are used in the literature, this distinction is important, as they may lead to different conclusions regarding model effectiveness. Explicitly clarifying the chosen evaluation strategy enables more accurate interpretation of results and ensures fairer comparisons across EF models.

2.3 Team Formation

A related line of research to collaborative expert finding is the Team Formation (TF) problem, which focuses on identifying groups of experts from collaboration networks using graph-based search techniques [20]. In these methods, experts are modeled as nodes and their past collaborations as edges, and the objective is to find a compact subgraph whose members collectively possess all the skills required for a given task while optimizing certain criteria such as communication cost or expert cohesion [20, 31, 40]. One of the earliest formulations considers the presence of a social network and aims to minimize the communication cost among selected members, measured through metrics such as the diameter or the cost of the minimum spanning tree [49]. Later approaches extended this formulation by incorporating Social Network Analysis features, accounting for interpersonal ties, communication frequency, shared project history, and friendship attributes to better capture team compatibility [31]. Building on this idea, Experts2team [40] combines task relevance with global cohesion and local diversity to ensure both efficiency and accuracy in team assembly. Other learning-based approaches employ neural architectures to capture collaboration patterns: for instance, [31] proposes a variational Bayesian network that learns team representations from historical co-work data and adopt a mapping function learning strategy to map

between expert and skill spaces, while [74] models skill transfer between teammates over time and uses a deep neural autoencoder to recommend new collaborators. Complementary studies have explored explainability and interpretability in expert and team search systems, such as ExES [29], which adapts saliency and counterfactual methods to explain why certain experts or teams are retrieved. From an optimization perspective, [36] and [39] address related problems of compact group discovery, proposing approximation algorithms to identify dense subgraphs with strong skill coverage under size constraints. Overall, the TF problem shares strong conceptual and methodological ties with EF. While TF extends the search to groups rather than individuals, it fundamentally relies on accurate expertise estimation and relevance ranking, core functions of EF systems. Thus, EF models form a crucial foundation for TF, enabling the identification of suitable individuals whose skills and social connections can be effectively combined into cohesive, high-performing teams.

2.4 Expert Finding in the Era of Large Language Models

The introduction of large language models such as ChatGPT has already changed the dynamics of community-based question answering platforms. Several studies show that their release was followed by sharp declines in traffic and posting activity on Stack Overflow, in some cases by about 25% within a few months [7, 17]. This reduction is not uniform: domains with rich historical data, often used to train LLMs, show larger decreases, while areas with less training data still rely more on human contributions [7, 12, 17, 32]. These trends raise concerns about the sustainability of CQA platforms, since lower participation reduces both the support available to developers and the fresh content needed to train future LLMs. This situation has been described as the "paradox of re-use," where models risk displacing the very behavior that generated their training data [17].

Beyond participation, LLMs also raise issues of reliability. While they provide quick and often relevant answers, they are prone to hallucinations (confident but incorrect outputs) and can sometimes generate misleading or harmful content [12, 18, 30]. Over-reliance on such systems can erode trust and weaken the collaborative exchange of ideas that makes online communities valuable.

At the same time, LLMs can bring some benefits. They may reduce low-quality or poorly formulated questions by helping users write clearer queries, and they can free up human experts to focus on more complex and nuanced issues [72]. In this sense, LLMs can complement CQA platforms rather than replace them.

For these reasons, EF models remain essential. LLMs alone cannot guarantee accuracy, depth, or domain-specific reliability, in particular for evolving and dynamic knowledge fields. EF models can ensure that complex and high-value questions are directed to the most qualified individuals, supporting both knowledge quality and community engagement. In doing so, they help maintain the long-term sustainability of CQA platforms and preserve the digital public goods on which both communities and LLMs depend.

Position of our work. Previous approaches to expert finding typically combine content and social signals at a global level or rely on data-intensive neural models that lack transparency and scalability. TUEF advances the state of the art by introducing a multi-layer graph representation of CQA platforms, where expertise is modeled through knowledge vectors and topic-specific similarities. This allows efficient and effective expert selection within the correct topical context, ensuring that influence and relevance are evaluated appropriately. In addition, TUEF integrates both a standard and an interpretable learning-to-rank variant, offering a balance between performance, transparency, and efficiency.

3 Topic-oriented User-interaction model for Expert Finding (TUEF)

The TUEF framework leverages topic-oriented content similarity and user relationships to facilitate the selection and ranking of experts. This approach is based on two fundamental considerations:

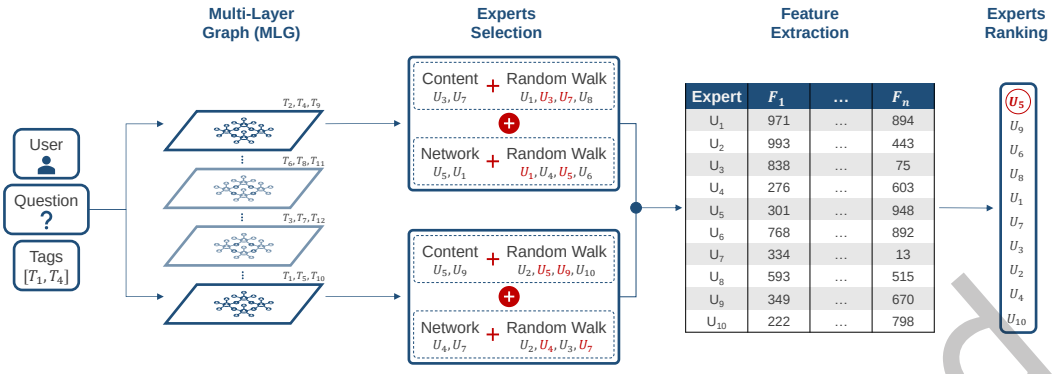

Fig. 1. Illustration of the TUEF approach highlighting the distinct components. At inference time, TUEF first determines the main topics to which the question q belongs and the corresponding graph layers (Multi-Layer Graph). Next, for each layer, it selects the candidate experts from two perspectives: i) *Network*, by identifying central users that may have considerable influence within the community; ii) *Content*, by identifying users who previously answered questions similar to q. The Multi-Layer Graph is used to collect candidate experts through Random Walks (Expert Selection). Following, TUEF extracts features based on text, tags, and graph relationships for each selected expert (Feature Extraction). Finally, TUEF uses a learned, precision-oriented model to score the candidates and rank them by expected relevance (Experts Ranking).

- CQA platform users utilize tags to characterize their questions and increase the likelihood of receiving relevant answers. The questions' tags help identify topical areas within the community relevant to specific questions.
- Unlike traditional social networks, user relationships on CQA platforms are not explicitly defined. However, implicit relationships can be discerned by analyzing users' question-answering behaviors. The knowledge gained from these interactions can enhance answer quality and user engagement with the platform.

Figure 1 illustrates the logical organization of TUEF. To effectively integrate the valuable information derived from tags, content, and user relationships, TUEF adopts a MLG representation (Section 3.1) that encapsulates the macro topics discussed within the online community. Each layer of this graph models user relationships at the level of specific macro topics, considering their similarities based on tags, questions, and answering behaviors. TUEF identifies community *experts*—users known for consistently providing accepted answers—and implements an exploratory algorithm that fully exploits the MLG structure to select a group of candidate experts for each relevant macro topic associated with the current question (Section 3.2).

This exploratory algorithm integrates social and content-based information, considering experts' centrality within the network and their expertise in the topics of interest. Central users often represent individuals who have garnered significant attention and engagement within the community. Similarly, users who have previously answered historical questions similar to the query are more likely to connect with experts relevant to the specific topic of interest. By strategically exploring the MLG starting from these nodes, the algorithm increases the probability of selecting appropriate experts early in the process.

Finally, TUEF extracts features that represent the identified candidate experts and applies Learning-to-Rank techniques [57] (Section 3.3) to rank them based on their expertise and likelihood of effectively answering the question. This systematic approach ensures the selection of high-quality experts tailored to the specific information needs of the user asking the question.

3.1 User Interaction Model

In TUEF, user relationships related to each specific topic are independently modeled. From here on, let U denote the set of active users on the CQA platform under consideration, and let Q represent the set of past questions posted and answered by users in U.

Topic Identification. TUEF employs a tag-clustering approach to identify the main topics discussed on the CQA platform using historical questions. The clustering approach, discussed in [33], utilizes the k-means algorithm [61] on a tag co-occurrence matrix M. Tags' co-occurrence patterns are leveraged to group semantically related tags, as tags often associated with the same questions tend to be conceptually similar [28]. For each question $q \in Q$, the tags associated with q are denoted as $tags(q)$. Let T be the set of all unique tags across questions in Q, and let F represent the set of the most frequent tags (top λ) that serve as clustering features. The co-occurrence matrix $M^{|T|\times|F|}$ is constructed as follows:

$$m_{i,j} = |\{q \in Q \mid \{t_i, f_j\} \subseteq tags(q),\ t_i \in T,\ f_j \in F\}| \tag{1}$$

Here, $m_{i,j}$ indicates the number of questions in Q where the i_{th} tag and the j_{th} feature co-occur. Finally, each row of the matrix is normalized to represent the relative frequency of tag co-occurrences with each feature, ensuring that the values reflect the proportion of times each tag appears alongside the corresponding feature. For a given tag t_i and feature f_j, the normalized matrix $\hat{M}$ is defined as:

$$\hat{m}_{i,j} = \frac{m_{i,j}}{\sum_{n=1}^{|F|} m_{i,n}} \tag{2}$$

Clustering the tag co-occurrence matrix to devise macro topics is based on the observation that CQA users often pair broad, common tags with more specific ones to achieve greater precision in topic identification. By considering the co-occurrences of specific tags with broader tags, we can enhance the categorization of questions. Using the normalized matrix $\hat{M}$ and a specified value of k for the k-means algorithm, the clustering process yields k disjoint clusters of tags representing primary community domains. The optimal k value is determined based on silhouette maximization criteria [69], as described in Section 4. Compared to clustering alternatives, k-means offers a robust, interpretable, and computationally efficient solution, making it particularly suitable for large-scale CQA datasets. This comprehensive approach to topic identification and clustering ensures that TUEF effectively captures and represents the main topical areas discussed within the online community, facilitating a good understanding of community dynamics and interactions.

Multi-Layer Graph. TUEF carefully models user relationships within each layer by treating users U as nodes in a MLG and establishing connections based on similar patterns of providing accepted answers within specific layers. Formally, the structure of TUEF is defined using a MLG $G = [L_1, ..., L_k]$, where each layer L_i corresponds to a tag cluster. Each layer $L_i = (V_i, E_i)$ is an independent graph, with nodes V_i representing users associated with the layer and edges E_i denoting their relationships. To ensure consistent and accurate answers within a specific layer, TUEF includes in V_i only those users who have provided a number of accepted answers equal to or exceeding a given ϵ percentile. Moreover, to characterize users' knowledge in V_i, each user $u \in V_i$ is assigned a topic vector b_u^i. This vector represents the user's contribution to various tags within the layer, normalized by their total contribution across all layers. By normalizing, TUEF accounts for differences in users' overall activity levels, allowing for a fairer comparison of their topical knowledge. Specifically, for a tag t_j within L_i, the j_{th} position of b_u^i is calculated as:

$$b_u^i[j] = \frac{accepted(L_i, u, t_j)}{\sum_{\forall z,m} accepted(L_z, u, t_m)} \tag{3}$$

Here, $accepted(L_i, u, t_j)$ denotes the number of accepted answers provided by user u for questions labeled with tag t_j in layer L_i. Finally, after computing the topic vectors for all the users in V_i, the cosine similarity is calculated between each pair of users within the layer. If the similarity between two users u_a and u_b exceeds a predefined threshold δ, an edge (u_a, u_b) weighted by the cosine similarity value is added to E_i.

It is crucial to emphasize that each question in TUEF is associated with tags assigned to specific layers within G. Consequently, a question can belong to multiple layers, and users answering these questions are represented across all associated layers. This multidimensional representation of user expertise and relationships across layers enables TUEF to capture a comprehensive view of users' knowledge and interactions within the CQA platform. This approach ensures that users' expertise and social connections transcend individual layers, reflecting the nature of community participation and expertise within the platform.

3.2 Expert Selection

The *Expert Selection* component has the goal of selecting for each new question q submitted to the CQA platform a set of users who are likely to be highly knowledgeable of the topics of q, i.e., the set of candidate experts. The component operates iteratively for each layer to which question q belongs. This selection process is structured into three distinct phases: (i) *Sorting phase*, where the nodes within each layer are arranged in a specified order to facilitate subsequent expert selection; (ii) *Collection phase*, which selects an initial set of candidate experts from the sorted node lists created in the previous step; (iii) *Exploratory phase*, that expands the initial set resulted from the Collection phase by exploring the graph structure to identify additional potential experts. The Expert Selection process is conducted independently by the Network-based and Content-based methods, each following steps (i)–(iii) with distinct node selection criteria, as detailed below. This process remains query-dependent, operating solely within the layers relevant to the query's topics.

The remainder of this section investigates the Expert Identification process, which classifies users $u \in U$ as either *experts* or *non-experts*, followed by a detailed exploration of the three phases mentioned above.

Expert Identification. Experts in CQA platforms can be identified heuristically by considering the number of *accepted answers* they provided in the past. Another signal of user trust is given by the acceptance ratio r_u, i.e., the ratio between the number of accepted answers and the total number of answers a given user provides. As in [14, 37, 63], in TUEF, we adopt an expert selection criterion based precisely on these two measures: first, we select the set $C \subseteq U$ of *candidate experts* by considering all the users having a number of accepted answers greater or equal to a specified threshold β: next, we label as *experts* the set $E \subseteq C$ of users whose acceptance ratio r_u is greater than the overall average $\bar{r}$:

$$E = \{u \in C \mid r_u > \bar{r}\}, \quad \bar{r} = \frac{\sum_{u \in C} r_u}{|C|} \tag{4}$$

Each layer within the MLG comprises nodes labeled as *experts* and *non-experts*. While the former are the primary targets of the Expert Selection process, the latter facilitates comprehensive exploration of the graph structure.

By implementing this process, we aim to identify users who consistently provide valuable and accepted solutions within the community, as they are likely to be proficient candidates for effectively addressing new questions posted on the platform.

Sorting. TUEF adopts a comprehensive approach to identify candidate experts for a given question q by leveraging content and social information. This process involves two key perspectives: the *Network-based* perspective, focusing on users' centrality within the network, and the *Content-based* perspective, assessing users' relevance to the newly posted query based on their past interactions. In the Network-based approach, TUEF utilizes Betweenness centrality [23], which quantifies the centrality of nodes within a graph by evaluating their influence over information flow. Nodes with higher Betweenness centrality are considered more central within

the network. Each node v_j^i in layer L_i is assigned a Betweenness score s_j^i, which is used to sort nodes in descending order. On the other hand, the Content-based approach involves sorting layer nodes according to their similarity to the query q based on questions previously answered by experts. This is achieved using Information Retrieval techniques and pre-built indexes, specifically the *TextIndex* and *TagIndex*, which index the text and associated tags of historical questions, respectively. When a new question is presented with its tags, the Content-based method employs a retrieval model (we specifically use BM25) independently on both indexes to retrieve a sorted list of relevant questions, along with information about the experts who provided accepted answers. The query for the *TextIndex* includes the concatenated question, title, and body, while for the *TagIndex* it consists of the concatenated question tags.

Subsequently, the ranked lists generated from the TextIndex and TagIndex are merged using a ranking fusion technique to ensure a balanced integration of content relevance and tag-based similarity. Specifically, each user u is assigned a ranking score computed as:

$$\text{score}(u) = \frac{1}{\min(\text{rank}_{\text{TagIndex}}(u), \text{rank}_{\text{TextIndex}}(u))}$$

where $\text{rank}_{\text{TagIndex}}(u)$ and $\text{rank}_{\text{TextIndex}}(u)$ represent the user's ranking positions in the TagIndex and TextIndex lists, respectively. Finally, users are sorted in descending order based on this score, ensuring that those with the highest relevance in at least one ranking method are prioritized.

This approach results in a query-specific ordering of nodes within each layer L_i: while the Content-based method dynamically ranks users based on their relevance to the query, the Network-based method relies on a precomputed static ordering derived from users' centrality within the selected layers. Nevertheless, since only the layers relevant to the query are considered, the Network-based ranking is also query-dependent.

Candidate Collection. In the candidate collection phase, the objective is to select a subset of experts $D_i \subseteq V_i$ from each layer L_i by sequentially scanning the sorted lists generated in the previous step. This process balances the need for a sufficiently large candidate pool to ensure high recall (the probability of obtaining the correct answer) with the desire to include relevant experts for high precision. To achieve a balance between precision and recall, we estimate the probability p of not receiving an answer from any user in D_i. Initially, p is set to 1 when D_i is empty and is incrementally reduced as experts are added to D_i. Each expert u added to D_i contributes to lowering p based on their acceptance ratio μ_u, defined as the ratio of accepted answers to total answers provided by the user. Furthermore, to refine the modeling of topic-based expertise, we smooth μ_u by considering the user's activity within the specific layer: the smoothing factor is computed by adjusting μ_u based on the ratio of expert answers within the layer to the maximum number of answers provided by any user within that layer. The probability p is updated iteratively using the formula:

$$p = p \cdot (1 - \mu_u) \tag{5}$$

This iterative process continues until p becomes less than or equal to a predefined threshold α. Once the threshold α is met, the inclusion of new candidates to D_i is halted, and the exploratory phase of the expert selection process begins. This strategic approach ensures that the candidate pool D_i comprises experts likely to provide valuable and accurate answers to the specific question while representing good starting points for MLG exploration.

The Candidate Collection phase processes the sorted lists from the Content-based and Network-based methods independently for each of the query's relevant layers.

Exploratory Phase. In the Exploratory phase, TUEF explores the graph structure to identify additional candidate experts that may not have been identified with the previously described phases, specifically exploiting the implicit relationships between users, aiming to enhance recall. The starting point for this exploration is the set of experts D_i from each layer L_i. Specifically, for each node $v_i \in D_i$, TUEF initiates a series of fixed-length Random

Walks. At each step of the Random Walk, the next node to visit is selected randomly based on a probability distribution d, computed considering the neighbors of the current node v_i and their link weights, representing the similarities with neighboring nodes.

More formally, let $\mathcal{N}i = v_{i1}, \ldots, v_{iN_i}$ denote the neighbors of expert node v_i, where $N_i = |\mathcal{N}_i|$ is the total number of neighbors of v_i. The probability d_j of visiting a neighbor node v_{ij} is calculated as:

$$d_j = \frac{w_{ij}}{\sum_{z=1}^{|N_i|} w_{iz}} \tag{6}$$

Here, w_{ij} represents the weight of the edge between node v_i and its neighbor v_{ij}. Nodes with stronger connections to the current node v_i are more likely to be selected during the Random Walks. Whenever an expert node is encountered during the exploration, it is added to the current set of candidate experts if not already included.

This exploratory process is carried out independently for each layer to which the question belongs, ensuring a comprehensive exploration of both Network-based and Content-based perspectives within each layer. Combining Random Walks with tailored probability distributions enriches the candidate pool D_i with potentially neglected experts, enhancing the overall recall and effectiveness of the expert selection process in the TUEF framework.

3.3 Ranking candidate experts

TUEF leverages *Learning to Rank* (LtR) algorithms [57] to learn an effective ranking function from training data. LtR methods exploit a labeled dataset to learn a scoring function σ that approximates the ideal ranking function inherent in the training examples. TUEF selects a subset of past questions, used to model user relationships as discussed in Section 3.1, to build the training set $T \subset Q$ for the LtR algorithm. Each query q in the training set T is associated with a set of candidate experts $CE \subset U$. The CE set consists of all experts identified during the Exploratory phase, incorporating candidates selected through both the Network-based and Content-based methods.

Moreover, for each query-candidate pair (q, u_i), where $q \in T$ and $u_i \in CE$, there exists a *binary relevance judgment* l_i, where only the expert who provided the accepted answer (ground truth) is assigned a relevance score of 1, while all other candidate experts receive a score of 0. Each query-candidate pair (q, u_i) is represented by a feature vector x that encapsulates information about the query, the candidate expert, and their relationship. The LtR algorithm learns a function $\sigma(x)$ that predicts a relevance score for the input feature vector x. This learned function $\sigma(x)$ is subsequently used during inference to compute scores for candidate experts and rank them accordingly.

The features used to model the query and candidate expert can be categorized into two groups: *Static* features and *Query-dependent* features. The group of Static features comprises those that remain the same for each query: the *Reputation* of the expert, the number of *Answers* and *AcceptedAnswers*, the *Ratio* (i.e., the ratio of answers to accepted answers), and *AvgActivity* and *StdActivity*, representing the average and standard deviation of time intervals between consecutive answers provided by the expert. The Query-dependent features, instead, are computed every time for each query. Some of these features are inherently static within a layer—such as BetweennessScore, which remains constant for a given user in a specific layer. However, they are still considered query-dependent because their values are selected based on the layers relevant to the query's topics. These features include:

- *LayerCount*: Number of distinct graph layers in which the expert is selected during the Expert Selection process.
- *QueryKnowledge*: Ratio of answers to accepted answers provided by the expert in layers relevant to the query q.

- *VisitCountContent* and *VisitCountNetwork*: Total number of times an expert is encountered in the Collection and Exploratory phases using Content-based and Network-based techniques.
- *StepsContent* and *StepsNetwork*: Number of steps required to discover the expert during Collection or Exploratory phases.
- *BetweennessPos* and *BetweennessScore*: Expert's rank in the list of users ordered by Betweenness score and the Betweenness score itself.
- *ScoreIndexTag* and *ScoreIndexText*: Sum of BM25 scores of historical questions answered by the expert in IndexTag and IndexText, respectively.
- *FrequencyIndexTag* and *FrequencyIndexText*: Number of distinct questions answered by the expert returned by the respective indexes.
- *Eigenvector*, *PageRank*, *Closeness*, *Degree*, *AvgWeights*: Centrality measures and network features computed based on the graph layers relevant to the query.

Specific rules are applied to aggregate feature values for candidate experts selected in multiple layers. Network-based features (e.g., *BetweennessScore*, *Closeness*, *PageRank*) consider maximum values, while Content-based features (e.g., *FrequencyIndex*, *ScoreIndex*, *QueryKnowledge*) utilize summation. Additionally, certain features like *StepsContent*, *StepsNetwork*, and *BetweennessPos* use minimum values to characterize the candidate experts effectively within the ranking framework. These features collectively contribute to the comprehensive ranking process in TUEF, accommodating both static and query-specific attributes of candidate experts for optimal ranking performance.

4 Experimental Setup

The following section presents the experimental setup and details the datasets used, the settings of the TUEF hyperparameters, and the metrics used to assess performance.

4.1 Datasets

We selected six real-world datasets from the Stack Exchange[1] platform. While these datasets originate from the same ecosystem and share a common structure, including voting mechanisms, reputation points, and moderation tools, they capture interactions from different communities with distinct linguistic styles, domain-specific terminologies, and user behaviors. Each Stack Exchange site is dedicated to a specific topic, shaping the discussions and the way knowledge is shared within that domain. However, the unified framework of the platform ensures that all datasets have the same features, enabling a fair comparison of our method across diverse community settings. This allows us to assess performance differences that arise from variations in language, domain and Q&A habits while maintaining consistency in the underlying data structure.

Users on these communities post questions by specifying a title, a detailed body, and a set of tags chosen from a predefined list, which categorize the question's topic. By using tags, community members can quickly find open questions where they can offer help in solving problems. As technology evolves rapidly, the distribution of tag usage follows a power-law distribution (c.f. Figure 2 in [9]): a few tags are used frequently, representing the major topics discussed on the platform, while the majority are used infrequently, typically reflecting more specific or niche problems users seek to solve. Finally, the community can upvote or downvote questions and answers, and importantly, only one answer can be marked as the *accepted answer* by the user who posted the question, thus identifying the *best solution* for the asker and, consequently, providing the human-assessed ground truth for the EF task.

The dataset for each community includes all the posted questions and answers. Questions can be categorized as *closed*, indicating the presence of an *accepted answer*, or *open*, which may have no answers. Information about the

[1] https://stackexchange.com/

user who posted a question or an answer is available, such as Reputation score, number of Views, UpVotes, and DownVotes. As users receive more upvotes and have their answers accepted, their Reputation score increases, reflecting their expertise and ability to provide high-quality solutions.

These datasets are publicly accessible[2] and encompass all the communities' activity, covering all content from the creation of each site to the present. The selected communities are:

- **Stack Overflow (SO)**: A platform for technical questions about programming languages, tools, frameworks, and coding challenges.
- **Unix (UX)**: A hub for sharing knowledge and solutions related to Unix/Linux commands, scripting, system management, and tools.
- **Ask Ubuntu (AU)**: Dedicated to troubleshooting and resolving issues related to the Ubuntu operating system, including installation and configuration.
- **Server Fault (SF)**: A community for IT professionals to address server-related issues, such as hardware, networking, and performance optimization.
- **Physics (PH)**: A forum for exploring physics principles, experimental methods, and theoretical frameworks in both classical and modern contexts.
- **Mathematics (MAT)**: A space for discussing topics ranging from algebra and calculus to number theory, geometry, and statistics.

Table 1. Number of questions, answers, users, askers, answerers (as a percentage of users), and unique tags for each community.

Community	#Questions	#Answers	#Users	#Askers	#Answerers	#Tags
Stack Overflow	24,081,764	35,553,848	6,269,699	4,954,743	3,095,426 (49.37%)	65,611
Unix	233,571	344,119	141,308	99,771	66,785 (47.26%)	2,810
Ask Ubuntu	409,621	517,032	309,695	218,320	145,847 (47.09%)	3,152
Server Fault	322,227	509,853	190,939	134,113	98,632 (51.66%)	3,906
Physics	226,159	330,331	89,826	70,814	32,558 (36.25%)	899
Mathematics	1,600,808	2,103,040	357,593	317,054	96,382 (26.95%)	1,977

Table 1 provides an overview of the characteristics of six communities, detailing the volume of questions, answers, users, and tags, as well as the distribution of participation between askers and answerers. Stack Overflow exhibits the largest scale in terms of activity, with over 24 million questions and 35 million answers, supported by 6.3 million users. This community also has the highest number of tags (65,611), highlighting its extensive thematic diversity. Approximately 49% of users on Stack Overflow contribute as answerers, indicating strong engagement from nearly half the user base. A similar proportion of answerers is observed in Unix, Ask Ubuntu, and Server Fault, with 47.26%, 47.09%, and 51.66%, respectively. In contrast, Physics and Mathematics report lower levels of participation. Although Mathematics has the second-highest number of users (357,593), only 26.95% contribute as answerers, suggesting a significant portion of the community primarily engages in asking questions or passive participation. Overall, the other five communities are considerably smaller compared to Stack Overflow in terms of scale and activity. Additionally, these smaller communities cover fewer topics, as evidenced by the significantly lower number of tags.

[2]https://archive.org/details/stackexchange

We statistically analyze the differences between these communities using the *Kolmogorov-Smirnov* (KS) test [2, 76], a non-parametric statistical test employed to determine whether two samples are drawn from the same underlying distribution. The *null hypothesis* of the KS test assumes that the two samples come from the same underlying distribution. The test returns a *p-value*, which quantifies the probability of observing the given data under the assumption that the null hypothesis is true. In this study, we set the significance level at $\alpha = 0.01$, ensuring that the probability of incorrectly rejecting the null hypothesis is limited to 1%. This conservative choice of α enhances the robustness of our findings by reducing the likelihood of false positives. If the p-value satisfies the condition p-value $< \alpha$, the test rejects the null hypothesis, indicating statistically significant differences between the two distributions. Moreover, we apply the Bonferroni correction [19], a method used when performing multiple dependent or independent statistical tests simultaneously. To reduce the likelihood of identifying false positives, the significance level α is adjusted by dividing it by the total number of comparisons n.

We evaluate the differences under two different groups:

- **Questions**: compares the question characteristics across the six communities, highlighting differences in content (BodyChar, TitleChar, Tags), engagement (Answers, Comments, Score), and resolution rates (%Solved). Details about the mean of these features across all communities are reported in the Appendix. Stack Overflow has the highest average of questions length (1,587.99 characters) but a moderate resolution rate (49.16%), while Ask Ubuntu achieves the highest resolution rate (67.13%) with shorter questions and high scores (3.30). Physics has the highest average tags (3.28) and a strong resolution rate (57.23%). Unix and Server Fault exhibit similar question lengths and moderate resolution rates (52.5%), while Mathematics has the lowest resolution rate (47.68%) and shortest questions (953.87 characters). The KS test confirms statistical differences in all question features across communities. Additionally, Figure 2 (a) illustrates tag overlaps between communities, which are generally low. The highest overlap is observed between Ask Ubuntu and Unix (26.42%), while most overlaps are below 15%, indicating limited thematic similarity between communities.
- **Users**: compares the user features across communities, including Reputation, Views, UpVotes, DownVotes, Answers, and Accepted answers. Details about the mean of these features across all communities are reported in the Appendix. Mathematics has the highest mean values across all features, indicating high contributions and visibility. Stack Overflow also demonstrates strong activity, while Physics and Unix exhibit moderate engagement. In contrast, Server Fault and Ask Ubuntu show the lowest interaction levels. These patterns reflect unique participation dynamics across communities. Figure 2 (b) shows user overlaps, with the highest observed between "Unix-Stack Overflow" (9.40%), followed by "Ask Ubuntu-Stack Overflow" (7.77%). The KS test rejects the null hypothesis for all pairs of communities across all user features, except for "Server Fault-Unix" and "Physics-Stack Overflow" in the DownVotes feature. These results suggest minimal user-sharing interactions between communities, underscoring their distinctiveness.

Overall, the analysis confirms that all communities exhibit distinct characteristics across both question and user features, reflecting unique engagement patterns, thematic focuses, and participation dynamics. These differences provide a diverse and challenging evaluation setting, allowing us to assess the robustness of our method across varying linguistic styles, user behaviors, and content structures.

4.2 Evaluation metrics

We use Precision@1 (P@1), Normalized Discounted Cumulative Gain @3 (NDCG@3), Mean Reciprocal Rank (MRR), and Recall@5 (R@5) as our evaluation metrics. The cutoffs considered are short, as finding the relevant experts at the top of the ranked lists for the EF task is essential. P@1 assumes a pivotal role in the EF task, emphasizing the primary objective of identifying the most suitable answerer for a given query. NDCG@3 and

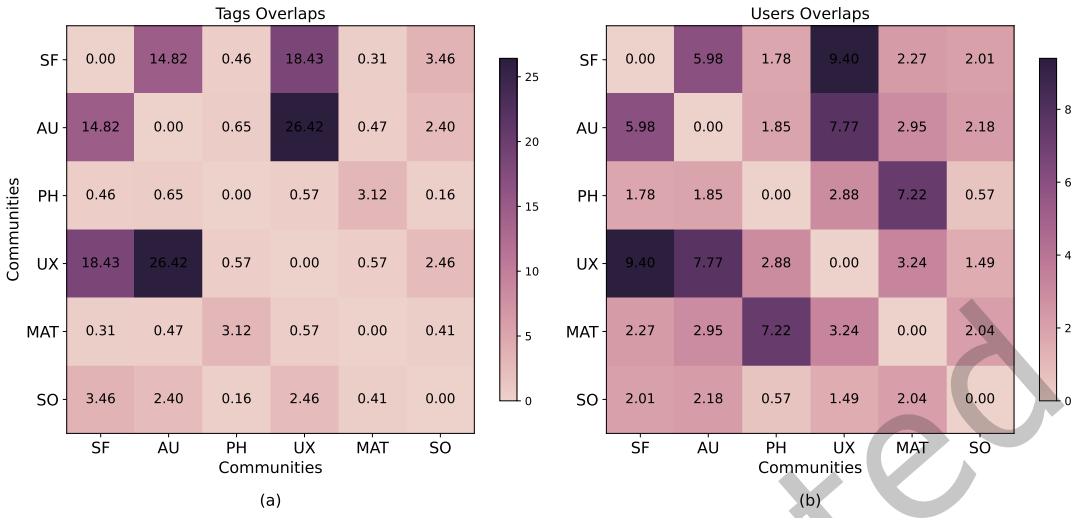

Fig. 2. (a) Heatmap showing the percentage of shared tags between communities. (b) Heatmap showing the percentage of shared users between communities. The values indicate the extent of overlap, with darker shades representing higher percentages.

MRR metrics serve as valuable tools when models exhibit identical P@1 scores, offering a detailed comparison by considering the actual position of the best answerer in the list. R@5, in turn, measures the model's efficacy in locating the best answerer within the top five positions. Moreover, we perform a paired t-test with a p-value<0.05 and apply Bonferroni correction. In the tables, we indicate statistically significant performance gains and losses relative to TUEF using the symbols and , respectively. Performance metrics and statistical significance tests are computed using the RanX Library [4].

Reproducibility. TUEF is prototyped in Python 3.8.17. All experiments are conducted on an Intel(R) Xeon(R) Platinum 8164 CPU 2.00GHz processor with 503GB RAM on Linux 5.4.0-153-generic. The source code of TUEF and the data used are publicly available[3].

4.3 Parameters setting

Here, we outline the pre-processing steps, as well as the parameter settings required by TUEF. The parameters, their descriptions, and the chosen values are summarized in Table 2. The values chosen for the experiments have been tuned using as the validation set the last 20% of questions from the Stack Overflow training set. To ensure a fair comparison, we varied one parameter at a time while keeping all others constant, isolating its impact on model performance. Further details are provided in the Appendix.

Pre-processing. We pre-processed the data to ensure a good set of questions and answers. We applied established data cleaning procedures, as outlined in [62], to ensure the quality of our dataset. We opted for an eight-year timespan for most communities, excluding StackOverflow and Mathematics. The decision to use a shorter timeframe for these two communities arises from their high daily question volumes, ensuring a more balanced representation across all communities. This strategy aims to maintain a roughly equivalent dataset size for each community. We removed questions and answers without a specified question's *ID* and *OwnerUserID*

[3]https://github.com/maddalena-amendola/TUEF

Table 2. Description of the main parameters used by TUEF

Parameter	Description	Value
λ	Number of most frequent tags used as clustering features	10
K	Number of layers in the MLG	$[2, 10]$
ϵ	Minimum number of accepted answers required for a user to be included as a node in the MLG	3
δ	Minimum similarity threshold for creating an edge between two users in the MLG	0.5
ω	Percentile threshold to determine the minimum number of accepted answers for a user to be considered an expert	95^{th}
p	Probability threshold for selecting initial candidate experts in the Candidate Collection phase	0.001
restarts	Number of Random Walks performed from each expert node selected in the Candidate Collection phase	5
steps	Number of maximum steps of Random Walks	10
ζ	Number of training queries for the LtR model	-

(i.e., the ID of the user who posted the question). Subsequently, we retained questions with an *AcceptedAnswerID* (i.e., closed questions) and answers with a valid *ParentID* (i.e., the respective *question ID*). Questions where the asker and the best answerer were the same user were also excluded. We split the dataset, allocating 80% for training/validation and the remaining 20% for the test set. Notably, we maintained the chronological order of the questions to preserve temporal integrity. Comprehensive statistics for the resulting datasets are provided in Table 3, i.e., the *Time Period* considered, the number of questions for *Train* and *Test*, and the number of *Answerers* with at least one accepted answer.

User Interaction Model. To identify the primary topics discussed in the community, we employed the clustering technique outlined in Section 3.1, focusing on the tags associated with questions in the training set. Considering the top $\lambda = 10$ most frequent tags as features, we determined the number of clusters within the range $K = [2, 10]$ that maximizes the Silhouette score. Each cluster represents a specific macro-area, corresponding to a layer in the MLG, which captures interactions among users who have provided a number of accepted answers equal to or exceeding $\epsilon = 3$. Users are represented by their topic vectors used to compute the pair-wise cosine similarities (see Section 3.1). When modeling relationships, we retained edges with a similarity equal to or greater than $\delta = 0.5$. Table 3 reports the number of *Tags* and *Clusters* (i.e., MLG layers), the *Silhouette* score, and the average number of nodes and edges (*AvgNodes* and *AvgEdges*) in the communities' MLG.

Expert Selection. We identify as experts the users with a number of accepted answers greater than or equal to the $\omega = 95th$ percentile of the distribution, following the procedure detailed in Section 3.2. The minimum number of accepted answers and the total number of users labeled as experts for each dataset are outlined in Table 3 under the rows *MinAccAns* and *Experts*, respectively. TagIndex and TextIndex return the 1,000 most similar past questions for each query in the expert selection process, respectively. For both Network and Content methods, after node sorting, the collection phase identifies the initial set of experts D that reaches a probability threshold

of $p = 0.001$, representing the probability of not receiving an answer. Subsequently, starting for each selected expert in D, we conduct 5 Random Walks of, at most, 10 steps.

Ranking. We utilize the LightGBM[4] [38] implementation of LambdaMART [6] to learn the TUEF ranking model. The TUEF explainable ranking model is instead trained using IlMart [60], a learning algorithm based on LambdaMART that generates interpretable models using only univariate and bivariate functions at training time. The IlMart learning strategy is made of three steps: i) *Main Effects Learning*, which learns with the LambdaMART boosting strategy an ensemble of trees $\tau(x_j)$ in which the only feature allowed for each split is x_j; ii) *Interaction Effects Selection*, which selects the top-K most important feature pairs (x_i, x_j); iii) *Interaction Effects Learning*, which learns with LambdaMART an ensemble of trees $\tau_{ij}(x_i, x_j)$, where the only features allowed in a split of each tree $t \in \tau_{ij}(x_i, x_j)$ are either x_i or x_j, i.e., one of the features pairs selected by the previous step. The resulting ranking models can achieve ranking performance close to unconstrained LambdaMART models without all their complexity, thus trading off between effectiveness and explainability.

For both the explainable and not explainable models, the LtR training set is constructed using queries from the training dataset where the accepted answerer is labeled as an expert, including ζ queries ordered from the most recent to the oldest. It is important to note that this is a TUEF parameter: if the dataset contains more than ζ queries answered by experts, TUEF will take the latest ζ; otherwise, it will consider only the available queries. Given the limited size of the datasets, we use all available queries for training the LtR model. The exact number of queries used for each dataset is reported in Table 3, under the *LtR Train* row. After performing the MLG exploration for each query of the LtR training set, we excluded all the queries for which TUEF couldn't include in the candidate set the expert who provided the accepted answer. For the remaining queries, we extracted features for each query-candidate expert pair, as outlined in Section 3.3. The training set was split into training and validation sets following an 80/20 split. Hyper-parameter tuning is carried out using MRR on the validation set, leveraging the HyperOpt library [5], and optimizing four learning parameters: $learning\_rate \in [0.0001, 0.15]$, $num\_leaves \in [50, 200]$, $n\_estimators \in [50, 150]$, $max\_depth \in [8, 15]$, and $min\_data\_in\_leaf \in [150, 500]$. Table 3 also shows the average length of expert lists to rank under the *AvgExpList* row.

5 Experimental analysis

The following Section discusses two blocks of experiments. The first block (Section 5.1) regards an ablation study of TUEF across all the communities to understand the contribution to the overall performance of TUEF's different components (detailed in Section 3). To make EF decisions transparent and interpretable, the experiments also include assessing the impact of the integration in TUEF of the interpretable LtR algorithm (IlMart [60]) along with examples of TUEF's decision-making process. In the second block of experiments (Section 5.2), we conduct a comprehensive evaluation of TUEF aimed at assessing its effectiveness in two scenarios: end-to-end *Expert Ranking* scenario, and an offline *Expert Subsampling Ranking* scenario, where we conduct experiments with sampled metrics [42], ranking only a small set of candidates for each query. In both categories, we compare TUEF with state-of-the-art competitors for which the implementation is publicly available. Finally, in Section 5.3, we study the ability of TUEF to scale with larger datasets by considering four datasets corresponding to 1, 2, 3, and 4 months of StackOverflow data.

5.1 Ablation study

In this section, we aim to address the first two research questions, RQ1 and RQ2. We compare TUEF in an end-to-end scenario with variants of the proposed solution, each exploiting only a subset of the TUEF components. By examining these different configurations, we can assess and quantify the impact of each component and gain

[4]https://lightgbm.readthedocs.io/en/stable/

Table 3. Statistics of the StackExchange communities used for the experiments.

	StackOverflow	Unix	AskUbuntu	Server Fault	Physics	Mathematics
Time Period	2020-12-01 2021-01-01	2015-01-01 2023-09-04	2015-01-01 2023-09-04	2015-01-01 2023-09-04	2015-01-01 2023-09-04	2022-01-01 2023-09-04
Train	39,581	51,995	43,469	30,797	53,412	46,190
Test	9,521	12,985	10,166	7,398	13,860	11,649
Answerers	11,753	6,458	9,186	6,341	5,612	4,626
Tags	5,365	1,876	2,021	2,137	811	1,351
Clusters	10	6	9	10	8	9
Silhouette	0.81	0.46	0.39	0.42	0.41	0.50
AvgNodes	987.20	813.83	616.67	615.30	1,212.12	1,032.67
AvgEdges	26,843.2	25,521	18,274	12,960	54,583	51,090
MinAccAns	11	21	9	15	31	41
Experts	350	183	265	180	146	141
LtR Train	8008	22098	16918	10041	19192	15128
AvgExpList	111	118	129	102	102	85

insights into the effectiveness of combining social and content information in our approach for addressing the EF task for CQA platforms:

- **BC**: It uses the MLG and sorts the experts in the layers related to the new question based on their Betweenness centrality score.
- **BM25**: It uses the MLG and sorts the experts in the question's layers based on the BM25 score computed between the query and their previously answered questions. As in TUEF, the ranked lists of candidate experts retrieved for the tag and content indexes are merged by alternating their elements.
- **TUEF $_{\mathbf{NB}}$**: It only uses the MLG and applies the Network-based method. Candidates are ranked using a LtR model exploiting the static and the following query-dependent features: LayerCount, QueryKnowledge, VisitCountNetwork, StepsNetwork, BetweennessPos, BetweennessScore, Eigenvector, PageRank, Closeness, Degree, and AvgWeights.
- **TUEF $_{\mathbf{CB}}$**: It only uses the MLG and applies the Content-based method. Candidates are ranked using a LtR model exploiting the static and the following query-dependent features: LayerCount, QueryKnowledge, VisitCountContent, StepsContent, ScoreIndexTag, ScoreIndexText, FrequencyIndexTag, FrequencyIndexTex, Degree, and AvgWeights.
- **TUEF $_{\mathbf{SL}}$**: Unlike TUEF, which constructs a MLG by clustering tags into at least two layers, this variant represents user relationships using a Single Layer graph. Instead of forming multiple clusters, it treats all tags as belonging to a single cluster, resulting in a single-layer representation. All other phases remain unchanged.
- **TUEF $_{\mathbf{NoRW}}$**: In contrast to TUEF, it skips the Exploratory phase and does not perform Random Walks to extend the set of candidate experts selected considering the probability of receiving an answer.

- **TUEF $_{\mathbf{Lin}}$**: It ranks the experts according to the solution of [71], which uses a linear combination of features modeling experts, questions, and users' expertise.
- **TUEF $_{\mathbf{IlMart}}$**: It integrates IlMart into the TUEF framework, the interpretable version of the LambdaMart algorithm proposed in [60], to learn the expert ranking model. All other TUEF components remain unchanged.

Table 4. Ablation Study Results. The table reports the P@1, NDCG@3, R@5, and MRR scores of TUEF and the baselines representing its different components. Scores marked with the symbol indicate that TUEF statistically outperforms the corresponding baseline according to the paired t-test with Bonferroni correction and p-value<0.05. Conversely, the symbol indicates that the corresponding baseline statistically outperforms TUEF. Underlined values indicate that the baseline has a higher score than TUEF, but the difference is not statistically significant.

Model \ Dataset	StackOverflow				Unix				AskUbuntu			
	P@1	NDCG@3	R@5	MRR	P@1	NDCG@3	R@5	MRR	P@1	NDCG@3	R@5	MRR
BC	0.009	0.017	0.045	0.030	0.046	0.087	0.165	0.122	0.006	0.010	0.051	0.044
*BM*25	0.209	0.259	0.325	0.259	0.116	0.217	0.396	0.254	0.098	0.196	0.352	0.225
$TUEF_{NB}$	0.062	0.127	0.236	0.140	0.27	0.353	0.494	0.376	0.126	0.182	0.299	0.208
$TUEF_{Lin}$	0.263	0.383	0.569	0.406	0.297	0.382	0.542	0.41	0.184	0.285	0.440	0.307
$TUEF_{SL}$	0.431	0.565	0.750	0.564	0.336	0.430	0.585	0.452	0.255	0.363	0.513	0.375
$TUEF_{CB}$	0.455	0.589	0.761	0.587	0.332	0.424	0.573	0.444	0.244	0.350	0.502	0.364
$TUEF_{NoRW}$	0.446	0.582	0.744	0.573	0.334	0.425	0.575	0.436	0.249	0.342	0.472	0.345
$TUEF_{IlMart}$	0.360	0.496	0.676	0.502	0.324	0.416	0.565	0.438	0.203	0.305	0.468	0.327
TUEF	0.459	0.592	0.760	0.590	0.331	0.425	0.581	0.448	0.241	0.345	0.500	0.363
Model \ Dataset	Server Fault				Physics				Mathematics			
	P@1	NDCG@3	R@5	MRR	P@1	NDCG@3	R@5	MRR	P@1	NDCG@3	R@5	MRR
BC	0.010	0.040	0.099	0.071	0.014	0.027	0.093	0.059	0.002	0.040	0.136	0.059
*BM*25	0.094	0.152	0.251	0.167	0.103	0.189	0.334	0.226	0.111	0.199	0.361	0.236
$TUEF_{NB}$	0.21	0.253	0.364	0.285	0.101	0.162	0.277	0.196	0.084	0.145	0.278	0.178
$TUEF_{LIN}$	0.260	0.368	0.524	0.388	0.139	0.217	0.357	0.253	0.177	0.27	0.427	0.301
$TUEF_{SL}$	0.279	0.389	0.547	0.406	0.220	0.299	0.440	0.331	0.208	0.315	0.471	0.335
$TUEF_{CB}$	0.267	0.377	0.538	0.395	0.215	0.294	0.429	0.325	0.219	0.327	0.482	0.345
$TUEF_{NORW}$	0.272	0.378	0.527	0.382	0.218	0.295	0.429	0.313	0.210	0.313	0.469	0.321
$TUEF_{IlMart}$	0.250	0.364	0.529	0.384	0.195	0.276	0.418	0.309	0.197	0.301	0.475	0.329
TUEF	0.266	0.382	0.547	0.400	0.221	0.300	0.440	0.332	0.207	0.316	0.478	0.340

Discussion. Table 4 reports the results of the ablation study. We mark statistically significant performance gains/losses with respect to TUEF with the symbols and (paired t-test with p-value<0.05 and Bonferroni correction), and we underline the performance figures numerically greater than those of TUEF.

TUEF consistently outperforms in all examined communities some baseline models, including *BC*, *BM*25, $TUEF_{NB}$, and $TUEF_{Lin}$. A recurring pattern emerges across communities, revealing *BC* and $TUEF_{NB}$ as the least effective baselines, both relying exclusively on network aspects. In contrast, *BM*25's performance highlights the

significance of the content-based component, as evidenced by a minimum of 10% of queries across communities being best addressed by experts who have previously answered similar questions. $TUEF_{Lin}$ consistently exhibits lower performance compared to TUEF, underscoring the efficacy of TUEF's LtR methodology in capturing hidden relationships among expert features.

$TUEF_{SL}$ achieves slightly higher or comparable performance across most communities, with the exceptions of StackOverflow and AskUbuntu. On StackOverflow, TUEF significantly outperforms $TUEF_{SL}$, whereas on AskUbuntu the opposite occurs, suggesting community-specific effects. This pattern is consistent with the Silhouette values reported in Table 3, which measure the effectiveness of clustering within the MLG. StackOverflow's high Silhouette value of 0.805 indicates a strong subdivision of tags into clusters, which benefits TUEF 's use of topic layers. In contrast, AskUbuntu shows the lowest Silhouette value (0.391), reflecting weaker cluster quality and contributing to the stronger performance of $TUEF_{SL}$. These results highlight an important implication: the benefit of MLG depends on the structural properties of the community. In communities such as Physics and Mathematics, where topical boundaries are less pronounced and activity is more evenly distributed, subdividing into multiple layers introduces limited additional signal and thus does not yield substantial gains. In such cases, the simpler single-layer representation remains competitive. Conversely, when communities show clear topical subdivisions and sufficient activity within each topic, MLG provides a significant advantage by ensuring that expertise is assessed in the appropriate topical context.

Moreover, the values of $TUEF_{CB}$ underscore content supremacy over the network component, although incorporating both components in TUEF marginally enhances system performances for most communities. Finally, $TUEF_{NoRW}$ exhibits lower recall (R@5) without a decrease in precision (P@1). The similar P@1 scores between TUEF and $TUEF_{NoRW}$ result from TUEF's random walk mechanism, which identifies a larger set of relevant experts. TUEF significantly improves recall (R@5) by consistently including the correct expert within the top five positions. However, this larger candidate set makes it more challenging to rank the best expert in the first position, thus keeping P@1 similar for both models. Overall, TUEF's random walk enhances recall by expanding the candidate set, even though it complicates achieving the highest precision score.

In summary, to address RQ1, our results show that adopting the MLG substantially improves the selection of highly relevant experts and thus boosts the model's accuracy, especially in communities where topics are well-structured. Moreover, analyzing the two main components of TUEF (i.e., Content-based and Network-based) reveals that content information primarily drives EF performance, with the network signals adding a smaller but still meaningful improvement.

Interpretability. The explanations derived from an interpretable model such as $TUEF_{llMart}$ can serve multiple purposes at both the user and system design levels. For example, they can guide feature selection and refinement for the original model by highlighting which signals are most influential, or they can increase user trust and engagement by making the decision process transparent and easier to understand. Additionally, explanations enable designers to identify model errors, improve answer relevance, and refine the interface to better support user interactions. These advantages, however, do not come for free. $TUEF_{llMart}$ consistently exhibits statistically lower performance than TUEF, with the most notable disparity occurring in the StackOverflow community, where the difference reaches 27.5% for P@1 and 12.43% for R@5. In other communities, the variations are still statistically significant, ranging from a maximum of 18.71% (AskUbuntu) to a minimum of 2.16% (Unix) in terms of P@1. However, the differences diminish significantly when considering R@5, with the highest value being 6.83% (AskUbuntu) and no statistical differences in Mathematics, indicating $TUEF_{llMart}$ proficiency in placing the top respondent within the first five positions.

In fact, $TUEF_{llMart}$ capacity to offer model result explanations may offset the observed performance decline. TUEF, on the other hand, employs a more sophisticated decision tree-based LtR approach, which contributes to higher performance but offers only minimal clarity regarding the significance of different factors influencing the

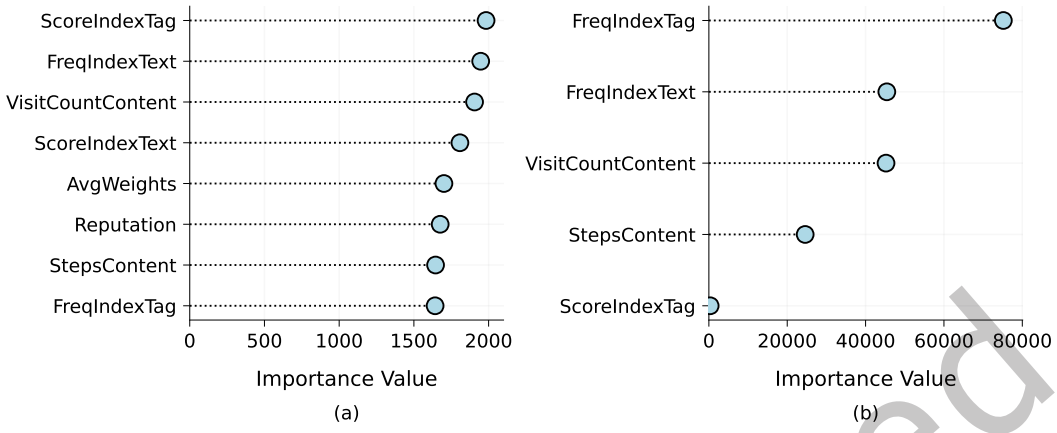

Fig. 3. LtR Algorithm's Feature Importances. Figure (a) on the left displays the eight most important features for the TUEF LtR algorithm. Figure (b) illustrates the feature importance values for all features considered by TUEF $_{\text{IlMart}}$.

ranking result. Figure 3 illustrates the feature importance of both models (LambdaMart and $TUEF_{IlMart}$) for the StackOverflow community. It reports LambdaMart's top height features and all those considered by $TUEF_{IlMart}$, all of which are present among LambdaMart's top eight features, albeit in a different order. For example, the ScoreIndexTag feature, most used by LambdaMart, is the least used by $TUEF_{IlMart}$. Most of the crucial features for LambdaMart are associated with the content-based method. Nevertheless, features from the network-based method, such as Closeness, PageRank, and AvgWeight, exhibit high importance values. On the other hand, all the features selected by $TUEF_{IlMart}$ are content-based, highlighting the importance of the text component for the EF task.

Figure 4 illustrates how $TUEF_{IlMart}$ can be used for interpretability purposes, showcasing the contribution to the predicted score of the three IlMart main effects with the highest feature importance values (i.e., FreqIndexTag, VisitCountContent, and FreqIndexText) and the most important interaction effect learned from $TUEF_{IlMart}$ for the StackOverflow community. High values of these features indicate proficient expert competence. The 2D plots and the heatmap are built by aggregating all the trees using the same features, i.e., by representing the contribution of each main effect $\tau(x_j)$ and each interaction effect $\tau(x_i, x_j)$, where $\tau(x_j)$ and $\tau_{ij}(x_i, x_j)$ are ensembles of trees. Through Table 5, we analyze instances where TUEF correctly ranks the best answerer first as well as cases where it fails, providing qualitative examples that offer insight into the model's internal decision process. Specifically, Table 5 presents four question examples: the first two showcase instances where TUEF successfully identifies the best expert by placing them in the first position. For a better understanding, we report the features of the top two ranked experts. Conversely, for the last two examples where TUEF fails, we show the feature values of the experts in ranking order up to the position where the ranker placed the best expert. The *Rank* column indicates the expert's ranking position in the final list. In contrast, the *BestExpert* column serves as a label, with a value equal to 1 for the ground truth expert and 0 otherwise.

A notable pattern emerges in both instances of TUEF's success and failure: the expert occupying the first position consistently demonstrates higher values for the features evaluated by the $TUEF_{IlMart}$ ranker. This results in a greater contribution and, consequently, a higher final score, as shown in Figure 4. For example, consider the first instance in Table 5, with $QID = 65438592$, and focus on the feature *FreqIndexTag*. The BestExpert (first row of Table 5) has a value of 45, corresponding to a contribution of approximately 1, as shown in the upper left

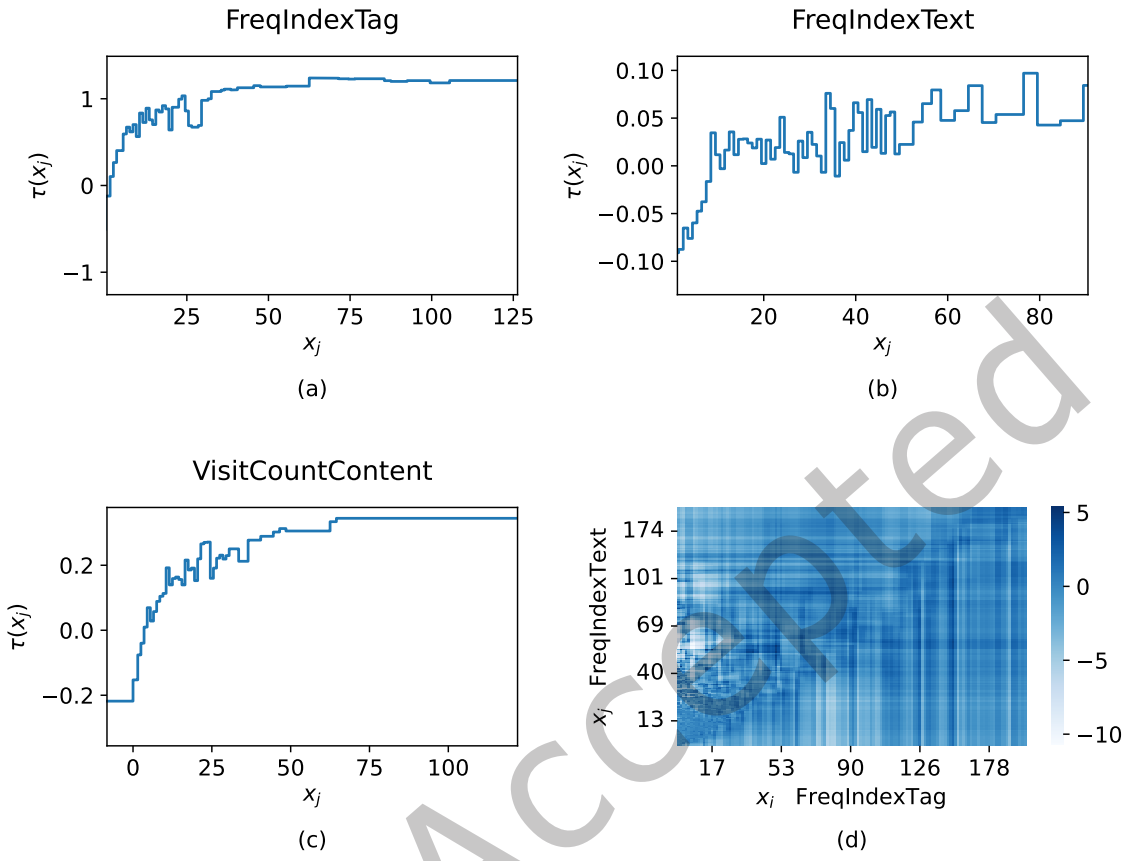

Fig. 4. The three most important main and interaction effects of TUEF $_{\text{IlMart}}$ on the StackOverflow dataset. Figures (a), (b), and (c) show the main effects of FreqIndexTag, FreqIndexText, and VisitCountContent, respectively. The x-axis represents the values the feature can have, while the y-axis represents the corresponding contribution of the feature to the predicted final score. Figure (d) illustrates the most important interaction effect learned by TUEF $_{\text{IlMart}}$, composed of the features FreqIndexTag (x-axis) and FreqIndexText (y-axis). The color bar indicates the interaction contribution.

figure. In contrast, the expert placed in the second position (second row of Table 5) has a value of 7, contributing to a measure of less than 0.5 (half of the first). By combining all feature contributions computed with the aid of Figure 4, we can comprehend the final ranking and understand why the model could or could not place the BestExpert in the first position.

In summary, to address RQ2, our results show that the interpretable model ($TUEF_{IlMart}$) exhibits a consistent drop in expert selection accuracy compared to the more complex $TUEF$, yet it still reliably places the top expert in the first five positions. In return, $TUEF_{IlMart}$ provides transparent explanations of its ranking decisions, helping users understand which features drive its predictions and system designers to identify potential weaknesses in the model, refine feature representations, and prioritize further training data collection or model improvements. Thus, there is a clear trade-off: while interpretability reduces absolute performance, it enhances model transparency, potentially benefiting scenarios where trust and explainability are critical.

Table 5. Examples of TUEF $_{\text{IlMart}}$ Decision-Making Process. The table presents four examples of questions: two where TUEF $_{\text{IlMart}}$ successfully identified the right expert (i.e., placing the right expert in the first position of the final ranking) and two where it was unsuccessful. For each question, the table reports the values of FreqIndexTag, VisitCountContent, and FreqIndexText, along with the final expert Score, the position in the final ranking, and the BestExpert label.

QID	FreqIndexTag	VisitCountContent	FreqIndexText	Score	Rank	BestExpert
65438592	45	17	54	2.39	1	1
	7	7	7	1.96	2	0
65438859	2	10	13	-0.12	1	1
	1	1	9	-1.7	2	0
65438631	81	62	82	3.29	1	0
	30	25	30	2.83	2	1
65439018	229	14	202	3.5	1	0
	14	5	12	2.58	2	0
	11	9	16	2.45	3	1

5.2 State-of-the-art Baselines

In this section, we evaluate TUEF based on the two approaches for the EF task discussed in Section 2. As done in Section 5.1, to assess the statistical significance of the performance gains/losses (and) measured with our experiments, we conducted a paired t-test with a p-value threshold of 0.05 and Bonferroni correction. Moreover, we underline the performance numerically (not statistically) greater than those of TUEF. With this analysis, we aim to answer the third question RQ3.

Expert Ranking. The first set of baselines follows the *Expert Ranking* configuration, which involves an end-to-end process of selecting users as experts, computing the similarity between a candidate expert and the query, and subsequently ranking the candidate experts. TUEF adopts this configuration, and we compare it with the models introduced in [37], enabling us to evaluate the effectiveness of current neural re-rankers on this task, as well as the *t-BGER* model [43].

Kasela et al. [37] employ a two-stage ranking architecture, where the initial stage uses a recall-oriented retriever, and the subsequent stage utilizes a precision-oriented approach. In the first stage, they use Elastic Search's *BM25* model to generate an ordered and limited list of candidate experts. In the second stage, they apply a linear combination of scores to re-rank these experts. This combination includes (i) the BM25 score, (ii) the similarity score computed by neural network-based re-ranker approaches, and for three of the six proposed baselines, (iii) the score generated by a personalized model based on user history. The neural network-based re-ranker models are *DistilBERT*[5] and *MiniLM*[6]. Both pre-trained models are fine-tuned for ten epochs for each community considered. The personalized model, called *TAG*, captures the similarity between the topics previously addressed by the asker and the answerer based on the tags in the questions asked and the answers provided. Finally, the scores are combined and computed as the weighted sum of the normalized model scores. The models' weights are determined using a grid search on the validation set. The proposed baselines are *BM25*, *BM25+TAG*,

[5]https://huggingface.co/distilbert-base-uncased
[6]https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

Table 6. Expert Ranking Comparison. The table reports the P@1, NDCG@3, R@5, and MRR scores of TUEF and the baselines in the Expert Ranking category for all selected StackExchange communities. Scores marked with the symbol indicate that TUEF statistically outperforms the corresponding baseline according to the paired t-test with Bonferroni correction and p-value<0.05. Conversely, the symbol indicates that the corresponding baseline statistically outperforms TUEF. Underlined values indicate that the baseline has a higher score than TUEF, but the difference is not statistically significant.

Model \ Dataset	Stack Overflow				Unix				Ask Ubuntu			
	P@1	NDCG@3	R@5	MRR	P@1	NDCG@3	R@5	MRR	P@1	NDCG@3	R@5	MRR
BM25	0.257	0.366	0.513	0.381	0.253	0.348	0.492	0.365	0.141	0.237	0.389	0.261
BM25+TAG	0.307	0.431	0.604	0.441	0.260	0.345	0.494	0.368	0.151	0.256	0.415	0.277
MiniLM	0.282	0.384	0.527	0.399	0.262	0.352	0.485	0.367	0.158	0.259	0.412	0.279
MiniLM+TAG	0.320	0.441	0.624	0.456	0.269	0.358	0.492	0.371	0.172	0.273	0.419	0.292
DistilBERT	0.288	0.394	0.542	0.407	0.268	0.356	0.492	0.373	0.161	0.264	0.415	0.283
DistilBERT+TAG	0.337	0.455	0.627	0.468	0.276	0.363	0.492	0.376	0.178	0.282	0.438	0.302
t-BGER	0.254	0.338	0.478	0.36	0.25	0.314	0.43	0.345	0.202	0.312	0.508	0.335
TUEF	0.459	0.592	0.760	0.590	0.331	0.425	0.581	0.448	0.241	0.345	0.500	0.363
Model \ Dataset	Server Fault				Physics				Mathematics			
	P@1	NDCG@3	R@5	MRR	P@1	NDCG@3	R@5	MRR	P@1	NDCG@3	R@5	MRR
BM25	0.199	0.286	0.430	0.311	0.164	0.234	0.364	0.267	0.149	0.235	0.378	0.265
BM25+TAG	0.218	0.309	0.475	0.331	0.166	0.239	0.374	0.271	0.149	0.232	0.377	0.258
MiniLM	0.200	0.293	0.436	0.313	0.172	0.248	0.382	0.278	0.159	0.248	0.403	0.277
MiniLM+TAG	0.224	0.315	0.476	0.337	0.172	0.25	0.386	0.279	0.161	0.248	0.405	0.277
DistilBERT	0.215	0.302	0.449	0.325	0.178	0.256	0.386	0.284	0.170	0.263	0.411	0.291
DistilBERT+TAG	0.223	0.319	0.478	0.341	0.179	0.254	0.384	0.283	0.17	0.262	0.414	0.291
t-BGER	0.275	0.356	0.474	0.375	0.080	0.151	0.258	0.180	0.122	0.169	0.295	0.215
TUEF	0.266	0.382	0.547	0.400	0.221	0.300	0.440	0.332	0.207	0.316	0.478	0.340

MiniLM, *MiniLM+TAG*, *DistilBERT*, and *DistilBERT+TAG*. Although not explicitly stated for readability, the last four baselines also incorporate *BM25*.

In contrast, Krishna et al. [43] introduce *t-BGER*, a temporal weighted bipartite graph model. Their study emphasizes the importance of utilizing tags for expert identification while reducing task complexity. The model is based on a user-tag bipartite graph, where the edge weight reflects the user's activity related to the specific tag. Additionally, they incorporate temporal discounting into this graph, assigning higher weights to recent activities. Finally, they apply resource allocation techniques through network-based inference on the bipartite graph, effectively managing highly sparse data.

As reported in Table 6, among the baselines, *DistilBERT+TAG* shows superior performance. However, *t-BGER* demonstrates better scores for the AskUbuntu and Server Fault communities. Overall, TUEF surpasses all baselines across various metrics, except for P@1 in the Server Fault community, where *t-BGER* exhibits a relative improvement of 3.38%, and R@5 in the AskUbuntu community, where *t-BGER* has a slightly higher score. Excluding these two exceptions, TUEF records minimum relative improvements of 19.31% for P@1 in the AskUbuntu community, 7.30% for NDCG@3, 15.40% for R@5, and 6.67% for MRR in the Server Fault community. In the remaining communities, TUEF reports higher relative improvements across all other metrics.

Expert Subsample Ranking. The second set of baselines follows the Expert Ranking evaluation approach and aims to rank a small set of 20 experts when presented with a new query (see Section 2). This set always includes (i) the users who have provided the answers, including the best answerer, and (ii) a variable number of users selected to reach a total of 20 candidate experts, chosen from the top 10% of users identified as the most active answerers. Since this definition of the EF task deviates from the task previously considered by requiring the prior knowledge of who will answer the query, we adapted TUEF to allow a fair comparison with the following two state-of-the-art models for the EF task, whose code is publicly available:

- **NeRank [52]**: It models the CQA platform as a heterogeneous network to learn representations for question raisers and question answerers through a metapath-based algorithm. Using this heterogeneous network, NeRank preserves relationship information while modeling the question's content with a single-layer LSTM. Finally, a CNN assigns a score to each expert for a given question, representing the probability of the expert providing the accepted answer.
- **PMEF [65]**: It consists of three main modules: a multi-view question encoder for learning comprehensive question features, an intra-view encoder to discover view-specific interactions among experts and target questions, and an inter-view encoder designed to extract expert/question features by integrating different view information in a personalized manner.

We always consider the questions selected for the experimentation phase for each community. However, the test set is smaller due to constraints imposed by NeRank, PMEF, and, in this specific setup, TUEF as well. Specifically, both NeRank and PMEF models require that the user posing the question has previously asked other questions to capture the interactions between the questioner and responder. Additionally, the test set includes only those questions for which TUEF, through graph exploration, included the best answerer in the set of potential experts. Moreover, to ensure a fair comparison among the three models, we required each to rank the same set of experts. This set comprises users who have answered the question, including the best answerer, and additional experts randomly chosen from those identified by TUEF as candidates. This procedure is the fairest, considering the way TUEF selects candidate experts. Specifically, as a topic-based model, TUEF chooses hard-negative samples because they are always users who have previously answered precisely the topics related to the query. In contrast, following the procedure used by NeRank and PMEF, which initially selects 20 experts from the top 10% of the most active users while always including the real answerer, increases the likelihood of non-relevant experts being included in the ranking. These experts are then placed at the bottom of the list with high probability. We consider in the comparison also TUEF and $TUEF_{IlMart}$ to study the behavior of the interpretable version of TUEF under this different configuration.

Table 7 shows the results of the three models across different communities. PMEF consistently outperforms NeRank, except in the StackOverflow and Mathematics communities. However, TUEF exhibits significantly better performance than PMEF, with the smallest relative improvement being 42.42% for P@1 and 29.8% for MRR in the Unix community, and 21.92% for R@5 in the AskUbuntu community. Other metrics in the remaining communities show even more substantial relative improvements than those mentioned.

$TUEF_{IlMart}$ consistently outperforms NeRank and PMEF. However, it exhibits lower performance across various communities compared to TUEF, except in the Unix and Physics communities, where it demonstrates statistical superiority in specific metrics. $TUEF_{IlMart}$ has already demonstrated performance comparable to TUEF in these communities during the ablation study (Table 4). As already highlighted in Section 5.1, while $TUEF_{IlMart}$ enhances transparency in the expert ranking process, it does so at the cost of a light, but statistically significant decrease in performance across multiple metrics. However, despite this degradation, $TUEF_{IlMart}$ remains a practical alternative in scenarios where interpretability and transparency are prioritized over achieving the highest possible ranking performance. In particular, $TUEF_{IlMart}$ proves to be a promising compromise for less topic-sensitive communities, where it demonstrates performance comparable to TUEF.

Table 7. Expert Subsample Ranking Comparison. The table reports the P@1, NDCG@3, R@5, and MRR scores of TUEF and the baselines in the Expert Ranking category for all selected StackExchange communities. Scores marked with the symbol indicate that TUEF statistically outperforms the corresponding baseline according to the paired t-test with Bonferroni correction and p-value<0.05. Conversely, the symbol indicates that the corresponding baseline statistically outperforms TUEF. Underlined values indicate that the baseline has a higher score than TUEF, but the difference is not statistically significant.

Model \ Dataset	StackOverflow				Unix				Ask Ubuntu			
	P@1	NDCG@3	R@5	MRR	P@1	NDCG@3	R@5	MRR	P@1	NDCG@3	R@5	MRR
Nerank	0.313	0.460	0.718	0.491	0.385	0.490	0.646	0.514	0.217	0.364	0.651	0.406
PMEF	0.116	0.213	0.422	0.273	0.429	0.556	0.718	0.567	0.298	0.472	0.740	0.492
$TUEF_{IlMart}$	0.700	0.815	0.939	0.804	0.619	0.744	0.899	0.741	0.528	0.673	0.875	0.673
TUEF	0.800	0.883	0.989	0.877	0.611	0.738	0.906	0.736	0.574	0.718	0.901	0.712
Model \ Dataset	Server Fault				Physics				Mathematics			
	P@1	NDCG@3	R@5	MRR	P@1	NDCG@3	R@5	MRR	P@1	NDCG@3	R@5	MRR
Nerank	0.292	0.403	0.597	0.442	0.187	0.341	0.621	0.380	0.233	0.372	0.608	0.407
PMEF	0.319	0.447	0.654	0.471	0.214	0.320	0.556	0.378	0.130	0.270	0.523	0.313
$TUEF_{IlMart}$	0.498	0.668	0.850	0.660	0.520	0.674	0.860	0.673	0.507	0.627	0.805	0.642
TUEF	0.594	0.745	0.916	0.733	0.487	0.635	0.858	0.640	0.648	0.669	0.907	0.760

Finally, note that TUEF performs considerably better in this set of experiments compared to the cases analyzed in Tables 4 and 6. This improvement is the result of two factors: (i) the test set includes only queries for which TUEF successfully identified the best answerer during graph exploration, and (ii) the set of candidates to be ranked is limited to twenty users, approximately one-fifth of those that TUEF typically selects and sorts on average (see Table 3). We remark that this configuration and the specific settings for these experiments were required to ensure a fair comparison with the other considered algorithms.

In summary, our findings indicate that the Expert Subsample Ranking setup simplifies the EF scenario by restricting ranking to a small, randomly chosen set of 20 users—always including the actual answerers. While this approach may accelerate evaluation, it does not reflect the real challenge of expert finding. In practice, an EF system must: (i) accurately select relevant experts from the entire pool of active community members for each query; (ii) precisely rank those selected experts, matching the query's topics and requirements with the experts' skills. Focusing exclusively on a small random subset of experts can mask the true complexity of EF, where both robust candidate selection and fine-grained ranking are critical for practical performance.

5.3 TUEF scalability

In this section, we aim to answer the last research question RQ4 by measuring the TUEF ability to scale and adapt to larger datasets.Table 8 shows key statistics for four datasets corresponding to 1, 2, 3, and 4 months of StackOverflow data, including the number of questions used for training the MLG and the LtR model, the number of answers, the number of users labeled as experts, the minimum number of accepted answers the experts have (MinAccAns), the number of tags associated with questions, the number of MLG's Layers, and the average length of experts lists to rank at inference time (AvgExpLists). The data selection follows a reverse chronological order: for a given period length, the most recent month is always included, and additional months are incrementally added backward in time. The test set remains consistent across all experiments and is composed of data occurring

after the selected training period. This approach ensures a fair evaluation of the model's performance while analyzing the impact of different historical data lengths.

The training set size increases approximately linearly, with a consistent monthly growth rate, expanding from 38,616 questions in the 1-month dataset to 165,320 questions in the 4-month dataset. The increase in the number of questions is associated with an increase in the number of unique tags, which rise from 5,365 to 10,514, indicating a richer set of topics being covered over time. As expected, using more data increases the number of identified experts (from 350 to 1,015), thereby enriching the expert base. This increase in identified experts by TUEF leads to a corresponding increment in the number of retrieved experts at inference time, as reported in the AvgList column.

Table 8. Statistics for four datasets corresponding to 1, 2, 3, and 4 months of StackOverflow data, including the number of questions used for train and LtR, the number of answers, the number of users labeled as experts, the minimum number of accepted answers the experts have (MinAccAns), the number of tags associated with questions, the number of MLG's layers, and the average length of experts lists to rank at inference time (AvgExpLists).

Months	Train	Answers	LtR	Experts	MinAccAns	Tags	Layers	AvgExpLists
1	38616	53717	8008	350	11	5365	10	111.03
2	79451	110169	19991	577	14	7546	10	146.74
3	122612	170131	33164	819	15	9136	10	182.68
4	165320	230246	44821	1015	16	10514	10	186.06

Figure 5 presents the performance metrics evaluated on a test set comprising 1,432 questions. The P@1 metric slightly decreases from 0.492 (1-month) to 0.449 (4-month). The NDCG@3 metric decreases from 0.631 to 0.590, while the MRR slightly drops from 0.625 to 0.589. However, the R@5 remains more stable for the datasets corresponding to 1, 2, and 3 months, and only slightly decreases from 0.802 to 0.771 for the 4-month dataset, indicating that the model consistently ranks the best answerer within the top five positions. The stability of the R@5 metric is particularly noteworthy, as it suggests that while TUEF may slightly reduce its performance in identifying the best experts at the top position, it still effectively includes them within the first five positions.

Contrary to the general expectation that larger training datasets improve model performance, TUEF exhibits a slight decline in accuracy metrics as the dataset size increases. This counterintuitive result is primarily linked to the growth in the number of eligible experts rather than the volume of data itself. As discussed in Section 4.3, an ideal Expert Finding method would consider all potential answerers when selecting experts for a question, thereby effectively mitigating the cold start problem. However, given the massive user base on platforms like StackOverflow, this approach is computationally impractical. Consequently, most Expert Finding methods employ pre-processing techniques to reduce the pool of eligible experts.

When the pool of eligible experts is smaller, the model achieves higher accuracy due to reduced ranking complexity. However, as the dataset size increases, the pool of eligible experts grows accordingly (see Experts and AvgExpLists in Table 8), leading to more complex ranking scenarios since the model must now distinguish among a larger pool of potentially highly relevant experts. Despite this increased complexity, TUEF demonstrates robust performance, maintaining competitive accuracy even as the task becomes more challenging.

Finally, the rightmost plot shows the average latency of a single query, which increases from 0.107 seconds for the 1-month dataset to 0.167 seconds for the 4-month dataset. This increment can be attributed to the larger size of the CQA community in the 4-month dataset. Specifically, the greater number of questions leads to a larger and more dense MLG as more users and their interactions are included. Consequently, the RW applied to this denser MLG may perform more steps, resulting in the selection of more candidate experts to rank, thereby

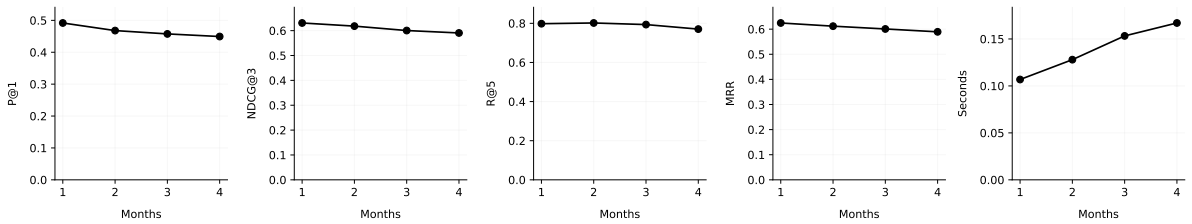

Fig. 5. TUEF performance with four datasets corresponding to 1, 2, 3, and 4 months of StackOverflow data. The x-axis specifies the number of months considered, while the y-axis reports the performance metrics, including P@1, NDCG@3, R@5, MRR computed on the same test set of 1,342 queries. The rightmost plot indicates the per-query TUEF average inference time in seconds.

increasing the time required to perform the inference. Overall, TUEF demonstrates robust scalability, effectively managing larger datasets while maintaining a high level of performance, with slight declines in precision due to the increased data volume and complexity.

In summary, to answer RQ4, the key findings indicate that although the number of eligible experts grows with dataset size—thereby increasing ranking complexity—TUEF maintains competitive accuracy. Specifically, despite a slight decline in precision metrics (e.g., P@1) due to the larger pool of potentially highly relevant experts, recall remains stable, demonstrating that TUEF adapts well to larger datasets. Moreover, inference time scales efficiently, showing only a modest increase from the 1-month to the 4-month dataset.

6 Conclusion

In this paper, we extended the analysis of TUEF, a Topic-oriented User-Interaction model for Expert Finding in Community Question&Answering platforms. TUEF integrates content and social data by constructing a Multi-Layer Graph to represent user interactions based on topical answering patterns. This approach allows TUEF to leverage both topic specificity and social relationships within the community to identify and rank the most knowledgeable users for any given question. The ablation study results show that TUEF's performance is particularly notable in larger communities with well-defined topic clusters. Additionally, by incorporating interpretable Learning-to-Rank algorithms (i.e., llMart [60]), TUEF achieves complete transparency, allowing a comprehensive understanding of the decision-making processes without significantly decreasing performance. Our extensive experiments conducted on multiple Stack Exchange communities demonstrate that TUEF outperforms state-of-the-art EF models in both the Expert Ranking and Expert Subsample Ranking categories. It excels in precision-oriented metrics such as P@1, NDCG@3, MRR, and R@5, with improvements over the state-of-the-art by a minimum of 42.42% in P@1, 32.73% in NDCG@3, 21.76% in R@5, and 29.81% in MRR. The study also highlighted TUEF's ability to handle larger datasets, as evidenced by its performance on datasets corresponding to 1, 2, 3, and 4 months of StackOverflow data collection. Specifically, TUEF consistently ranks relevant experts in top positions, demonstrating its reliability and robustness.

While TUEF advances the state of the art in expert finding, we recognize some of its limitations. First, the framework requires setting several parameters, such as the number of clusters, which need to be carefully tuned for each domain. Once macro-topics are defined, the resulting layer structure is rigid, and performance advantages may degrade in communities where topic boundaries are overlapping. In addition, the current representation of expertise relies on knowledge vectors that capture sub-topic contributions, but omit other potentially relevant signals such as answer quality. The interpretable LtR variant also comes with a trade-off, as the gain in transparency sometimes comes at the cost of predictive accuracy. Finally, our ablation study shows

that the benefits of MLG vanish especially in small communities, where limited activity and weak topic separation reduce the benefits of MLG, making a single-layer graph equally effective.

Future work could mitigate these limitations by developing adaptive methods to tune parameters and improve clustering in noisy or sparse settings. Another promising direction is to enrich user representations with additional signals (e.g., quality or temporal factors) and design variants of TUEF that better balance interpretability and accuracy across different community structures.

Acknowledgments

This work was partially supported by: the H2020 SoBig- Data++ project (#871042); the CAMEO PRIN project (#2022ZLL7MW) funded by the MUR; the HEU EFRA project (#101093026) funded by the EC under the NextGeneration EU programme. A. Passarella's and R. Perego's work was partly funded under the PNRR - M4C2 - Investimento 1.3, PE00000013 - "FAIR" project. However, the views and opinions expressed are those of the authors only and do not necessarily reflect those of the EU or European Commission-EU. Neither the EU nor the granting authority can be held responsible for them.

References

[1] Maddalena Amendola, Andrea Passarella, and Raffaele Perego. 2024. Towards Robust Expert Finding in Community Question Answering Platforms. In *Advances in Information Retrieval*, Nazli Goharian, Nicola Tonellotto, Yulan He, Aldo Lipani, Graham McDonald, Craig Macdonald, and Iadh Ounis (Eds.). Springer Nature Switzerland, Cham, 152–168.

[2] Kolmogorov An. 1933. Sulla determinazione empirica di una legge didistribuzione. *Giorn Dell'inst Ital Degli Att* 4 (1933), 89–91.

[3] Çiğdem Aslay, Neil O'Hare, Luca Maria Aiello, and Alejandro Jaimes. 2013. Competition-based networks for expert finding. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. 1033–1036.

[4] Elias Bassani. 2022. ranx: A blazing-fast python library for ranking evaluation and comparison. In *European Conference on Information Retrieval*. Springer, 259–264.

[5] James Bergstra, Daniel Yamins, and David Cox. 2013. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *International conference on machine learning*. PMLR, 115–123.

[6] Christopher JC Burges. 2010. From ranknet to lambdarank to lambdamart: An overview. *Learning* 11, 23-581 (2010), 81.

[7] Gordon Burtch, Dokyun Lee, and Zhichen Chen. 2024. The consequences of generative AI for online knowledge communities. *Scientific Reports* 14, 1 (2024), 10413.

[8] Rocío Cañamares and Pablo Castells. 2020. On target item sampling in offline recommender system evaluation. In *Proceedings of the 14th ACM Conference on Recommender Systems*. 259–268.

[9] Hui Chen, John Coogle, and Kostadin Damevski. 2019. Modeling stack overflow tags and topics as a hierarchy of concepts. *Journal of Systems and Software* 156 (2019), 283–299. doi:10.1016/j.jss.2019.07.033

[10] Gianni Costa and Riccardo Ortale. 2023. Ask and Ye shall be Answered: Bayesian tag-based collaborative recommendation of trustworthy experts over time in community question answering. *Information Fusion* 99 (2023), 101856.

[11] Gianni Costa and Riccardo Ortale. 2023. Here are the answers. What is your question? Bayesian collaborative tag-based recommendation of time-sensitive expertise in question-answering communities. *Expert Systems with Applications* 225 (2023), 120042.

[12] Leuson Da Silva, Jordan Samhi, and Foutse Khomh. 2025. LLMs and Stack Overflow discussions: Reliability, impact, and challenges. *Journal of Systems and Software* (2025), 112541.

[13] Alexander Dallmann, Daniel Zoller, and Andreas Hotho. 2021. A case study on sampling strategies for evaluating neural sequential item recommendation models. In *Proceedings of the 15th ACM Conference on Recommender Systems*. 505–514.

[14] Arash Dargahi Nobari, Sajad Sotudeh Gharebagh, and Mahmood Neshati. 2017. Skill translation models in expert finding. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*. 1057–1060.

[15] Mahdi Dehghan and Ahmad Ali Abin. 2019. Translations Diversification for Expert Finding: A Novel Clustering-based Approach. *ACM Trans. Knowl. Discov. Data* 13, 3 (2019), 32:1–32:20. doi:10.1145/3320489

[16] Mahdi Dehghan, Maryam Biabani, and Ahmad Ali Abin. 2019. Temporal expert profiling: With an application to T-shaped expert finding. *Inf. Process. Manag.* 56, 3 (2019), 1067–1079. doi:10.1016/j.ipm.2019.02.017

[17] R Maria del Rio-Chanona, Nadzeya Laurentsyeva, and Johannes Wachs. 2024. Large language models reduce public knowledge sharing on online Q&A platforms. *PNAS nexus* 3, 9 (2024), pgae400.

[18] Fabrizio Dell'Acqua, Edward McFowland III, Ethan R Mollick, Hila Lifshitz-Assaf, Katherine Kellogg, Saran Rajendran, Lisa Krayer, François Candelon, and Karim R Lakhani. 2023. Navigating the jagged technological frontier: Field experimental evidence of the effects

of AI on knowledge worker productivity and quality. *Harvard Business School Technology & Operations Mgt. Unit Working Paper* 24-013 (2023).
[19] Olive Jean Dunn. 1961. Multiple comparisons among means. *Journal of the American statistical association* 56, 293 (1961), 52–64.
[20] Roohollah Etemadi, Morteza Zihayat, Kuan Feng, Jason Adelman, Fattane Zarrinkalam, and Ebrahim Bagheri. 2024. It Takes a Team to Triumph: Collaborative Expert Finding in Community QA Networks. In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*. 164–174.
[21] Muhammad Shahzad Faisal, Ali Daud, Abubakr Usman Akram, Rabeeh Ayaz Abbasi, Naif Radi Aljohani, and Irfan Mehmood. 2019. Expert ranking techniques for online rated forums. *Computers in Human Behavior* 100 (2019), 168–176.
[22] Zohreh Fallahnejad and Hamid Beigy. 2022. Attention-based skill translation models for expert finding. *Expert Systems with Applications* 193 (2022), 116433.
[23] Linton C Freeman. 1977. A set of measures of centrality based on betweenness. *Sociometry* (1977), 35–41.
[24] Chaogang Fu. 2019. Tracking user-role evolution via topic modeling in community question answering. *Information Processing & Management* 56, 6 (2019), 102075.
[25] Chaogang Fu. 2020. User correlation model for question recommendation in community question answering. *Applied Intelligence* 50 (2020), 634–645.
[26] Jinlan Fu, Yi Li, Qi Zhang, Qinzhuo Wu, Renfeng Ma, Xuanjing Huang, and Yu-Gang Jiang. 2020. Recurrent memory reasoning network for expert finding in community question answering. In *Proceedings of the 13th international conference on web search and data mining*. 187–195.
[27] Negin Ghasemi, Ramin Fatourechi, and Saeedeh Momtazi. 2021. User embedding for expert finding in community question answering. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 15, 4 (2021), 1–16.
[28] Eirini Giannakidou, Vassiliki Koutsonikola, Athena Vakali, and Yiannis Kompatsiaris. 2008. Co-Clustering Tags and Social Data Sources. In *2008 The Ninth International Conference on Web-Age Information Management*. 317–324. doi:10.1109/WAIM.2008.61
[29] Kiarash Golzadeh, Lukasz Golab, and Jaroslaw Szlichta. 2024. Explaining Expert Search Systems with ExES. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 5465–5468.
[30] Ben Goodrich, Vinay Rao, Peter J Liu, and Mohammad Saleh. 2019. Assessing the factual accuracy of generated text. In *proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 166–175.
[31] Radin Hamidi Rad, Hossein Fani, Ebrahim Bagheri, Mehdi Kargar, Divesh Srivastava, and Jaroslaw Szlichta. 2023. A variational neural architecture for skill-based team formation. *ACM Transactions on Information Systems* 42, 1 (2023), 1–28.
[32] Denis Helic and Tiago Santos. 2025. Stack Overflow Is Not Dead Yet: Crowd Answers Still Matter. *arXiv preprint arXiv:2509.05879* (2025).
[33] Felipe Hoffa. 2019. *Making Sense of the Metadata: Clustering 4,000 Stack Overflow tags with BigQuery k-means.* https://stackoverflow.blog/2019/07/24/making-sense-of-the-metadata-clustering-4000-stack-overflow-tags-with-bigquery-k-means/
[34] Katja Hofmann, Lihong Li, Filip Radlinski, et al. 2016. Online evaluation for information retrieval. *Foundations and Trends® in Information Retrieval* 10, 1 (2016), 1–117.
[35] Bo Jiang, Ziyan Zhang, Doudou Lin, Jin Tang, and Bin Luo. 2019. Semi-supervised learning with graph learning-convolutional networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11313–11320.
[36] Mehdi Kargar, Lukasz Golab, Divesh Srivastava, Jaroslaw Szlichta, and Morteza Zihayat. 2020. Effective keyword search over weighted graphs. *IEEE Transactions on Knowledge and Data Engineering* 34, 2 (2020), 601–616.
[37] Pranav Kasela, Gabriella Pasi, and Raffaele Perego. 2023. SE-PEF: a Resource for Personalized Expert Finding. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*. 288–309.
[38] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems* 30 (2017).
[39] Abeer Khan, Lukasz Golab, Mehdi Kargar, Jaroslaw Szlichta, and Morteza Zihayat. 2020. Compact group discovery in attributed graphs and social networks. *Information Processing & Management* 57, 2 (2020), 102054.
[40] Yue Kou, Yingxuan Du, Derong Shen, Xiangmin Zhou, Dong Li, Tiezheng Nie, and Ge Yu. 2025. Experts2team: Task Relevance-Induced Team Formation by Combining Global Cohesion with Local Decoupling. In *Conference: the 30th International Conference on Database Systems for Advanced Applications (DASFAA 2025)*. 1–16.
[41] Walid Krichene and Steffen Rendle. 2020. On sampled metrics for item recommendation. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 1748–1757.
[42] Walid Krichene and Steffen Rendle. 2020. On Sampled Metrics for Item Recommendation. In *KDD 2020*. https://dl.acm.org/doi/10.1145/3394486.3403226
[43] Vaibhav Krishna and Nino Antulov-Fantulin. 2023. Temporal-Weighted Bipartite Graph Model for Sparse Expert Recommendation in Community Question Answering. In *Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization*. 156–163.
[44] Dipankar Kundu and Deba Prasad Mandal. 2019. Formulation of a hybrid expertise retrieval system in community question answering services. *Applied Intelligence* 49 (2019), 463–477.

[45] Dipankar Kundu, Rajat Kumar Pal, and Deba Prasad Mandal. 2019. Finding active experts for question routing in community question answering services. In *Pattern Recognition and Machine Intelligence: 8th International Conference, PReMI 2019, Tezpur, India, December 17-20, 2019, Proceedings, Part II*. Springer, 320–327.
[46] Dipankar Kundu, Rajat Kumar Pal, and Deba Prasad Mandal. 2020. Preference enhanced hybrid expertise retrieval system in community question answering services. *Decision Support Systems* 129 (2020), 113164.
[47] Dipankar Kundu, Rajat Kumar Pal, and Deba Prasad Mandal. 2021. Time-aware hybrid expertise retrieval system in community question answering services. *Applied Intelligence* 51, 10 (2021), 6914–6931.
[48] Dipankar Kundu, Rajat Kumar Pal, and Deba Prasad Mandal. 2021. Topic sensitive hybrid expertise retrieval system in community question answering services. *Knowledge-Based Systems* 211 (2021), 106535.
[49] Theodoros Lappas, Kun Liu, and Evimaria Terzi. 2009. Finding a team of experts in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 467–476.
[50] Long T Le and Chirag Shah. 2018. Retrieving people: Identifying potential answerers in community question-answering. *Journal of the association for information science and technology* 69, 10 (2018), 1246–1258.
[51] Longzhuang Li, Yi Shang, and Wei Zhang. 2002. Improvement of HITS-based algorithms on web documents. In *Proceedings of the 11th international conference on World Wide Web*. 527–535.
[52] Zeyu Li, Jyun-Yu Jiang, Yizhou Sun, and Wei Wang. 2019. Personalized question routing via heterogeneous network embedding. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 192–199.
[53] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference*. 689–698.
[54] Shangsong Liang. 2019. Unsupervised Semantic Generative Adversarial Networks for Expert Retrieval. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, Ling Liu, Ryen W. White, Amin Mantrach, Fabrizio Silvestri, Julian J. McAuley, Ricardo Baeza-Yates, and Leila Zia (Eds.). ACM, 1039–1050. doi:10.1145/3308558.3313625
[55] Hongtao Liu, Zhepeng Lv, Qing Yang, Dongliang Xu, and Qiyao Peng. 2022. Efficient Non-sampling Expert Finding. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 4239–4243.
[56] Hongtao Liu, Zhepeng Lv, Qing Yang, Dongliang Xu, and Qiyao Peng. 2022. Expertbert: Pretraining expert finding. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 4244–4248.
[57] Tie-Yan Liu. 2009. Learning to Rank for Information Retrieval. *Found. Trends Inf. Retr.* 3, 3 (mar 2009), 225–331. doi:10.1561/1500000016
[58] Xiaoming Liu, Johan Bollen, Michael L Nelson, and Herbert Van de Sompel. 2005. Co-authorship networks in the digital library research community. *Information processing & management* 41, 6 (2005), 1462–1480.
[59] Yue Liu, Weize Tang, Zitu Liu, Lin Ding, and Aihua Tang. 2022. High-quality domain expert finding method in CQA based on multi-granularity semantic analysis and interest drift. *Information Sciences* 596 (2022), 395–413.
[60] Claudio Lucchese, Franco Maria Nardini, Salvatore Orlando, Raffaele Perego, and Alberto Veneri. 2022. ILMART: Interpretable Ranking with Constrained LambdaMART. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2255–2259.
[61] James MacQueen et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Vol. 1. Oakland, CA, USA, 281–297.
[62] Sara Mumtaz, Carlos Rodriguez, and Boualem Benatallah. 2019. Expert2vec: Experts representation in community question answering for question routing. In *Advanced Information Systems Engineering: 31st International Conference, CAiSE 2019, Rome, Italy, June 3–7, 2019, Proceedings 31*. Springer, 213–229.
[63] Arash Dargahi Nobari, Mahmood Neshati, and Sajad Sotudeh Gharebagh. 2020. Quality-aware skill translation models for expert finding on StackOverflow. *Information Systems* 87 (2020), 101413.
[64] Qiyao Peng, Hongtao Liu, Zhepeng Lv, Qing Yang, and Wenjun Wang. 2023. Contrastive Pre-training for Personalized Expert Finding. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.
[65] Qiyao Peng, Hongtao Liu, Yinghui Wang, Hongyan Xu, Pengfei Jiao, Minglai Shao, and Wenjun Wang. 2022. Towards a multi-view attentive matching for personalized expert finding. In *Proceedings of the ACM Web Conference 2022*. 2131–2140.
[66] Qiyao Peng, Wenjun Wang, Hongtao Liu, Yinghui Wang, Hongyan Xu, and Minglai Shao. 2022. Towards comprehensive expert finding with a hierarchical matching network. *Knowledge-Based Systems* 257 (2022), 109933.
[67] Lingfei Qian, Jian Wang, Hongfei Lin, Bo Xu, and Liang Yang. 2022. Heterogeneous information network embedding based on multiperspective metapath for question routing. *Knowledge-Based Systems* 240 (2022), 107842.
[68] Lingfei Qian, Jian Wang, Hongfei Lin, Liang Yang, and Yu Zhang. 2022. Multi-hop interactive attention based classification network for expert recommendation. *Neurocomputing* 488 (2022), 436–443.
[69] Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* 20 (1987), 53–65.
[70] Pradeep Kumar Roy and Jyoti Prakash Singh. 2024. Early prediction of promising expert users on community question answering sites. *International Journal of System Assurance Engineering and Management* (2024), 1–12.

[71] Pradeep Kumar Roy, Jyoti Prakash Singh, and Amitava Nag. 2018. Finding active expert users for question routing in community question answering sites. In *Machine Learning and Data Mining in Pattern Recognition: 14th International Conference, MLDM 2018, New York, NY, USA, July 15-19, 2018, Proceedings, Part II 14*. Springer, 440–451.

[72] Aida Sanatizadeh, Yingda Lu, Keran Zhao, and Yuheng Hu. 2025. Engagement or entanglement? The dual impact of generative artificial intelligence in online knowledge exchange platforms. *Information & Management* (2025), 104178.

[73] Lei Sang, Min Xu, ShengSheng Qian, and Xindong Wu. 2019. Multi-modal multi-view Bayesian semantic embedding for community question answering. *Neurocomputing* 334 (2019), 44–58.

[74] Anna Sapienza, Palash Goyal, and Emilio Ferrara. 2019. Deep neural networks for optimal team composition. *Frontiers in big Data* 2 (2019), 14.

[75] Guy Shani and Asela Gunawardana. 2011. Evaluating recommendation systems. *Recommender systems handbook* (2011), 257–297.

[76] Nickolay Smirnov. 1948. Table for estimating the goodness of fit of empirical distributions. *The annals of mathematical statistics* 19, 2 (1948), 279–281.

[77] Soroosh Sorkhani, Roohollah Etemadi, Amin Bigdeli, Morteza Zihayat, and Ebrahim Bagheri. 2022. Feature-based question routing in community question answering platforms. *Information Sciences* 608 (2022), 696–717.

[78] Jiankai Sun, Bortik Bandyopadhyay, Armin Bashizade, Jiongqian Liang, P. Sadayappan, and Srinivasan Parthasarathy. 2019. ATP: Directed Graph Embedding with Asymmetric Transitivity Preservation. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, 265–272. doi:10.1609/aaai.v33i01.3301265

[79] Jiankai Sun, Bortik Bandyopadhyay, Armin Bashizade, Jiongqian Liang, P Sadayappan, and Srinivasan Parthasarathy. 2019. Atp: Directed graph embedding with asymmetric transitivity preservation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 265–272.

[80] Jiankai Sun, Sobhan Moosavi, Rajiv Ramnath, and Srinivasan Parthasarathy. 2018. QDEE: Question Difficulty and Expertise Estimation in Community Question Answering Sites. In *Proceedings of the Twelfth International Conference on Web and Social Media, ICWSM 2018, Stanford, California, USA, June 25-28, 2018*. AAAI Press, 375–384. https://aaai.org/ocs/index.php/ICWSM/ICWSM18/paper/view/17854

[81] Jiankai Sun, Sobhan Moosavi, Rajiv Ramnath, and Srinivasan Parthasarathy. 2018. QDEE: question difficulty and expertise estimation in community question answering sites. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 12.

[82] Jiankai Sun, Jie Zhao, Huan Sun, and Srinivasan Parthasarathy. 2021. Endcold: An end-to-end framework for cold question routing in community question answering services. In *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence*. 3244–3250.

[83] Weizhao Tang, Tun Lu, Dongsheng Li, Hansu Gu, and Ning Gu. 2020. Hierarchical attentional factorization machines for expert recommendation in community question answering. *IEEE Access* 8 (2020), 35331–35343.

[84] Rohan Tondulkar, Manisha Dubey, and Maunendra Sankar Desarkar. 2018. Get me the best: predicting best answerers in community question answering sites. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 251–259.

[85] Xuchao Zhang, Wei Cheng, Bo Zong, Yuncong Chen, Jianwu Xu, Ding Li, and Haifeng Chen. 2020. Temporal Context-Aware Representation Learning for Question Routing. In *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, James Caverlee, Xia (Ben) Hu, Mounia Lalmas, and Wei Wang (Eds.). ACM, 753–761. doi:10.1145/3336191.3371847

A Appendix

A.1 StackExchange communities

In Section 4.1, we statistically demonstrate how the selected Stack Exchange communities differ by applying the non-parametric Kolmogorov–Smirnov (KS) test [2, 76], which assesses whether two samples derive from the same underlying distribution. To support our analysis, Table 9 and Table 10 in this appendix summarize the *question-level* and *user-level* features, respectively, for each community.

Table 9 focuses on question-level characteristics, including the average length of a question's body and title, the average number of tags, answers, and comments, the average score, and the percentage of solved questions (i.e., those with an accepted answer). Instead, Table 10 highlights user-level metrics such as average reputation, views, upvotes, and downvotes, as well as the average number of answers and accepted answers contributed.

Table 9. Average number of body and title characters, tags, answers, comments, score, and the percentage of solved questions (closed questions) for each community.

Community	AvgBody	AvgTitle	AvgTags	AvgAnswers	AvgComments	AvgScore	%Solved
Stack Overflow	1587.99	55.07	2.97	1.48	1.99	2.29	49.16%
Unix	1288.55	53.38	2.85	1.47	2.03	3.97	52.50%
Ask Ubuntu	1208.11	52.01	2.80	1.26	2.03	3.30	67.13%
Server Fault	1366.45	53.75	2.90	1.58	1.45	2.18	52.80%
Physics	1019.77	59.54	3.28	1.46	2.37	2.54	57.23%
Mathematics	953.87	60.82	2.41	1.31	2.48	1.89	47.68%

Table 10. Average reputation score, views, upvotes, downvotes, answers, and accepted answers for each community.

Community	AvgReputation	AvgViews	AvgUpVotes	AvgDownVotes	AvgAnswers	AvgAccepted
Stack Overflow	598.07	70.24	57.32	7.42	11.35	3.91
Unix	383.38	29.44	24.00	2.18	5.03	1.62
Ask Ubuntu	237.15	35.78	16.32	2.10	3.48	0.91
Server Fault	246.64	22.26	15.50	2.39	5.06	1.52
Physics	459.80	106.64	40.25	8.25	9.79	2.87
Mathematics	843.65	182.58	79.07	11.07	20.97	8.37

A.2 Parameters evaluation

Table 11 presents, for each parameter, the value tested, the metrics performance, the inference time (in minutes), and the average length of the experts' list ranked by the LtR model. The underlined rows indicate the selected values, and we report the results of the paired t-test with a p-value < 0.05 and Bonferroni correction, demonstrating statistically significant differences compared to alternative values.

In general, we prioritized values that maximized the performance metrics; however, three exceptions were made. For K, we chose a smaller MLG, as increasing K did not yield a significant improvement in performance but did increase the complexity of the experts' list. Moreover, we preferred a higher p value which creates a smaller list of experts keeping the same performance of lower values. Ideally, an Expert Finding model should rank all potential answerers to avoid the *cold-start* problem. However, this is challenging due to scalability issues. Pre-processing techniques that filter candidates based on activity levels are commonly used to reduce complexity and focus on more active users. In this case, the ω parameter controls the number of selected experts. Higher values result in a smaller, more specialized set, improving performance by focusing on the most relevant candidates and reducing noise from less qualified ones. A smaller expert set also decreases the number of evaluable queries in the test set, as only questions answered by the selected experts are considered. This enhances both performance metrics and computational efficiency. For this reason, we preferred the 95^{th} percentile for ω, as it balances a smaller expert set with sufficient coverage for evaluation. Finally, the number of restarts and steps control the graph exploration process. Since varying the number of restarts resulted in similar performance metrics, we selected five restarts to maintain high performance while keeping the experts' list length concise.

From Table 11, we can observe the sensitivity of the model's performance to each parameter. Overall, the graph exploration parameters (i.e., p, *restarts*, and *steps*) have a smaller impact on performance. In contrast, the other parameters can lead to an average performance variation of approximately 4%, with ω and δ showing the most significant impact, reaching differences of up to 11%.

Table 11. Evaluation of parameter configurations. For each parameter, we report the tested values along with performance metrics (P@1, NDCG@3, R@5, MRR), inference time (in minutes), and the average length of the experts' lists ranked by the LtR model. Underlined rows indicate the selected values. Statistical significance (paired t-test with Bonferroni correction, $p < 0.05$) is marked with (significantly higher) or (significantly lower) compared to the chosen value.

			λ			
Value	P@1	NDCG@3	R@5	MRR	Time (min)	AvgExpsLists
5	0.455	0.600	0.789	0.599	2.22	52.87
<u>10</u>	<u>0.491</u>	<u>0.627</u>	<u>0.809</u>	<u>0.625</u>	<u>2.31</u>	<u>52.06</u>
15	0.488	0.621	0.809	0.622	1.97	37.72
20	0.452	0.588	0.775	0.589	1.97	40.43
			K			
Value	P@1	NDCG@3	R@5	MRR	Time (min)	AvgExpsLists
<u>10</u>	<u>0.486</u>	<u>0.625</u>	<u>0.805</u>	<u>0.623</u>	<u>2.31</u>	<u>52.00</u>
14	0.454	0.603	0.792	0.600	2.24	50.72
19	0.489	0.628	0.809	0.627	2.22	49.41
25	0.470	0.618	0.801	0.615	2.25	47.76

30	0.450	0.588	0.790	0.594	2.30	48.61

ω

Value	P@1	NDCG@3	R@5	MRR	Time (min)	AvgExpsLists
90	0.626	0.748	0.878	0.738	1.26	87.80
95	0.657	0.779	0.894	0.763	1.12	51.92
99	0.738	0.837	0.932	0.826	0.93	14.96

ϵ

Value	P@1	NDCG@3	R@5	MRR	Time (min)	AvgExpsLists
1	0.468	0.613	0.803	0.609	2.51	32.41
3	0.492	0.629	0.809	0.628	2.26	52.00
5	0.472	0.618	0.819	0.619	2.30	63.02
7	0.448	0.584	0.777	0.588	2.24	71.56

δ

Value	P@1	NDCG@3	R@5	MRR	Time (min)	AvgExpsLists
0.1	0.463	0.612	0.810	0.613	2.68	58.84
0.3	0.463	0.596	0.785	0.604	2.67	59.74
0.5	0.491	0.628	0.800	0.627	2.32	52.00
0.7	0.460	0.594	0.774	0.594	2.30	40.65
0.9	0.383	0.505	0.663	0.498	2.07	24.03

p

Value	P@1	NDCG@3	R@5	MRR	Time (min)	AvgExpsLists
0.05	0.471	0.607	0.787	0.605	2.07	29.43
0.01	0.466	0.609	0.796	0.604	2.17	39.60
0.005	0.479	0.620	0.803	0.618	2.26	43.32
0.001	0.486	0.625	0.805	0.623	2.23	52.00
0.0005	0.482	0.625	0.808	0.623	2.53	54.30
0.0001	0.487	0.626	0.819	0.625	2.35	60.77

restarts						
Value	P@1	NDCG@3	R@5	MRR	Time (min)	AvgExpsLists
1	0.488	0.627	0.800	0.622	2.05	25.08
5	0.486	0.625	0.805	0.623	2.28	52.00
10	0.477	0.620	0.811	0.621	2.72	69.55
15	0.488	0.627	0.814	0.627	3.01	77.58
steps						
Value	P@1	NDCG@3	R@5	MRR	Time (min)	AvgExpsLists
5	0.461	0.612	0.800	0.607	2.10	38.97
10	0.489	0.632	0.799	0.627	2.28	52.00
15	0.466	0.608	0.806	0.610	2.41	64.94
20	0.487	0.626	0.807	0.625	2.54	72.92

Received 9 June 2024; revised 29 September 2025; accepted 8 January 2026