International Journal of Innovation and Multidisciplinary Research
DOI.ORG/10.67505/IJIAMR.v6.i2.62007

Vol. 6 (2), 2026
E-ISSN: 2583-4452

Semantic-Based Expert Identification in Community Question Answering Platforms

Iram Fatima[1*], Ms. Seema[2]

[1] Graduate Trainee, HCL Tech, Noida, Uttar Pradesh

[2] Department of Computer Science, Shaheed Rajguru College of Applied Sciences for Women, University of Delhi

[*] Corresponding Author ✉ siddiquiram022@gmail.com

Abstract

Community Question Answering (CQA) platforms provides opportunity for users to ask and answer questions on variety of domains. Identifying the experts for a new question being posted on these platforms has become difficult as the number of users and content are increasing in large number. Semantic approaches play a crucial role by capturing contextual meaning between questions which enables more accurate matching of queries to suitable experts beyond simple keyword overlap. Traditional approaches like TF-IDF and BM25 techniques provide lexical keyword matching only and fail to show the semantic meaning of technical queries. This paper explores the effects of different textual representation strategies on expert identification by comparing five different retrieval-based techniques on a Stack Overflow dataset within a similar retrieval and selection framework to maintain uniformity and fairness. Two lexical keyword-based (TF-IDF and BM25) and three dense embedding-based models such as Sentence BERT (all-mpnet-base-v2), BGE-large (BAAI/bge-large-en), and CodeBERT have been compared. To evaluate the performance various metrics such as Hit Rate, Mean Reciprocal Rank (MRR) and Normalised Discounted Cumulative Gain (nDCG) have been used. The results indicate that dense embedding models have performed better than traditional keyword methods. Out of all, BGE-large has achieved the highest performance and showed strong semantic discrimination in expert ranking. CodeBERT has shown lower performance and indicates that models pre-trained on code do not provide an advantage for question-question matching in CQA. These results indicate the significance of dense semantic embeddings for expert identification in CQA systems.

Keywords: *Expert Finding, Semantic matching, BGE-large, Sentence Bert, TF-IDF*

1. Introduction

Community Question Answer (CQA) platforms like Quora, Yahoo, Stack Overflow are the source of knowledge platforms in which users ask and answer questions (Zhao et al., 2015). As these platforms are growing, it has become a challenging task to find the appropriate expert to a newly posted question. Identifying domain experts among diverse set of users is a fundamental research problem that links information seekers with high-quality answers (Seema & Banati, 2025). Performance of expert identification contributes to the improvement of the quality of responses, reduces the number of unanswered questions and increases community interactions.

Most of the traditional expert identification methods are based on lexical information (i.e. surface-level word features such as term frequency and keyword overlap) retrieval techniques like Term-Frequency-Inverse Document Frequency (TF-IDF) (Salton & Buckley, 1988), BM25 (Robertson & Walker, 1994) ,Query Likelihood Language Model (Balog et al., 2006) and Language Modelling with Dirichlet Smoothing (Zhai & Lafferty, 2001), These methods measure a similarity between a new question and previously answered questions in terms of keyword overlapping. Although it is effective for keyword-based matching, it fails to capture the deeper semantic meaning between questions specifically in technical domains in which conceptually similar questions can have different keywords.

The recent developments in transformer-based language models have enabled the development of dense semantic representations that capture contextual meaning beyond keyword overlap. Models like Sentence BERT (SBERT) (Reimers & Gurevych, 2019) and BGE large (Xiao et al., 2024) utilized contrastive training objectives to generate dense embeddings that are optimized for semantic similarity and retrieval tasks. Additionally, there has been the development of programming-based models such as CodeBERT to represent natural language and source code together (Feng et al., 2020). Since platforms like Stack Overflow contains large number of code samples, it is still needed to explore whether code-pretrained representations can provide better results in expert identification when compared to semantic embedding models. These developments CQA platforms remain the area to explore.

In this study, we conducted a comparative study of different semantic matching models for expert identification in CQA platforms. All the models are implemented in a single retrieval framework based on question to question similarity and aggregation of model at the expert level. Temporal hold-out split is employed to measure the ranking performance of new queries and then it is evaluated using HitRate@k, MRR@k, and nDCG@k to measure the effectiveness of ranking. The main

Received on: 26 September 2026 | Accepted on: 30 October 2026 | Published Online on: 20 November 2026

© 2026 The Author(s). This article is an open access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0).

contributions of this paper are:

1. An integrated comparative evaluation of lexical, dense semantic, and code-aware models for expert identification.
2. A large-scale temporal hold-out evaluation protocol for evaluating expert ranking performance on new queries.
3. Experimental study shows that dense semantic embeddings are better than lexical sparse methods in terms of expert identification.

2. Related Work

The identification of experts has been extensively studied in the information retrieval field. Approaches like probabilistic language modelling in document collections for expert finding has been used. Their study showed that the document-centric approach performed better than candidate-centric approach, which highlight the matching of documents with experts (Balog et al., 2006). Other studies have also focused on examination of topic modelling approaches like Latent Dirichlet Allocation (LDA) to expert profiling and graph-based authority estimation methods in CQA platforms (Jurczyk & Agichtein, 2007; Riahi et al., 2012). Traditional lexical retrieval methods such as TF-IDF (Salton & Buckley, 1988) and BM25 (Robertson & Walker, 1994) have been applied to new queries by matching it with previously answered queries. However, their keyword based similarity are not accurate to capture semantic relationship between queries. To resolve this challenge, dual-encoder dense retrieval models like DPR (Karpukhin et al., 2020) and Contriever (Izacard et al., 2022) were proposed, which learned query and passage representations independently, and interaction models like Colbert (Khattab & Zaharia, 2020) were introduced which added token level matching for finer-grained retrieval.

Dense semantic embeddings have proposed the introduction of deep learning specifically Sentence-BERT (SBERT)(Reimers & Gurevych, 2019) models which are very effective at modelling similarity (Ha et al., 2021), and recently contrastively trained models like BGE (Xiao et al., 2024) have been used to further enhance dense-retrieval performance. More recently, retrieval-augmented generation (RAG)(Lewis et al., 2020) and LLM-assisted retrieval approaches(Shi et al., 2024) have been explored for knowledge-intensive tasks, but their computational demands limit direct applicability to large-scale CQA expert finding. Code-aware models like CodeBERT which builds on representation learning but operate on natural language and source code (Feng et al., 2020). Despite of these developments, many existing approaches combine textual, graph-based and reputation signals which makes it difficult to determine the impact of a representation decision on retrieval performance. Therefore, a systematic comparative analysis of lexical, dense semantic and code-oriented transformer models in a single expert identification framework is limited. This study fills this gap by performing a comparative study of these representations under a uniform experimental setting.

3. Problem Formulation

For expert identification problem in CQA platforms, researchers often combined textual similarity features with graph-based and reputation features. These techniques have performed effectively but they make it difficult to identify the influence of representation of texts on expert ranking.

Let $Q = \{q_1, q_2, ..., ..., q_n\}$ represents the set of questions and $U = \{u_1, u_2, ..., ..., u_n\}$ represents the set of candidate experts. In order to identify experts for a new query question q, the task is to rank users in U based on their relevancy to the query. In theory, the objective is to learn a ranking function

$$f(q, u) = s(q, u) \quad (1)$$

where $s(q, u)$ is the score of relevance of expert u for question q and $f(q, u)$ is the ranking function's output. The experts are then ranked by the decreasing order $s(q, u)$ to generate a list of ranked candidate experts.

In this study, a dataset from Stack Overflow has been used. Stack Overflow is a community question answering platform where user can ask and answer questions in technical domains and it also provides knowledge sharing among users worldwide. In Stack overflow, an expert is the user who had answered the question on Stack Overflow with an accepted answer, as found in the AcceptedAnswerId in the data dump. There was no use of reputation scores, badges or manual annotation for the labeling process. This approach is not based on direct user profile modelling but rather on similarity based on question-to-question similarity as a proxy for identification by experts. Platform-based approaches equate platform activity with query-specific expertise, a highly reputed user might not have the expertise needed for a specific query. Question similarity-based retrieval, on the other hand, is based on explicit acceptance judgments from the question asker, not on the less reliable signal of upvotes.

Therefore, there is a limitation in the comparison study between lexical, dense semantic and code-oriented transformer models in a unified framework. To address this gap, our experimental design has used different textual representation models which are evaluated using the same candidate generation as well as expert-level aggregation, and a strategy of temporal validation protocols. This study enabled to measure the effect of the choice of representation on the expert ranking.

4. Research Methodology and Experimentation

This paper explores the strategy of text representation influence on expert identification in CQA platforms. A retrieval based experimental framework is developed to compare the different representation models. The experimentation begins by using the stack overflow dataset which was first pre processed and converted to textual representation by different approaches such as lexical, dense embedding and code-oriented models. Each query question is then retrieved by the system using the similar historical questions, which are then mapped to the accepted answerers and similar scores are aggregated at the expert level. The

obtained experts are ranked based on aggregated semantic evidence and the performance of each representation model is evaluated using standard ranking metrics like HitRate@K, MRR@K, nDCG@K. The overall architecture of this framework is illustrated in Figure 1.

The step-by-step procedure of retrieval based expert identification pipeline is mentioned in the pseudocode below:

Input:
Q = Set of historical questions
A = Set of answers with accepted answerers
M = Representation model (TF-IDF / BM25 / SBERT / BGE-large / CodeBERT)
K = anking depth (number of top-ranked experts considered for evaluation)
Output:
Ranked list of candidate experts for each query question
1. Preprocess dataset
2. **For** each question q in Q do
   Construct textual representation:
   text_q = Title + Tags + Body
   End For
3. Build question representations using model M
   If M == TF-IDF then
   Compute sparse vector representations
   Else if M == BM25 then
   Build BM25 inverted index
   Else
   Encode text_q into dense embeddings
   Apply L2 normalization
   End If
4. Build retrieval index over all question representations
5. Create mapping:
   QuestionID → AcceptedExpertID
6. **For** each query question q_i in Q do
   Retrieve top-(K+1) similar questions
   Remove self-match
   For each retrieved question q_j do
   expert ← AcceptedExpert(q_j)
   Store similarity score for expert
   End For
   For each expert u do
   Aggregate similarity scores:
   SemSim_max
   SemSim_mean
   SemSim_sum
   Frequency count
   Threshold-based count ≥ 0.60
   End For
   Rank experts by SemSim_max (descending)
   End For
7. Evaluate ranking using:
   HitRate@K, MRR@K, nDCG@K

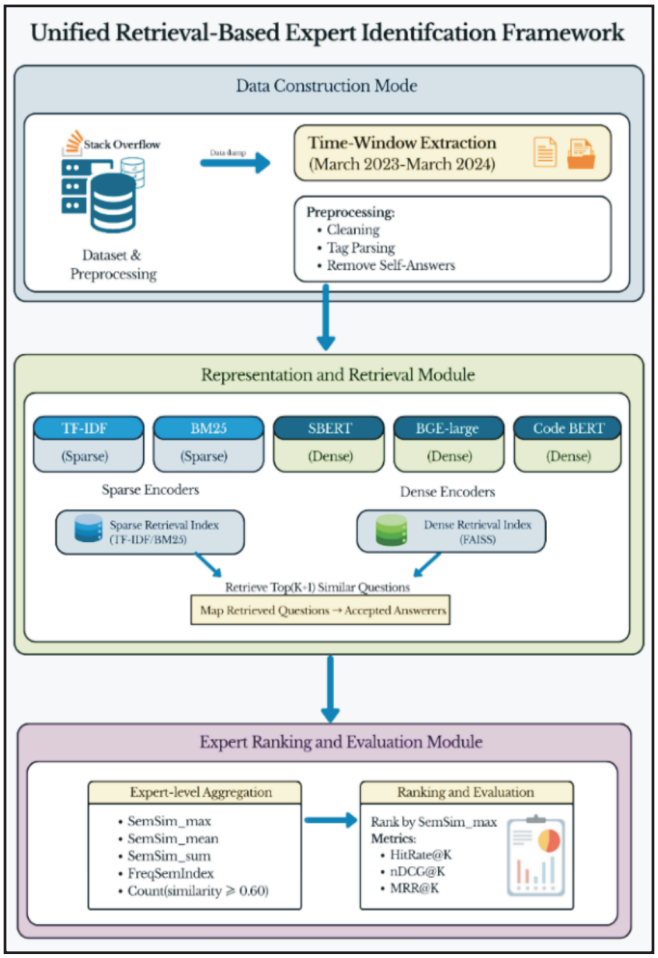


Figure. 1: *Architectural Diagram*

4.1 Dataset Description and Preprocessing

A subset of the public data dump from Stack Overflow (from March 2023 -March 2024) is used for experimentation. The raw dataset comprised 876,079 questions, 888,251 answers, and 22,484,235 registered users. There were 195,238 questions, 258,939 answers and 169,585 unique users after the pre-processing. An average of 3 tags per question (min: 1, max: 5) exist in each question, which include Title, Tags, Body, CreationDate, and AcceptedAnswerId. The dataset covers multiple technical domains including Python (15,329), JavaScript (9,605), R (5,152), C# (5,007), Java (4,218), and CSS (3,564). Expert labels are created by performing a join on the AcceptedAnswerId and answers tables to obtain the OwnerUserId of the answerer who accepted the answer, resulting in one ground-truth expert per question from the perspective of the question asker themselves. Total of 155,807 questions were mapped to an expert label. Questions are sorted by CreationDate and used an 80/20 split. There are 156,190 questions and 208,246 answers in the training set, and 39,048 questions and 50,693 answers in the test set, representing 142,259 users in total, with evaluation carried out solely on future questions to prevent the leakage of data.

Semantic Retrieval Framework

Expert Identification is defined as a ranking problem based on the similarity of questions.

Let:

$Q = \{q_1, q_2, ..., ..., q_n\}$ are the set of questions,

$U = \{u_1, u_2, ..., ..., u_n\}$ are the set of users

The representation of each question q is the concatenation of Title, Tags and Body. In transformer-based models special separator is used in order to maintain structural boundaries. For a given query question q_{new}:

1. Retrieve the top-300 most similar historical questions using the model-specific retrieval mechanism.

2. Remove the query itself from the retrieved set and map retrieved questions to their accepted answerers.
3. Aggregate similarity scores at the expert level and then rank experts based on maximum similarity score (SemSim_max)

All compared techniques use this identical retrieval and aggregation pipeline to ensure fair comparison.

4.2 Compared Approaches

To evaluate the effect of semantic representation on expert identification in Stack Overflow, a comparative study of five different retrieval models is conducted in which all models have the same candidate generation and aggregation in order to ensure fairness.

4.3.1 Lexical Baselines:

We have implemented TF-IDF (Salton & Buckley, 1988) and BM25(Robertson & Walker, 1994) keyword based lexical baselines.

4.3.1.1 TF-IDF:

The vectorizer is set with vocabulary size of 20,000 terms.

- N-gram range of (1,2)
- Minimum and maximum document frequencies of 2 and 0.85 respectively
- A standard English stop-word removal.

Cosine similarity is calculated through effective linear kernel processes.

4.3.1.2 BM25:

The top-300 most similar questions are retrieved from the tokenized corpus. Then, those similar questions are searched with a standard BM25 retrieval engine, as it naturally accounts for saturation of term frequency and document length normalization.

4.3.2 Dense Semantic Models:

To capture the contextual semantic relationship between queries, two trained dense retrieval models that are SBERT (Reimers & Gurevych, 2019) and BGE-large (Xiao et al., 2024) are evaluated.

4.3.2.1 SBERT:

All-mpnet-base-v2 SBERT model architecture is used in which questions are encoded into a dense embedding of 768 dimension.

4.3.2.2 BGE-Large:

Questions are encoded into embedding of 1024 dimension to take advantage of its powerful contrastive supervision.

For both dense models, the obtained embeddings are L2-normalized and indexed using FAISS (IndexFlatIP). The execution of retrieval is through inner-product similarity which is mathematically equivalent to cosine similarity with L2 normalization.

4.3.3 Code-Aware Representation: CodeBERT

We implemented CodeBERT to test the effectiveness of joint natural language and source code representations of experts (Feng et al., 2020). Since CodeBERT does not produce any sentence embeddings directly, we get the token embeddings from the last hidden layer and apply mean pooling over tokens to obtain fixed-length embeddings. The maximum sequence length is the same as for the dense models and is set to 512, after which the model is unable to process the tokens and the tokenizer truncates them. Compared to SBERT and BGE-large, which explicitly used contrastive objectives to learn sentence-level embeddings that are coherent across the whole sentence, the token representations of CodeBERT were trained to align tokens with the code-text input, and mean pooling was a less direct approximation of semantic similarity. Similar to the dense models, these embeddings are also L2-normalized and indexed using FAISS (IndexFlatIP) for similarity search.

4.3 Expert-Level Aggregation

For each query, similarity scores of retrieved questions that have been previously answered by expert u are aggregated using the following features: maximum similarity score (SemSim_max), mean similarity score (SemSim_mean), sum of similarity scores (SemSim_sum), frequency of matched questions (FreqSemIndex), threshold-based count of similarities above 0.60.

All five features are calculated, but the final ranking is done using SemSim_max, because SemSim_max is based only on the strongest relevance signal, unlike features based on frequency, which favour users with high answer volumes regardless of query relevance, or aggregation based on mean, which is sensitive to low-similarity noise in the candidate pool.

In order to maintain consistency between methods, the candidate pool is restricted to top-300 experts per query.

4.4 Evaluation Metrics

The evaluation of models is based on performance using standard ranking-based metrics widely used in expert identification and information retrieval tasks. Given that there is one accepted expert per query, binary relevance is assumed.

Let:

- Q denote the set of evaluation queries
- K denotes the cut-off rank
- $rel_i \in \{0,1\}$ denote the relevance level or rank i
- $rank_q$ denote the position of the accepted expert for query q

4.4.1 HitRate@K

HitRate@K measures the presence of correct (accepted) expert in the top-K ranked experts for a given query (Deshpande & Karypis, 2004).

For a single query:

$$HitRate(q) = \begin{cases} 1, & if\ accepted\ \text{exp}ert\ appears\ in\ top{-}K \\ 0, & otherwise \end{cases} \quad (2)$$

The overall HitRate@K is computed as:

$$HitRate@K = \frac{1}{|Q|}\sum_{q\in Q} HitRate(q) \quad (3)$$

The metric evaluates retrieval coverage but does not consider the exact ranking position within top-K.

4.4.2 Mean Reciprocal Rank (MRR@K)

MRR measures how well the relevant expert ranked at the top (Voorhees & Tice, 2000).

For a single query:

$$RR(q) = \begin{cases} \frac{1}{rank_q}, & if\ rank_q \leq K \\ 0, & otherwise \end{cases} \quad (4)$$

The Mean Reciprocal Rank is:

$$MRR@K = \frac{1}{|Q|}\sum_{q\in Q}^{R} RR(q) \quad (5)$$

MRR focuses mainly on the most recently ranked correct predictions, and assign scores to the expert who appear earlier in the ranking. It is computed with truncation at rank K.

4.4.3 Normalized Discounted Cumulative Gain (nDCG@K)

nDCG is used to measure the ranking quality in information retrieval and uses logarithmic discounting to lower ranking positions (Järvelin & Kekäläinen, 2002).

For a query q, the Normalized Discounted Cumulative Gain (nDCG) is:

$$DCG@\text{K} = \sum_{i=1}^{k} \frac{2^{rel_i - 1}}{\log_2(i+1)} \quad (6)$$

The final score is averaged across all queries:

$$nDCG@K = \frac{DCG@K}{IDCG@K} \quad (7)$$

nDCG@K captures both the number of ranks containing the correct answers and the accumulated ranking with diminishing penalty for lower ranks and IDCG@K is the DCG score for the ideal ranking.

Metrics evaluation is performed at multiple cutoff levels (K=5,10, and 20) to provide a complete performance ranking in which each value of K shows different practical expert routing situations. Lower values measures how well the system place the correct expert among the topmost position, which is essential in real-time question routing. Moderate values compare ranking consistency with relaxed cutoffs, while higher cutoffs compare broader retrieval coverage. Multiple k values allow to ensure that the conclusions are not biased by a single threshold and gives a strong comparison of model effectiveness at different recommendation depths.

5. Results and Discussion

5.1 Quantitative Results

The comparative study of TF-IDF, BM25, SBERT, BGE large, and CodeBERT is evaluated using HitRate@K, MRR@K, and nDCG@K at K=5,10,20.

The results are demonstrated in Table 1-3

Table 1: *Comparative Analysis at K = 5*

Methods	HitRate@5	MRR@5	nDCG@5
TF-IDF	0.184231	0.113011	0.130639
BM25	0.24006	0.151846	0.173733
SBERT	0.266246	0.165422	0.190429
BGE large	**0.298162**	**0.186519**	**0.214229**
CodeBERT	0.079262	0.047177	0.055094

Table 2: *Comparative Analysis at K = 10*

Methods	HitRate@20	MRR@20	nDCG@20
TF-IDF	0.320155	0.1266	0.169585
BM25	0.388309	0.166817	0.216372
SBERT	0.43287	0.182187	0.238289
BGE large	**0.470933**	**0.204105**	**0.264073**
CodeBERT	0.160542	0.055155	0.078216

Table 3: *Comparative Analysis at K = 20*

Methods	HitRate@10	MRR@10	nDCG@10
TF-IDF	0.250112	0.121747	0.151886
BM25	0.313656	0.16165	0.197516
SBERT	0.347141	0.176216	0.216587
BGE large	**0.384564**	**0.198088**	**0.242207**
CodeBERT	0.116077	0.052086	0.066995

At all cutoff levels we observed a distinct performance hierarchy. Among all the metrics, BGE large has scored the highest in all the metrics including HitRate@k, MRR@k, and nDCG@k and then followed by SBERT. BM25 and TF-IDF are showing moderate performance. On the other hand, CodeBERT has performed the lowest across all metrics when compared with other techniques.

This metrics trend clearly indicates that dense semantic embeddings are more efficient compared to lexical and code-oriented techniques for expert identification in CQA platforms.

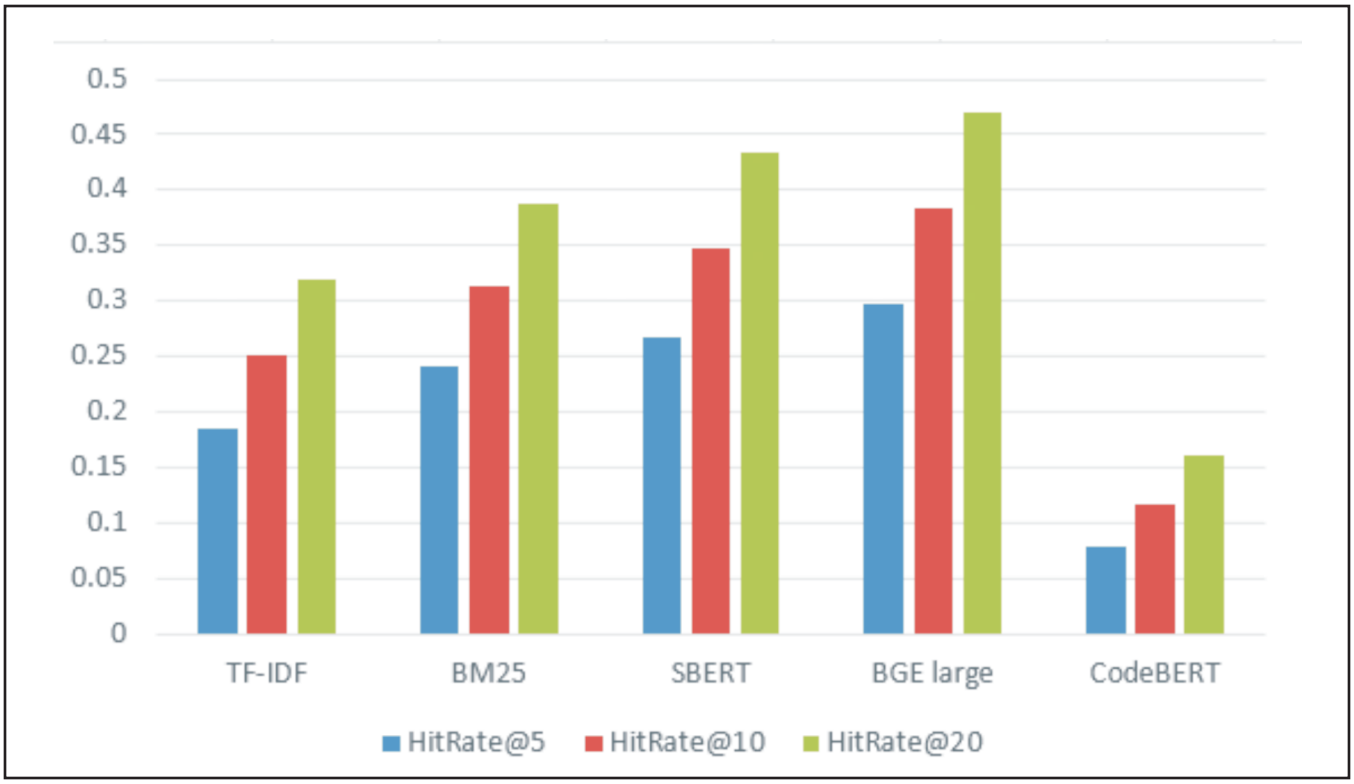


Figure. 2: *HitRate@k metrics comparison of different models for K=5,10,20*

In Figure. 2, HitRate@k metrics at all cutoff levels is performed. The ranking order is approximately same for k=5, 10 and 20, which demonstrates that the representation models are robust at different levels of recommendations. At k=20, BGE-large achieves the highest performance with a HitRate@20 of approximately 0.47, as compared to SBERT which is 0.43 and BM25 which is 0.39. This difference in metrics demonstrates the advantage of dense semantic retrieval techniques over lexical matching approaches. CodeBERT performs significantly lower across all cutoff values, indicating that it is not effective at measuring question-to-question similarity in this context.

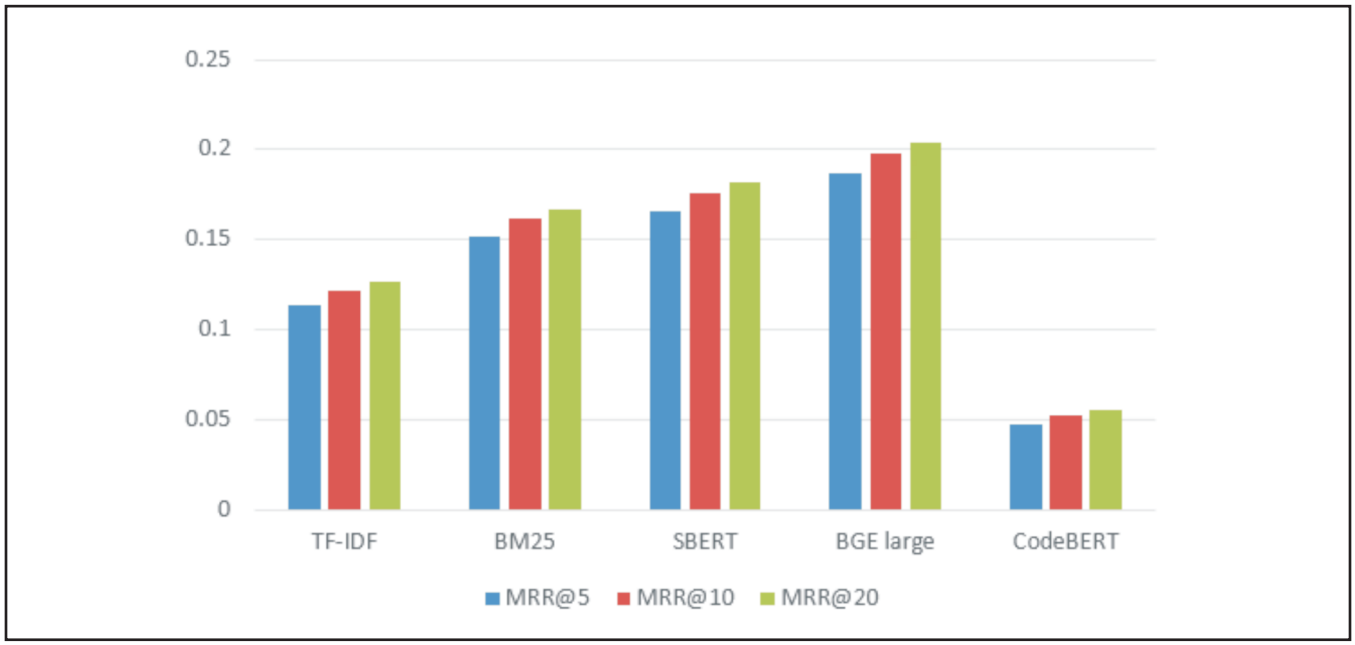


Figure 3: *MRR@k metrics comparison of different models for K=5,10,20*

In Figure. 3, MRR@k metrics is performed at different cutoff levels. At k=10, BGE large converges on approximately 0.20, followed by SBERT of approximately 0.17. These both semantic techniques outperformed BM25 and TF-IDF which indicates that dense embedding models are more effective in positioning the correct expert at higher ranks. The gap between BGE-large and lexical baselines shows that contextual semantic similarity is highly significant in expert identification.

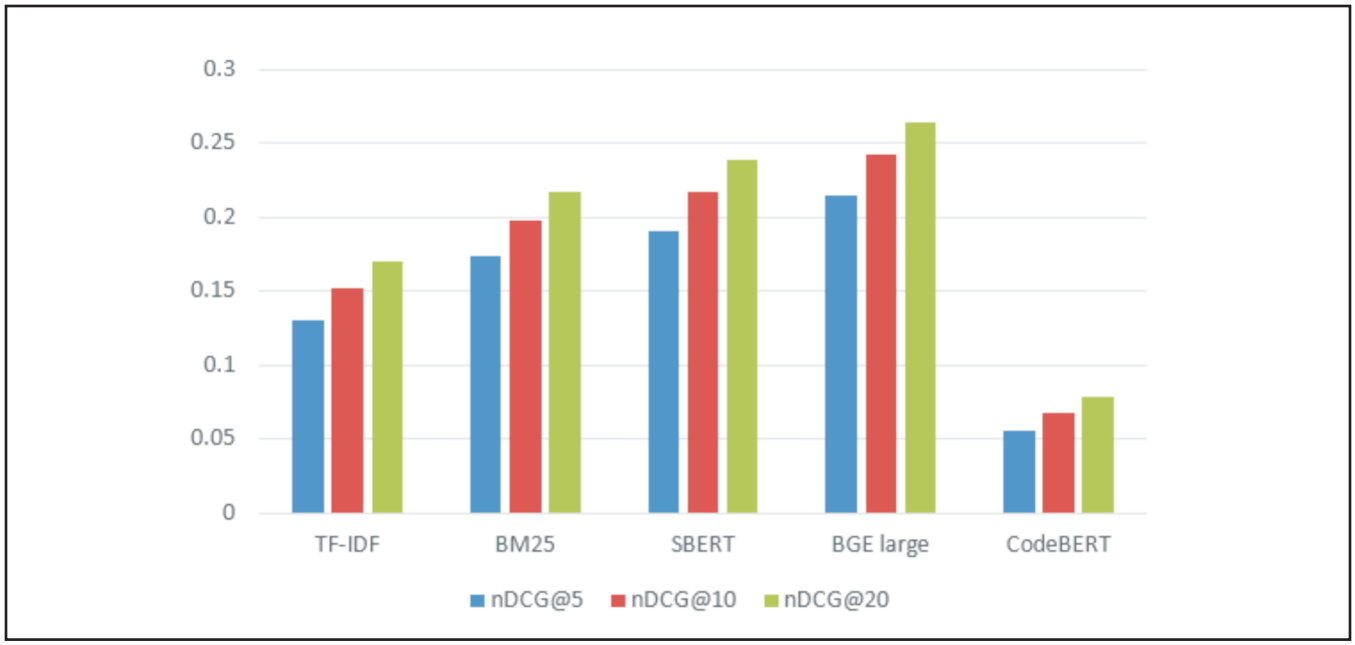


Figure 4: *nDCG@k metrics comparison of different models for K=5,10,20*

In Figure. 4, nDCG@k is performed which measures the quality of ranking at all cutoff levels. At k=20, BGE large again achieves the highest nDCG score, indicating superior ranking quality across deeper candidate lists. SBERT performs competitively, while CodeBERT remains substantially lower. The preservation of the performance hierarchy across all metrics and cutoff values suggests that the observed improvements are systematic rather than threshold-specific.

Table 4: *Pairwise Comparison of BGE-large Against Baselines (K=10, n=31,238)*

Comparison	HitRate@10	MRR@10	nDCG@10	Result
BGE-large vs TF-IDF	+0.1345	+0.0815	+0.1126	p<0.001
BGE-large vs BM25	+0.0709	+0.0363	+0.0447	p<0.001
BGE-large vs SBERT	+0.0375	+0.0219	+0.0256	p<0.001
BGE-large vs CodeBERT	+0.2685	+0.1460	+0.1752	p<0.001

The paired two-tailed t-test results in Table 4 show that all performance differences are statistically significant at K = 10 with 31,238 per-query scores analyzed for BGE-large compared to each baseline. The biggest improvement is over CodeBERT (HitRate@10: +0.2685, MRR@10: +0.1460, nDCG@10: +0.1752), which shows that it is truly underperforming and not just a fluke. As all models are deterministic, there is no variance across runs, and only variance across the 95% CI for BGE-large nDCG@10 [0.2383, 0.2461] is applicable. Tables 1–3 also show that the ranking order is similar across K = 5, 10 and 20, providing further evidence of robustness.

5.2 Discussion

The experimental results obtained clearly demonstrates that dense semantic embedding models are significantly outperforming lexical retrieval methods for expert identification in CQA platforms. While BM25 improves over TF-IDF due to probabilistic term weighting but both are limited to surface-level term matching. BGE-large consistently achieves the best performance across all metrics and cutoff levels, indicating that large-scale contrastive embedding training enhances semantic discrimination for expert retrieval. On the other hand, although CodeBERT is pretrained on natural language and source code pairs, it remains show lower performance.

This indicates that code-aware pretraining does not directly optimise the sentence-level semantic similarity for question-question retrieval in CQA queries. Overall, the trend is similar across K=5, 10, and 20, confirming the robustness of the proposed comparative framework and strengthening the conclusion that retrieval-optimized semantic embeddings are more suitable for expert identification in CQA platforms.

The poor performance of CodeBERT could be explained by the contrast between their pretraining objective and the requirement of this task. The CodeBERT is pre-trained with bimodal pairs of code and docstrings at a token-level level of code-text alignment, as opposed to a sentence-level similarity of meaning. Therefore, the mean-pooled representation from its final hidden layer is not semantically coherent like SBERT and BGE-large trained with contrastive methods. Furthermore, the answers to the questions asked on Stack Overflow are mostly in natural language, and the code is used to explain the problem, which further reduces the benefit of code-aware pretraining for code to question retrieval.

6. Conclusion

This paper compared lexical, dense semantic and code-oriented models in a single framework for expert identification in Stack Overflow. The finding demonstrates that dense embedding approaches are better than traditional lexical methods across all evaluation metrics. Specifically, BGE-large achieved the strongest performance which highlights the value of large-scale contrastive pretraining and SBERT also significantly performed higher than lexical baselines. In contrast, CodeBERT exhibits lower performance which indicates that code-aware pretraining does not directly improve retrieval performance on queries that are dominated by natural language.

Generally, these findings highlight the effectiveness of retrieval optimized dense representations for expert routing on community question answering platforms.

Future work can involve assessing integrated framework, having temporal decay of expertise signals, as well as in the context of a combination of question similarity and lightweight user activity signals. It can explore methods to identify multiple relevant experts per query and improve candidate selection. It can also integrate semantic feature with temporal and community features and identify the expert with a learning to rank model.

Conflict of Interest

Authors declares no conflict of interest.

Ethical Declarations

The authors declare that this research does not involve human participants, animals, or any procedures requiring ethical approval. All data used in this study were obtained and processed in accordance with applicable ethical and legal guidelines.

References

Data Availability: https://archive.org/download/stackexchange

1. Balog, K., Azzopardi, L., & De Rijke, M. (2006). Formal models for expert finding in enterprise corpora. *Proceedings of the Twenty-Ninth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, *2006*, 43–50. https://doi.org/10.1145/1148170.1148181
2. Deshpande, M., & Karypis, G. (2004). Item-based top-N recommendation algorithms. *ACM Transactions on Information Systems*, *22*(1), 143–177. https://doi.org/10.1145/963770.963776
3. Feng, Z., Guo, D., Tang, D., Duan, N., Feng, X., Gong, M., Shou, L., Qin, B., Liu, T., Jiang, D., & Zhou, M. (2020). CodeBERT: A Pre-Trained Model for Programming and Natural Languages. *Findings of the Association for Computational Linguistics Findings of ACL: EMNLP 2020*, 1536–1547. https://doi.org/10.18653/v1/2020.findings-emnlp.139
4. Ha, T. T., Nguyen, V. N., Nguyen, K. H., Nguyen, K. A., & Than, Q. K. (2021). Utilizing SBERT For Finding Similar Questions in Community Question Answering. *Proceedings - International Conference on Knowledge and Systems Engineering, KSE*, *2021-November*. https://doi.org/10.1109/KSE53942.2021.9648830
5. Izacard, G., Caron, M., Hosseini, L., Riedel, S., Bojanowski, P., Joulin, A., & Grave, E. (2022). Unsupervised Dense Information Retrieval with Contrastive Learning. *Transactions on Machine Learning Research*. https://mlanthology.org/tmlr/2022/izacard2022tmlr-unsupervised/
6. Järvelin, K., & Kekäläinen, J. (2002). Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)*, *20*(4), 422–446. https://doi.org/10.1145/582415.582418
7. Jurczyk, P., & Agichtein, E. (2007). Discovering authorities in question answer communities by using link analysis. *International Conference on Information and Knowledge Management, Proceedings*, 919–922. https://doi.org/10.1145/1321440.1321575
8. Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., & Yih, W. T. (2020). Dense Passage Retrieval for Open-Domain Question Answering. *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 6769–6781. https://doi.org/10.18653/V1/2020.EMNLP-MAIN.550
9. Khattab, O., & Zaharia, M. (2020). ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. *SIGIR 2020 - Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 39–48. https://doi.org/10.1145/3397271.3401075;TOPIC:TOPIC:CONFERENCE-COLLECTIONS
10. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Advances in Neural Information Processing Systems*, *33*, 9459–9474. https://github.com/huggingface/transformers/blob/master/
11. Li, Z., Zhang, X., Zhang, Y., Long, D., Xie, P., Zhang, M., & Group, A. (2023). *Towards General Text Embeddings with Multi-stage Contrastive Learning*. https://arxiv.org/pdf/2308.03281
12. Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 3982–3992. https://doi.org/10.18653/v1/D19-1410
13. Riahi, F., Zolaktaf, Z., Shafiei, M., & Milios, E. (2012). Finding expert users in Community Question Answering. *WWW'12 - Proceedings of the 21st Annual Conference on World Wide Web Companion*, 791–798. https://doi.org/10.1145/2187980.2188202
14. Robertson, S. E., & Walker, S. (1994). Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1994*, 232–241. https://doi.org/10.1007/978-1-4471-2099-5_24
15. Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, *24*(5), 513–523. https://doi.org/10.1016/0306-4573(88)90021-0
16. Seema, & Banati, H. (2025). *Expert Identification in Online Q&A with MMFSA*. 1–6. https://doi.org/10.1109/icrito66076.2025.11241769
17. Shi, W., Min, S., Yasunaga, M., Seo, M., James, R., Lewis, M., Zettlemoyer, L., & Yih, W. (2024). *REPLUG: Retrieval-Augmented Black-Box Language Models*. 8371–8384. https://doi.org/10.18653/V1/2024.NAACL-LONG.463
18. Voorhees, E. M., & Tice, D. M. (2000). *The TREC-8 Question Answering Track*. http://www.lrec-conf.org/proceedings/lrec2000/pdf/26.pdf
19. Wang, L., Yang, N., Huang, X., Jiao, B., Yang, L., Jiang, D., Majumder, R., & Wei, F. (2022). *Text Embeddings by Weakly-Supervised Contrastive Pre-training*. https://arxiv.org/pdf/2212.03533
20. Xiao, S., Liu, Z., Zhang, P., Muennighoff, N., Lian, D., & Nie, J. Y. (2024). C-Pack: Packed Resources For General Chinese Embeddings. *SIGIR 2024 - Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 641–649. https://doi.org/10.1145/3626772.3657878

21. Zhai, C., & Lafferty, J. (2001). A study of smoothing methods for language models applied to ad hoc information retrieval. *SIGIR Forum (ACM Special Interest Group on Information Retrieval)*, 334–342. https://doi.org/10.1145/383952.384019

22. Zhao, Z., Zhang, L., He, X., & Ng, W. (2015). Expert Finding for Question Answering via Graph Regularized Matrix Completion. *IEEE Transactions on Knowledge and Data Engineering*, *27*(4), 993–1004. https://doi.org/10.1109/TKDE.2014.2356461

❖ ❖ ❖