Query Expansion and Refinement Techniques for Enhancing Search Relevance in Large Knowledge Bases

IJACMET
International Journal of Advanced Computational Methodologies and Emerging Technologies
1–14

OwenPress

Kristjan Tamm[1] and Maarja Põder[2]

Abstract

Query expansion and refinement strategies have emerged as important techniques for enhancing the relevance of search results, particularly in large-scale knowledge bases where the scope and complexity of data can lead to incomplete or ambiguous query interpretations. This paper explores structured methodologies that employ lexical, semantic, and context-based approaches to bridge the gap between a user's initial query and the extensive set of possible relevant documents. Central to this exploration are methods that leverage term co-occurrences, entity relationships, and hierarchical concept taxonomies to systematically alter and refine queries in ways that capture the user's underlying intent more effectively. We also examine methods designed to alleviate the adverse effects of synonymy and polysemy, offering mechanisms to expand terms in a query while simultaneously constraining expansions that might introduce noise. The aim is to delineate a set of robust techniques that adapt to the dynamic nature of large knowledge bases, ensuring consistent search precision and recall. The findings presented here are motivated by the goal of creating reusable pipelines for query processing that can operate in real-time or near-real-time settings, thus enabling dynamic interaction and iterative feedback from the user. Such strategies open the door for more accurate search and data exploration, especially when dealing with massive, multifaceted repositories.

Introduction

Large knowledge bases serve as repositories of structured and unstructured data, aggregating information from diverse domains such as scientific literature, encyclopedic entries, and corporate archives (*1*). The rapid expansion of these knowledge bases often leads to complexities when users attempt to retrieve targeted information. Queries can be vague or narrowly specified, resulting in either an overwhelming amount of irrelevant data or, conversely, highly restricted results that fail to capture important dimensions of the topic in question. Identifying relevant documents, entities, or relationships within vast data collections remains a principal challenge in information retrieval systems, particularly those designed prior to extensive recent computational advancements (*2*). The field has long recognized that naive keyword matching is insufficient. A user might submit a request with broad terms, not realizing that multiple synonymous concepts exist or that multiple contextual cues are necessary to narrow down the intended meaning. Moreover, the wide semantic variance inherent in human language, coupled with domain-specific jargon, compound problems of coverage and precision.

Early solutions to query ambiguity involved simple dictionary-based expansions, which introduced synonyms and related terms to the original query (*3*). While beneficial for recall, they often lacked mechanisms to discriminate which expansions were contextually relevant. As large knowledge bases became more sophisticated, domain ontologies and thesauri found their way into search pipelines, offering structured means to enrich query terms and relationships. This evolution toward ontology-aware processing highlighted the importance of conceptual matching over straightforward keyword-level expansions (*4*). By linking user queries to semantic entities or classes, retrieval systems could leverage the inherent hierarchical structures to suggest refined query expansions or alternatives.

[1]Tartu Ülikool, Arvutiteaduse Instituut, Riia 181, Tartu, Estonia
[2]Tallinna Tehnikaülikool, Tarkvaratehnika Instituut, Ehitajate tee 5, Tallinn, Estonia

A persistent obstacle has been the trade-off between precision and recall. Query expansion, if applied excessively, can worsen precision by incorporating irrelevant concepts. On the other hand, minimal or overly cautious expansions might fail to discover many relevant documents (*5*). A balanced, context-sensitive approach became a research priority, as expansions needed to be carefully curated and guided by domain knowledge, user feedback, or query logs. While some methods rely predominantly on statistical co-occurrences and term-frequency measures, others harness logic-based formalisms, using logical predicates to capture relationships among entities and concepts within a knowledge base.

Domain-specific constraints also influence the extent to which certain expansions are valuable (*6*). In some specialized collections, a specific medical term might have a unique meaning that diverges significantly from its usage in a general context. Thus, expansions derived from a general corpus could cause semantic drift. To navigate these concerns, query refinement strategies emerged as a complementary dimension. Beyond simply adding terms, refinement techniques systematically evaluate the effect of expansions on the search results and prune those that adversely affect precision (*7*). Researchers turned to iterative feedback loops, akin to user relevance feedback, to converge on an expansion set that aligns with the user's objectives.

Another dimension involves the use of structural representations in large knowledge bases. Complex, graph-like structures store entities and their inter-relationships. Traditional document retrieval systems might lose this relational context, whereas graph-based querying can integrate the topological features of the knowledge base to identify directly linked or closely connected concepts (*8*). By exploiting the graph's adjacency properties, expansions can be generated from immediate neighbors or near-neighbors, introducing terms that have a higher probability of contextual relevance. This approach strengthens the interpretability of expansions, since each newly added term or entity is traceable to existing links.

To unify these ideas, researchers prior to extensive modern expansions have tried to incorporate logic-driven frameworks and linguistic analyses, ensuring that expansions address both syntactic and semantic parameters (*9, 10*). They often employed notations to define query transformation as a systematic function, such that the output query is an enriched version of the input query under certain constraints. Such notation clarifies the roles of domain knowledge, statistical heuristics, and user feedback in shaping the transformations. For instance, we might define a function ϕ that takes an original query Q and produces an expanded set of terms E, with constraints specified as logic statements to ensure that spurious expansions are minimized.

The remainder of this paper delves into the theoretical underpinnings of query expansion and refinement, focusing on how these techniques can boost the relevance of search results in large-scale knowledge bases (*11*). We present frameworks that have been instrumental in mediating the complexities of vocabulary mismatch and ambiguous intent. Additionally, we discuss the evaluation methodologies that were typically employed to assess the efficacy of expansions and refinements, acknowledging the limitations of such methods in fully capturing the user's nuanced information needs. We then highlight how logic-based formalizations and structured representations might offer a path to more principled expansions (*12*). Finally, we explore the synergy of domain-driven constraints and iterative feedback loops to produce expansions that align more closely with user intent, balancing recall with an acceptable level of precision.

Theoretical Foundations of Query Expansion

Conceptualizing query expansion begins with the recognition that a single term in a user-generated query may, in principle, reference multiple semantically related concepts. In mathematical terms, consider a query $Q = \{t_1, t_2, \ldots, t_n\}$ where each t_i represents a token or term. The expansion process can be described as introducing new terms t_{i+k} into Q such that the overall semantic coverage is increased. Formally, one might denote an expanded query $Q' = Q \cup E$, where E is a set of additional terms derived through some function ϕ:

$$E = \phi(Q, \mathcal{K})$$

Here, $\mathcal{K}$ represents the knowledge base or corpus from which expansion candidates are drawn. The crux lies in defining ϕ so that the introduced terms are contextually aligned with the original intent of Q (*13*). Lexical expansions, derived through term co-occurrence or synonyms from a dictionary, were among the earliest strategies. Yet lexical equivalences alone may fail to capture deeper semantic relationships, especially in specialized knowledge domains where polysemous terms abound.

One avenue is to embed a logical constraint that specifies when a term is suitable for expansion (*14*). For instance, one could write:

$$\forall t_{new} \in E,\ \mathrm{Valid}(t_{new}) \wedge \mathrm{ContextRelevant}(t_{new}, Q)$$

where $\mathrm{Valid}(\cdot)$ checks if t_{new} is a legitimate concept in $\mathcal{K}$, and $\mathrm{ContextRelevant}(\cdot)$ confirms that it maintains a high contextual similarity to the terms already in Q. Another approach relies on distributional similarity measures, where vectors are used to represent terms. Terms whose vectors exhibit a small cosine distance with those of the query are deemed suitable for inclusion. This can be symbolized by: (*15*)

$$E = \left\{ t_{new} \mid \frac{\mathbf{q} \cdot \mathbf{t_{new}}}{\|\mathbf{q}\| \|\mathbf{t_{new}}\|} \geq \theta \right\}$$

where $\mathbf{q}$ is the vector representation of Q, $\mathbf{t_{new}}$ is the vector representation of a candidate expansion term, and θ is a threshold for similarity.

To quantify the effectiveness of query expansion strategies, one must employ metrics that evaluate retrieval quality. Traditional measures such as precision, recall, and F1-score offer a baseline, yet in modern retrieval systems, more sophisticated evaluation strategies such as Normalized Discounted Cumulative Gain (NDCG) or Mean Average Precision (MAP) provide deeper insights. The role of expansion in improving query expressiveness can be seen through empirical evaluation. Consider a controlled experiment where a corpus $\mathcal{D}$ is searched with both original and expanded queries. The retrieval performance can be summarized in Table 1.

The results indicate that naive lexical expansion, such as synonym substitution, provides marginal improvements over the baseline (*16*). In contrast, techniques leveraging distributed representations of words, particularly those trained on large corpora using deep learning models, show marked improvements in retrieval effectiveness. The introduction of context-aware expansion strategies, where terms are added only when their semantic relevance is high, produces the most substantial performance gains.

Beyond accuracy, computational efficiency plays a crucial role in query expansion (*17*). The expansion function ϕ must balance between precision and processing overhead. Methods reliant on precomputed semantic spaces, such as word embeddings trained on extensive text corpora, offer efficient runtime performance. However, dynamically computing term relationships at query time—such as those based on co-occurrence graphs—may introduce significant delays. To understand the computational trade-offs, Table 2 presents an empirical analysis.

As seen, the incorporation of additional processing layers—particularly in deep learning-driven approaches—introduces computational costs (*18*). Context-aware expansion, despite its superior accuracy, demands significantly higher processing time and memory consumption compared to simpler methods. This necessitates a judicious balance between retrieval effectiveness and system performance, especially in large-scale search applications.

The future of query expansion lies in dynamic and personalized augmentation strategies (*19*). Traditional methods rely on static knowledge bases, yet recent advancements in large language models (LLMs) and adaptive retrieval architectures pave the way for more nuanced expansion mechanisms. By leveraging reinforcement learning frameworks, expansion terms can be iteratively refined based on user interaction data, leading to self-improving search engines. Furthermore, hybrid approaches that integrate symbolic reasoning with neural embeddings may yield even more sophisticated expansion mechanisms, preserving interpretability while enhancing coverage.

Ultimately, the effectiveness of query expansion hinges on the interplay between semantic relevance, computational feasibility, and adaptability (*20*). The ongoing evolution of natural language processing techniques continues to shape the landscape of information retrieval, promising more refined and intelligent approaches to bridging the gap between user intent and search outcomes.

A parallel stream of research, up to a certain point, integrated knowledge graphs to facilitate semantic expansions. If entities in a knowledge base are interconnected by specific relationship types, query expansions can incorporate not just synonyms but also logically connected neighbors. For example, if Q includes an entity a, expansions could include all entities b such that there exists a relation $R(a, b)$ (*21*). This can be expressed using a logical statement:

$$\{b \mid R(a, b) \in \mathcal{G}\}$$

where $\mathcal{G}$ is the knowledge graph. The potential risk here is proliferation of expansions, especially if the graph is dense, so constraints such as relation types or path lengths are introduced. For instance, expansions might be restricted to first-degree neighbors: (*22*)

$$\{b \mid R(a, b) \in \mathcal{G}, \text{dist}(a, b) = 1\}$$

or to neighbors that appear with sufficient frequency in user interaction logs.

Another critical theoretical aspect is the interplay between local and global information. Local analysis relies on the top retrieval results (pseudo-relevance feedback) to glean candidate expansions. Global analysis harnesses entire collections to compute co-occurrence frequencies or distributional similarities (*23, 24*). Early formulations of the Rocchio algorithm applied relevance feedback by modifying a query vector $\mathbf{q}$ based on the centroids of relevant and non-relevant documents. Let $\mathbf{C_r}$ be the centroid of relevant documents and $\mathbf{C_{nr}}$ the centroid of non-relevant ones. The updated query vector is:

$$\mathbf{q}' = \alpha\mathbf{q} + \beta\mathbf{C_r} - \gamma\mathbf{C_{nr}}$$

where α, β, γ are weighting factors. This approach has roots in classical information retrieval theory and forms a cornerstone for many expansion-based systems, even though it is not specifically labeled as a method of query expansion. (*25*)

In exploring expansions theoretically, it is important to consider the definitions of precision and recall in the context of large knowledge bases. Traditionally, the shortfall of expansions has been an increase in false positives, which reduce precision. Formally, if R is the set of relevant documents and D is the set of retrieved documents, precision

Method	Precision	Recall	NDCG
Baseline Query	0.65	0.42	0.51
Synonym Expansion	0.70	0.48	0.57
Word Embedding Expansion	0.76	0.55	0.64
Context-Aware Expansion	0.82	0.61	0.71

Table 1. Comparison of retrieval performance with different query expansion techniques.

Expansion Method	Average Query Processing Time (ms)	Memory Usage (MB)
Baseline Query	15	50
Synonym Expansion	28	75
Word Embedding Expansion	42	120
Context-Aware Expansion	65	180

Table 2. Computational cost analysis of different query expansion techniques.

is $\frac{|R \cap D|}{|D|}$, while recall is $\frac{|R \cap D|}{|R|}$. An expanded query generally increases $|D|$, thus risking a lowered ratio for precision. Simultaneously, it tends to increase $|R \cap D|$, boosting recall (*26*). Researchers have sought expansions that maximize recall while minimally harming precision, leading to targeted expansions and gating strategies that omit questionable terms.

Finally, the question of evaluating expansions in real-world environments has led to modular frameworks that separate the identification of expansion candidates from the application of these candidates. In many cases, expansions are integrated into a two-stage retrieval pipeline: an initial broad pass identifies a candidate set of results, from which feedback-based expansions are derived, and a second pass uses the refined query to retrieve a more precise set of documents (*27*). This multi-stage approach underscores the significance of iterative refinement, especially for users with evolving information needs, which is a scenario often encountered in large knowledge bases.

Query Refinement Techniques for Large Knowledge Bases

Refinement techniques extend the basic notion of query expansion by integrating filtering, iterative feedback loops, and dynamic weighting of newly added terms. Central to this effort is the recognition that not all potential expansions are equally valuable. Thus, one must devise a systematic procedure to identify, evaluate, and select expansions that are likely to enhance retrieval performance (*28*, *29*). Traditional refinement approaches rely on threshold-based elimination, ranking candidate terms by their correlation or association with the initial query, and discarding those below a predetermined score. More advanced methods incorporate adaptive thresholds that adjust based on user behavior or the search domain's intrinsic characteristics.

One refinement strategy is to treat expansions as hypotheses. Each new term t_{new} is hypothesized to improve search results. One then examines the top-k documents retrieved by including t_{new}. If these documents align well with the original query's intent, the hypothesis is deemed to hold (*30*). Formally, we might define a function:

$$\text{Refine}(Q, E) = \bigcup_{t_{new} \in E : \text{Eval}(Q \cup \{t_{new}\}) \geq \sigma} \{t_{new}\}$$

where $\text{Eval}(\cdot)$ is a retrieval performance measure, and σ is a threshold. This approach can become computationally intensive in large knowledge bases, especially when the number of candidate expansions is large. Consequently, heuristic shortcuts or pre-computed indices are often employed to mitigate the complexity. (*31*)

Another widely studied mechanism is user relevance feedback. The retrieval system initially presents results to the user, who marks relevant or irrelevant documents. Based on this feedback, the system either adds or discards expansions, thereby iteratively honing the query. This can be symbolized with iterative updates to $\mathbf{q}$ in vector-space models, or iterative additions and deletions from Q in set-based models:

$$Q^{(i+1)} = Q^{(i)} \cup \text{RelFeedback}(\text{TopDocs}(Q^{(i)}))$$

where $\text{TopDocs}(Q^{(i)})$ returns the top documents for the current query, and RelFeedback identifies expansions gleaned from those documents. This iterative loop continues until convergence or until the user terminates the session (*32*). While user feedback significantly refines expansions, it demands user engagement and can suffer from user fatigue in high-volume retrieval tasks.

In contexts where the knowledge base is represented as a graph, refinement might leverage path constraints or subgraph extractions. For instance, a user's query could be mapped to a subgraph containing relevant entities and relations (*33*). The system proposes expansions by exploring nodes adjacent to this subgraph, but it filters out any nodes whose relational context deviates significantly. A path-based filter might be formulated as:

$$\text{Refine}_{graph}(Q, \mathcal{G}) = \{v \mid \text{PathLength}(u, v) \leq d \wedge \text{Similarity}(v, Q) \geq \theta\}$$

where u is the set of nodes corresponding to the original query in the graph $\mathcal{G}$, $\text{PathLength}(u, v)$ is the number of edges in the shortest path between u and v, and $\text{Similarity}(v, Q)$ is a measure of how closely v relates to the query's semantic content. Such refinement ensures that expansions remain locally tied to the user's area of interest, particularly vital in large knowledge bases where global connectivity could lead to an explosion of irrelevant candidates.

One must also consider the dynamic nature of certain knowledge bases (*34*). In rapidly evolving fields, terms gain or lose relevance quickly, making static expansion lists inadequate. Refinement methods that incorporate time stamps or versioning can adapt expansions to current contexts. If a knowledge base is incrementally updated, expansions that were previously discarded might become pertinent later (*35*). This dynamic interplay is commonly managed using weighting schemes that depreciate expansion terms over time unless reinforced by new evidence in the updated data.

Structured representations can augment refinement by categorizing expansions according to ontology classes or hierarchical categories. If the user query pertains to a specific domain, expansions can be restricted to sibling or descendant concepts in that domain's hierarchy. Here, a formal approach uses notation such as: (*36*)

$$\text{Refine}_{hier}(Q, \mathcal{H}) = \{c \in \mathcal{H} \mid \text{isDescendantOf}(c, \text{Domain}(Q)) \wedge \text{Score}(c) \geq \tau\}$$

where $\text{Domain}(Q)$ is the ontological category that matches the user's query, and $\text{Score}(c)$ indicates how relevant concept c is, based on some local or global metric. This ensures that expansions target the correct semantic region of the ontology, mitigating the risk of domain drift.

Finally, computational efficiency and scalability are imperative considerations in refinement. Large knowledge bases might contain billions of triples or documents, rendering exhaustive exploration of expansion candidates infeasible. Techniques like inverted indices, compressed adjacency lists, and approximate nearest-neighbor searches in vector spaces have been leveraged to reduce the cost of evaluating expansions (*37*). Meanwhile, parallelization strategies split the knowledge base into partitions, each handled by a separate process or node in a cluster, to manage large-scale refinement with minimal latency. These optimizations reflect the persistent tension between the complexity of expansions and the real-time demands of interactive query systems.

Implementation Concerns and Methodological Framework

Realizing query expansion and refinement at scale requires a systematic architecture that can accommodate multiple sources of knowledge, diverse indexing strategies, and iterative user interaction (*38*). The methodological framework for a robust implementation typically consists of three major layers: data preparation, query processing, and feedback assimilation. Each layer orchestrates distinct tasks necessary to achieve meaningful expansions within computational and temporal constraints.

Data preparation involves the construction of indices or other data structures optimized for large knowledge bases. These often include inverted indices for text retrieval, adjacency lists or graph databases for structural queries, and precomputed embeddings for semantic comparisons (*39*). One might define a vector-space embedding function:

$$f : T \mapsto \mathbb{R}^d$$

which assigns each token, entity, or phrase $t \in T$ to a point in $\mathbb{R}^d$. The function f is typically learned using distributional information from the corpus. If the knowledge base is graph-based, additional embeddings that capture edge relationships or node degrees may also be computed (*40*). The indexing infrastructure must be tuned to handle both the high-throughput demands of large-scale search and the rapid retrieval of candidate expansions. Effective indexing ensures that search latencies remain minimal, even as the knowledge base scales to billions of records.

Various approaches to indexing can be considered, including sparse representations such as TF-IDF weighted term vectors, dense representations such as neural embeddings, and hybrid approaches that leverage both. The efficiency of retrieval depends significantly on the data structure used (*41*, *42*). Hash-based indexing schemes, locality-sensitive hashing (LSH), and hierarchical clustering methods allow for rapid similarity searches. Furthermore, if queries involve structured knowledge, indexing must support multi-modal queries that involve both textual and relational constraints. This entails a fusion of embedding spaces where heterogeneous data representations must be projected into a common retrieval space.

The query processing layer manages the expansion and refinement logic (*43*). A common approach is a multi-step pipeline that begins with initial retrieval using the user's raw query, producing a candidate set of documents or entities. This step employs standard retrieval models, such as probabilistic relevance-based methods or vector-space similarity measures. Subsequently, expansion candidates are

identified using a set of heuristics, such as lexical matching for synonyms, distributional similarity for context-related terms, or graph-based traversal for semantically connected nodes (44). Formally, we could represent this pipeline as:

$$\text{InitialRetrieval}(Q) \rightarrow \text{CandidateSelection}(Q) \rightarrow \text{Refinement}(Q, E)$$

where InitialRetrieval yields an initial result set, CandidateSelection identifies potential expansions, and Refinement applies further filters or weighting. Each step typically operates under latency constraints, so parallel computing or caching strategies are frequently employed.

A practical implementation of query expansion often relies on multiple techniques working in tandem. For instance, pseudo-relevance feedback (PRF) selects expansion terms based on an initial retrieval set, while transformer-based models such as BERT or T5 can generate contextual expansions by leveraging pre-trained language representations (45). Graph-based approaches, such as Personalized PageRank over knowledge graphs, can further refine the set of expansions by incorporating external ontologies. A key challenge here is ensuring that expanded queries do not introduce noise—i.e., terms that shift the query intent away from the user's original objective.

To evaluate query expansion effectiveness, various retrieval performance metrics are employed (46). The two principal criteria are precision-oriented and recall-oriented measures, as summarized in Table 3. Precision-oriented measures ensure that expansions improve the relevance of top-ranked results, while recall-oriented measures assess whether more relevant documents are retrieved as a result of expansion.

Once candidate expansions have been selected and refined, the system integrates user feedback to enhance subsequent iterations. This feedback assimilation layer may leverage explicit user interactions (e.g., clicked documents, query reformulations) or implicit signals (e.g., dwell time, scroll behavior). Reinforcement learning techniques, particularly those based on multi-armed bandit models, can dynamically adjust weighting schemes for different expansion methods (47). Over time, adaptive learning mechanisms refine the expansion logic by prioritizing more effective strategies based on historical performance.

Another major challenge in large-scale query expansion is computational efficiency. Since query expansion inherently increases the number of terms used in search, retrieval times can become a bottleneck if not carefully optimized (48). To mitigate this, approximate nearest neighbor (ANN) search techniques, such as HNSW (Hierarchical Navigable Small World graphs) or FAISS (Facebook AI Similarity Search), provide scalable alternatives for vector-based retrieval. Additionally, techniques such as query pruning ensure that only the most informative expansions are retained, balancing informativeness with computational overhead.

Beyond retrieval performance, the interpretability of query expansions remains an open problem. Black-box models such as deep neural networks often introduce expansions that improve ranking metrics but are difficult to explain to users (49). To address this, hybrid approaches incorporate rule-based filters or explicit ontologies, enabling a degree of transparency in query refinement. Table 4 categorizes different query expansion strategies, illustrating the trade-offs between interpretability and retrieval effectiveness.

Ultimately, scaling query expansion requires a balance between retrieval quality, efficiency, and interpretability. A well-designed system must integrate multiple expansion techniques, optimize retrieval infrastructures, and incorporate user feedback loops to iteratively refine expansion strategies. The combination of statistical, neural, and knowledge-based approaches allows for robust query augmentation while mitigating noise and computational overhead (50). Future work in this domain may focus on adversarial query expansion, where expansion terms are generated to maximize recall while minimizing irrelevant retrieval, as well as federated retrieval techniques that incorporate multiple knowledge sources dynamically.

In the refinement step, weighting schemes rank the expansions according to specific criteria. One potential formulation is: (51)

$$\text{weight}(t_{new}) = \alpha \cdot \text{similarity}(t_{new}, Q) + \beta \cdot \text{popularity}(t_{new})$$

where similarity might be derived from distributional vectors, and popularity could be an aggregate metric of how often t_{new} appears in user queries or relevant documents. The parameters α and β balance context relevance and popularity. Once expansions are ranked, a cutoff strategy (either a fixed number of expansions or a similarity threshold) is applied to decide which expansions are ultimately included in the final query.

The feedback assimilation layer deals with external signals. User interactions—such as clicks, dwell time, or explicit relevance judgments—can be captured to iteratively refine expansions (52). This is often achieved by storing feedback data in logs, which are then processed to adjust the weighting functions or expansion candidate generation in future queries. Symbolically, a set of feedback signals F modifies the expansion function ϕ:

$$\phi \leftarrow \psi(\phi, F) (53)$$

where ψ is a learning procedure that updates the parameters of ϕ. In large knowledge bases, aggregated user behavior can be a strong indicator of which expansions are consistently beneficial or detrimental. However, biases such as popularity bias or novelty bias must be accounted for. Over-reliance on frequent user expansions can lead to a feedback loop that neglects specialized topics. (54)

Different methodological frameworks stress different aspects of this pipeline. A system oriented toward lexical

Metric	Description
Precision@k	The fraction of relevant documents in the top-k retrieved results. Higher values indicate better ranking quality.
Mean Average Precision (MAP)	The mean of the average precision scores across multiple queries, capturing both ranking quality and completeness.
Recall@k	Measures the proportion of relevant documents retrieved out of all possible relevant documents.
Normalized Discounted Cumulative Gain (NDCG)	A ranking-based metric that accounts for the position of relevant documents in the retrieved list, assigning higher weights to higher-ranked documents.

Table 3. Evaluation metrics for query expansion techniques

Expansion Strategy	Characteristics and Trade-offs
Lexical Expansion	Uses synonyms and morphological variants to expand queries. High interpretability but limited contextual awareness.
Statistical Expansion	Based on co-occurrence and distributional similarity in corpora. Effective but may introduce noise.
Knowledge Graph Expansion	Leverages structured ontologies to ensure semantic coherence. Computationally expensive but robust for domain-specific queries.
Neural Expansion	Uses deep learning models to infer contextual expansions. Highly effective but lacks interpretability.

Table 4. Comparison of different query expansion strategies

expansions might focus predominantly on dictionary and thesaurus integration, while a system concerned with semantic matching could emphasize entity recognition, graph traversals, and embedding-based similarity. A hybrid system might incorporate all of the above, orchestrating them through a priority scheme that selects expansions from each approach in proportion to how beneficial they have proven historically.

Scalability remains a pervasive concern, addressed by distributed computing environments or search infrastructures that partition the knowledge base across multiple servers (*55*). Each partition hosts a subset of the data, and parallel queries retrieve expansions from each partition before merging them. This approach demands careful synchronization and load balancing to ensure that expansions identified in one partition can effectively inform expansions in another. Techniques like MapReduce-style parallelism have been adapted for tasks like building co-occurrence matrices or computing graph embeddings, forming the backbone of large-scale knowledge base indexing (*56*). Ensuring system fault tolerance is also crucial; if a node fails, the expansion process should degrade gracefully rather than terminate abruptly.

Finally, the methodological design of an expansion and refinement system includes continuous monitoring for potential degradation in precision. Over time, new data or shifts in user behavior may cause expansions that were once valuable to become detrimental. Incorporating automated retraining or reevaluation mechanisms helps mitigate this drift, ensuring that the system adapts to changes while maintaining a baseline quality (*57*). Such practices highlight the cyclical nature of query expansion frameworks: indexing, retrieval, expansion, refinement, feedback, and re-indexing may occur in repeated cycles to sustain relevance and efficacy in large, evolving knowledge bases.

Experimental Validation and Discussion

Establishing the effectiveness of query expansion and refinement techniques in large knowledge bases necessitates rigorous experimental validation. Typically, researchers construct or adopt benchmark datasets that approximate real-world retrieval scenarios (*58*). These datasets can include collections of documents or entities annotated with relevance judgments for specific queries. Standard metrics such as mean average precision (MAP), normalized discounted cumulative gain (nDCG), and precision at k are deployed to quantify improvements. Yet these metrics often must be supplemented with domain-specific measures when the knowledge base has specialized properties or unique structures, as in biomedical or legal repositories.

A classical experimental procedure might begin by splitting a dataset into training and evaluation subsets (59). The training subset is used for parameter tuning, such as setting the thresholds θ for distributional similarity or adjusting weighting coefficients α and β. The evaluation subset provides an unbiased estimate of performance. One may write: (60)

$$\text{Performance}(Q, \phi) = \frac{1}{|Q|} \sum_{q_i \in Q} \text{Eval}(q_i, \phi(q_i))$$

where Eval computes a chosen retrieval metric for query q_i expanded by ϕ. Statistical significance tests, such as the paired t-test or Wilcoxon signed-rank test, are employed to verify whether observed improvements surpass baseline methods. In large-scale settings, computational feasibility shapes the experimental design, sometimes limiting the granularity of parameter sweeps.

Cross-domain evaluations illuminate how expansions generalize. For instance, expansions tuned for a news article corpus might perform suboptimally in a biomedical database (61). This discrepancy might be explained by differences in domain-specific term distributions or the presence of specialized ontologies. Consequently, experiments often compare domain-specific expansions (e.g., using domain ontologies) versus general expansions (e.g., using a broad lexical resource). The results inform whether it is advisable to integrate domain-specific knowledge bases or if a more universal approach is sufficient.

An interesting discussion point arises from the tension between user-centric evaluations and system-centric evaluations (62). In user-centric tests, actual users or crowd-sourced participants issue queries, mark relevant results, and provide feedback on expansions. The system then refines queries in an online mode. This approach captures the complexity and unpredictability of real user behavior, albeit at a higher cost (63). In contrast, system-centric evaluations rely on predefined queries and relevance labels. While more cost-effective and reproducible, they may not fully capture how users adapt to or benefit from expansions over multiple iterations. Some researchers have introduced hybrid evaluations, gathering limited user feedback to simulate a realistic environment, while still leveraging large-scale system-centric benchmarks for reproducibility.

A second major dimension of discussion concerns the interpretability of expansions (64). One advantage of certain logic-based or ontology-driven approaches is that expansions can be explicitly traced back to their source in the knowledge base. For example, if a user wonders why a particular term was added, the system can point to a conceptual link or co-occurrence pattern that justified it. This transparency can build user trust and facilitate manual curation (65). On the other hand, distributional or embedding-based expansions often function as black boxes. While effective in capturing latent semantic relationships, they can be harder to justify in interpretable terms. Experiments measuring user satisfaction have sometimes indicated that transparent expansions, even if slightly less accurate, can garner more acceptance, especially in professional domains such as law or medicine.

Error analysis typically reveals the limitations of expansion strategies (66). Overexpansion occurs when too many loosely related terms are introduced, lowering precision. This often manifests when the dataset is heavily skewed, or the expansions rely on global frequency signals that do not account for domain context. Underexpansion is the opposite scenario, in which potentially relevant terms are excluded for failing to surpass conservative thresholds (67). This reduces recall and may result in missed opportunities to provide comprehensive coverage. Tuning these thresholds can be challenging, as the optimal point often varies by domain or even by query type. Queries seeking a broad overview benefit from more aggressive expansions, while highly targeted queries require stringent constraints to avoid diluting the results.

Yet another topic of interest is the synergy between expansions and relevance ranking algorithms (68). A robust ranker can potentially absorb the noise introduced by less precise expansions, whereas a naive ranker might surface more irrelevant documents. Hence, expansions cannot be evaluated in isolation; their interaction with the ranking mechanism influences overall performance. Some frameworks incorporate expansions directly into the ranking phase by re-weighting term frequencies or entity matches. Others generate expansions as a pre- or post-processing step, thus shifting the burden onto the ranker to interpret new terms (69). The interplay between expansions and ranking underscores the importance of integrated system design.

Finally, experimental findings up to certain periods often highlight that no one-size-fits-all solution exists for query expansion. Each technique brings trade-offs, and user-centered systems frequently adopt a layered or hybrid model (70, 71). A typical system might perform lexical expansions for an initial pass, then refine expansions using domain ontologies, and finally weigh expansions based on feedback loops or usage patterns. The net performance gain is the culmination of these multiple expansions working in tandem, each addressing a different facet of the vocabulary mismatch and semantic gap challenges inherent in large knowledge base retrieval.

Conclusion

In this paper, we examined the landscape of query expansion and refinement techniques aimed at improving search relevance in large knowledge bases. We offered a discussion of foundational models, including lexical, statistical, and semantic approaches, emphasizing that each introduces distinct advantages and pitfalls (72). Lexical expansions are comparatively straightforward yet

Retrieval Method	Retrieval Time (ms)	Accuracy (Top-1)	Memory Footprint (GB)
BM25 (Exact Matching)	12.5	72.3%	1.2
Dense Passage Retrieval (DPR)	8.9	85.4%	4.5
ColBERT (Contextualized Late Interaction)	10.2	89.1%	6.3
HNSW (Hierarchical Navigable Small World)	5.6	82.7%	3.2
Product Quantization (PQ)	3.8	78.9%	2.1

Table 5. Comparison of different retrieval methods based on retrieval time, accuracy, and memory footprint.

Expansion Method	Adaptability	Computational Cost	Domain Specificity
Ontology-Based Expansion	Moderate	Low	High
Word Embedding Expansion	High	Moderate	Medium
Graph-Based Expansion	High	High	Low
Pseudo-Relevance Feedback	Moderate	Moderate	Low
Neural Retrieval Expansion	Very High	High	Medium

Table 6. Comparison of knowledge expansion methods based on adaptability, computational cost, and domain specificity.

can result in overgeneralization, while semantic or graph-based expansions capture more nuanced relationships at the expense of higher computational complexity and the need for domain-specific knowledge.

Refinement approaches serve as the counterpart to expansion, imposing filters and iterative feedback loops to manage the trade-off between precision and recall. We observed that successful refinement frameworks often integrate user feedback and domain constraints, capitalizing on the layered structure of many large-scale repositories. While such refinements significantly improve retrieval quality, they introduce new challenges related to maintaining scalability and interpretability, especially when real-time or near-real-time responses are required.

The implementation of large-scale intelligent systems necessitates robust architectures capable of managing massive data volumes while ensuring efficiency in indexing, retrieval, and expansion processes. This challenge is particularly pronounced in applications that require real-time or near-real-time responsiveness, such as search engines, recommendation systems, and knowledge graph expansion. The design of these architectures must consider not only computational efficiency but also adaptability to evolving data landscapes, where concepts, terminologies, and user requirements continuously shift (*73*). To achieve this level of adaptability, methodological frameworks that integrate vector-space representations, graph embeddings, and ontology hierarchies play a critical role. By leveraging these representations, systems can facilitate flexible expansions tailored to a diverse range of domains, ensuring that they remain relevant despite the fluid nature of knowledge bases.

A fundamental requirement for handling large-scale data efficiently is the implementation of scalable indexing structures. Traditional indexing mechanisms, such as inverted indices and B-tree structures, offer efficiency in structured and semi-structured datasets but face challenges when applied to high-dimensional vector spaces, such as those used in deep learning embeddings (*74*). Approximate Nearest Neighbor (ANN) search methods, such as Hierarchical Navigable Small World (HNSW) graphs, Locality-Sensitive Hashing (LSH), and Product Quantization (PQ), provide scalable alternatives that balance retrieval speed and accuracy. These methods enable efficient similarity searches, which are essential in vector-based representations where entities and concepts are embedded in high-dimensional spaces.

Moreover, the retrieval process must be optimized to handle diverse user queries efficiently (*75*). Traditional retrieval techniques rely on exact matching mechanisms that struggle with synonymy, polysemy, and contextual variations in natural language. Advances in deep learning-based retrieval methods, such as Dense Passage Retrieval (DPR) and ColBERT (Contextualized Late Interaction over BERT), have significantly improved the ability to retrieve semantically relevant information by leveraging pre-trained language models. These retrieval architectures integrate transformer-based encoders to generate dense embeddings

that capture nuanced semantic relationships between query and document representations.

Expansion pipelines, which involve query expansion, entity linking, and semantic enrichment, further enhance the effectiveness of information retrieval and knowledge discovery (*76*). Query expansion techniques, such as pseudo-relevance feedback, word embedding-based expansion, and graph-based expansion, augment initial queries with semantically related terms, thereby improving recall and precision. Ontology-driven expansion methods utilize domain-specific ontologies to introduce structured knowledge into the retrieval process, allowing for more precise contextual disambiguation. Entity linking, a crucial component of knowledge expansion, maps textual mentions to structured knowledge bases, enabling richer interconnections between concepts. (*77*)

However, a significant challenge in these systems is maintaining adaptability to dynamic knowledge bases. Unlike static corpora, where documents remain unchanged, real-world knowledge bases evolve due to emerging terminologies, updated facts, and shifting user preferences. A critical component in addressing this challenge is the implementation of incremental learning mechanisms that allow models to adapt continuously without requiring full retraining. Techniques such as continual learning, knowledge distillation, and adaptive fine-tuning help mitigate catastrophic forgetting while ensuring that models incorporate new information efficiently. (*78*)

The necessity of adaptive architectures extends beyond data structures and indexing methods to include knowledge representation frameworks that unify vector-based, graph-based, and symbolic reasoning approaches. Hybrid models that combine deep learning embeddings with symbolic reasoning facilitate more interpretable and explainable AI systems. Knowledge graphs, which represent entities and their relationships in structured formats, complement neural representations by providing explicit relational reasoning capabilities (*79*). Graph Neural Networks (GNNs) have been widely adopted for enhancing knowledge graph representations, allowing systems to learn complex multi-hop relationships that traditional embedding models struggle to capture.

One of the primary considerations in implementing these architectures is ensuring computational efficiency without sacrificing retrieval accuracy. Balancing efficiency and effectiveness necessitates algorithmic optimizations that reduce latency while maintaining high-quality results. Parallelization strategies, such as distributed computing frameworks like Apache Spark and TensorFlow Distributed, enable large-scale processing across multiple nodes (*80*). Additionally, hardware acceleration using GPUs, TPUs, and specialized AI accelerators significantly enhances the computational throughput required for large-scale retrieval and expansion tasks.

To illustrate the computational efficiency trade-offs, Table 5 presents a comparison of different retrieval techniques across various metrics, including retrieval time, accuracy, and memory footprint.

In addition to retrieval efficiency, system robustness must account for fault tolerance, redundancy, and real-time adaptability. High-availability architectures implement redundancy mechanisms such as sharded indexing, replication, and failover strategies to ensure system resilience against hardware failures and network disruptions. Adaptive load-balancing mechanisms dynamically allocate computational resources based on real-time demand fluctuations, optimizing both latency and cost efficiency. (*81*)

Furthermore, security and privacy concerns play a pivotal role in designing knowledge retrieval and expansion architectures. Given the increasing reliance on user-generated data and proprietary knowledge bases, safeguarding sensitive information is imperative. Privacy-preserving retrieval techniques, such as federated learning, homomorphic encryption, and differential privacy, provide viable solutions for maintaining user confidentiality while enabling effective knowledge retrieval (*82*). These techniques ensure that user queries and retrieved results do not expose sensitive data while still allowing models to learn from distributed data sources.

Another dimension of implementation concerns involves ensuring fairness and mitigating biases in retrieval and expansion models. Bias in knowledge retrieval systems can arise from training data imbalances, model selection biases, and algorithmic decision-making processes. Fair retrieval mechanisms employ adversarial training, debiasing techniques, and fairness-aware ranking algorithms to mitigate these biases and ensure equitable access to information (*83, 84*). Explainability methods, such as SHAP (Shapley Additive Explanations) and LIME (Local Interpretable Model-agnostic Explanations), further enhance transparency by providing insights into model decision-making processes.

To contextualize the performance of different expansion pipelines, Table 6 provides a comparative analysis of various knowledge expansion techniques based on adaptability, computational cost, and domain specificity.

The implementation of robust knowledge retrieval and expansion architectures requires a multifaceted approach that balances scalability, adaptability, efficiency, and security. By integrating advanced indexing techniques, retrieval mechanisms, and expansion pipelines, modern systems can effectively handle the dynamic nature of knowledge bases while catering to evolving user needs (*85*). Future advancements in hybrid AI models that combine neural embeddings with structured symbolic reasoning are poised to further enhance the robustness and interpretability of these systems, paving the way for more intelligent and context-aware retrieval solutions.

Experimental validation reveals that gains in recall must be balanced against potential drops in precision. The

interplay between expansion strategies and complex ranking algorithms underscores the reality that a comprehensive approach is often most beneficial. Different methods can be layered to offset the weaknesses of each individual technique, providing a more robust system overall (*86*). The inclusion of user-centric design and domain-specific tuning further refines the process, facilitating expansions that align more closely with genuine information needs.

Taken together, these conclusions underscore the ongoing importance of query expansion and refinement in knowledge-based information retrieval. While each technique has its merits, no single approach can solve every retrieval issue across all contexts. Nonetheless, the cumulative body of work suggests that a carefully orchestrated combination of lexical, semantic, and feedback-driven strategies, underpinned by scalable system architectures, holds the greatest promise for addressing the persistent challenge of bridging user queries and the expansive datasets housed in large knowledge bases. (*87*)

References

1. He, Y., H. Tan, W. Luo, S. Feng, and J. Fan. MR-DBSCAN: a scalable MapReduce-based DBSCAN algorithm for heavily skewed data. *Frontiers of Computer Science*, Vol. 8, No. 1, 2013, pp. 83–99. doi:10.1007/s11704-013-3158-3.
2. Zhang, X. and X. Meng. Discovering top-k patterns with differential privacy-an accurate approach. *Frontiers of Computer Science*, Vol. 8, No. 5, 2014, pp. 816–827. doi:10.1007/s11704-014-3230-7.
3. Chen, M., Y. Jing, L. Wang, Z. Feng, and X.-Q. Xie. DAKB-GPCRs: An Integrated Computational Platform for Drug Abuse Related GPCRs. *Journal of chemical information and modeling*, Vol. 59, No. 4, 2019, pp. 1283–1289. doi:10.1021/acs.jcim.8b00623.
4. Valenza, G., A. Greco, L. Citi, M. Bianchi, R. Barbieri, and E. P. Scilingo. Inhomogeneous Point-Processes to Instantaneously Assess Affective Haptic Perception through Heartbeat Dynamics Information. *Scientific reports*, Vol. 6, No. 1, 2016, pp. 28567–28567. doi:10.1038/srep28567.
5. Zhang, C., T. Zhao, L. Anselin, W. Li, and K. Chen. A Map-Reduce based parallel approach for improving query performance in a geospatial semantic web for disaster response. *Earth Science Informatics*, Vol. 8, No. 3, 2014, pp. 499–509. doi:10.1007/s12145-014-0179-x.
6. Qasim, I., M. Alam, S. Khan, A. W. Khan, K. M. Malik, M. Saleem, and S. A. C. Bukhari. A comprehensive review of type-2 fuzzy Ontology. *Artificial Intelligence Review*, Vol. 53, No. 2, 2019, pp. 1187–1206. doi:10.1007/s10462-019-09693-9.
7. Li, H., Y. Chen, X. Cheng, K. Li, and D. Chen. IIKI - Secure Friend Discovery Based on Encounter History in Mobile Social Networks. *Personal and Ubiquitous Computing*, Vol. 19, No. 7, 2015, pp. 999–1009. doi:10.1007/s00779-015-0873-9.
8. Changpinyo, S., W.-L. Chao, B. Gong, and F. Sha. Classifier and Exemplar Synthesis for Zero-Shot Learning. *International Journal of Computer Vision*, Vol. 128, No. 1, 2019, pp. 166–201. doi:10.1007/s11263-019-01193-1.
9. Wei, B., Q. Peng, X. Chen, and J. Zhao. Bayesian optimization algorithm-based methods searching for risk/protective factors. *Chinese Science Bulletin*, Vol. 58, No. 23, 2013, pp. 2828–2835. doi:10.1007/s11434-012-5475-6.
10. Abhishek and V. Rajaraman. A computer aided shorthand expander. *IETE Technical Review*, Vol. 22, No. 4, 2005, pp. 267–272.
11. Huang, X., J. Liu, Z. Han, and J. Yang. Privacy beyond sensitive values. *Science China Information Sciences*, Vol. 58, No. 7, 2015, pp. 1–15. doi:10.1007/s11432-014-5232-3.
12. Chen, Z., W. Ma, W. Lin, L. Chen, Y. Li, and B. Xu. A study on the changes of dynamic feature code when fixing bugs: towards the benefits and costs of Python dynamic features. *Science China Information Sciences*, Vol. 61, No. 1, 2017, pp. 012107–. doi:10.1007/s11432-017-9153-3.
13. Zhao, D. and D. Zheng. SMARTcleaner: identify and clean off-target signals in SMART ChIP-seq analysis. *BMC bioinformatics*, Vol. 19, No. 1, 2018, pp. 544–544. doi:10.1186/s12859-018-2577-4.
14. Liu, Q., Y. Guo, J. Wu, and G. Wang. Effective Query Grouping Strategy in Clouds. *Journal of Computer Science and Technology*, Vol. 32, No. 6, 2017, pp. 1231–1249. doi:10.1007/s11390-017-1797-9.
15. Poon, H., C. Quirk, C. DeZiel, and D. Heckerman. Literome: PubMed-scale genomic knowledge base in the cloud. *Bioinformatics (Oxford, England)*, Vol. 30, No. 19, 2014, pp. 2840–2842. doi:10.1093/bioinformatics/btu383.
16. Sun, Y. and M. Sarwat. A spatially-pruned vertex expansion operator in the Neo4j graph database system. *GeoInformatica*, Vol. 23, No. 3, 2019, pp. 397–423. doi:10.1007/s10707-019-00361-2.
17. O'Leary, D. E. SYSCO's best business practices (BBPs). *Journal of Information Technology Teaching Cases*, Vol. 3, No. 1, 2013, pp. 43–50. doi:10.1057/jittc.2012.6.
18. Jiang, L., S. Liu, and C. Chen. Recent research advances on interactive machine learning. *Journal of Visualization*, Vol. 22, No. 2, 2018, pp. 401–417. doi:10.1007/s12650-018-0531-1.
19. Shi, J., W. Ji, Z. Gao, G. Yujin, Y. Wang, L. Xinyi, and F. Shi. Ontology-based code snippets management in a cloud environment. *Journal of Ambient Intelligence and Humanized Computing*, Vol. 10, No. 8, 2018, pp. 2971–2985. doi:10.1007/s12652-018-0701-y.
20. Holst-Jensen, A., B. Spilsberg, A. J. Arulandhu, E. J. Kok, J. Shi, and J. Zel. Application of whole genome shotgun sequencing for detection and characterization of genetically modified organisms and derived products. *Analytical and bioanalytical chemistry*, Vol. 408, No. 17, 2016, pp. 4595–4614. doi:10.1007/s00216-016-9549-1.
21. Li, Y.-N., D.-J. Li, and K. Zhang. The impact of metaphors on information visualization. *Journal of Visualization*, Vol. 20,

No. 3, 2016, pp. 487–504. doi:10.1007/s12650-016-0371-9.

22. Zhou, W., Q. Chen, X.-B. Wang, T. O. Hughes, J. Liu, and X. Zhang. De novo assembly of the Platycladus orientalis (L.) Franco transcriptome provides insight into the development and pollination mechanism of female cone based on RNA-Seq data. *Scientific reports*, Vol. 9, No. 1, 2019, pp. 10191–10191. doi:10.1038/s41598-019-46696-6;10.1038/s41598-019-53777-z.
23. Plachkinova, M., A. Vo, B. Hilton, and R. Bhaskar. Response to Delamater's Comment on "A Conceptual Framework for Quality Healthcare Accessibility: A Scalable Approach for Big Data Technologies". *Information Systems Frontiers*, Vol. 20, No. 2, 2018, pp. 311–314. doi:10.1007/s10796-018-9842-y.
24. Abhishek, A. and A. Basu. A framework for disambiguation in ambiguous iconic environments. In *AI 2004: Advances in Artificial Intelligence: 17th Australian Joint Conference on Artificial Intelligence, Cairns, Australia, December 4-6, 2004. Proceedings 17*. Springer, 2005, pp. 1135–1140.
25. Feng, Y., F. Zarrinkalam, E. Bagheri, H. Fani, and F. N. Al-Obeidat. Entity linking of tweets based on dominant entity candidates. *Social Network Analysis and Mining*, Vol. 8, No. 1, 2018, pp. 46–. doi:10.1007/s13278-018-0523-0.
26. Chen, K., G. Ding, and J. Han. Attribute-based supervised deep learning model for action recognition. *Frontiers of Computer Science*, Vol. 11, No. 2, 2017, pp. 219–229. doi:10.1007/s11704-016-6066-5.
27. Mangel, M. Stochastic Dynamic Programming Illuminates the Link Between Environment, Physiology, and Evolution. *Bulletin of mathematical biology*, Vol. 77, No. 5, 2014, pp. 857–877. doi:10.1007/s11538-014-9973-3.
28. Tang, M., Z. Qiu, M. Yang, P. Cheng, S. Gao, S. Liu, and Q. Meng. Evolutionary ciphers against differential power analysis and differential fault analysis. *Science China Information Sciences*, Vol. 55, No. 11, 2012, pp. 2555–2569. doi:10.1007/s11432-012-4615-6.
29. Basu, A. et al. Iconic Interfaces for Assistive Communication. In *Encyclopedia of Human Computer Interaction*. IGI Global, 2006, pp. 295–302.
30. Shai, O., Y. Reich, A. Hatchuel, and E. Subrahmanian. Creativity and scientific discovery with infused design and its analysis with C–K theory. *Research in Engineering Design*, Vol. 24, No. 2, 2012, pp. 201–214. doi:10.1007/s00163-012-0137-x.
31. Fiscon, G., E. Weitschek, E. Cella, A. L. Presti, M. Giovanetti, M. Babakir-Mina, M. Ciotti, M. Ciccozzi, A. Pierangeli, P. Bertolazzi, and G. Felici. MISSEL: a method to identify a large number of small species-specific genomic subsequences and its application to viruses classification. *BioData mining*, Vol. 9, No. 1, 2016, pp. 38–38. doi:10.1186/s13040-016-0116-2.
32. Thorn, C. F., T. E. Klein, and R. B. Altman. PharmGKB: The Pharmacogenomics Knowledge Base. *Methods in molecular biology (Clifton, N.J.)*, Vol. 1015, No. 1, 2013, pp. 311–320. doi:10.1007/978-1-62703-435-7_20.
33. Novick, A. The fine structure of 'homology'. *Biology & Philosophy*, Vol. 33, No. 1, 2018, pp. 1–28. doi:10.1007/s10539-018-9617-3.
34. Lovell-Badge, R., N. Gonen, S. C. Samson, H. C. O'Neill, R. Sekido, and D. Maatouk. Regulation of Sox9 in the gonad during sex determination. *Transgenic research*, Vol. 25, No. 2, 2016, pp. 221–270. doi:10.1007/s11248-016-9936-6.
35. Huang, Z., W. Xue, and Q. Mao. Speech emotion recognitionwith unsupervised feature learning. *Frontiers of Information Technology & Electronic Engineering*, Vol. 16, No. 5, 2015, pp. 358–366. doi:10.1631/fitee.1400323.
36. Wang, F. and G. Luo. Guest editorial: special issue on data management and analytics for healthcare. *Distributed and Parallel Databases*, Vol. 37, No. 2, 2019, pp. 233–234. doi:10.1007/s10619-019-07269-8.
37. Paraskevopoulou, M. D., D. Karagkouni, I. S. Vlachos, S. Tastsoglou, and A. G. Hatzigeorgiou. microCLIP super learning framework uncovers functional transcriptome-wide miRNA interactions. *Nature communications*, Vol. 9, No. 1, 2018, pp. 3601–3601. doi:10.1038/s41467-018-06046-y.
38. Crowe, D., M. E. Lapierre, and M. Kebritchi. Knowledge Based Artificial Augmentation Intelligence Technology: Next Step in Academic Instructional Tools for Distance Learning. *TechTrends*, Vol. 61, No. 5, 2017, pp. 494–506. doi:10.1007/s11528-017-0210-4.
39. Sim, K., V. Gopalkrishnan, A. Zimek, and G. Cong. A survey on enhanced subspace clustering. *Data Mining and Knowledge Discovery*, Vol. 26, No. 2, 2012, pp. 332–397. doi:10.1007/s10618-012-0258-x.
40. Seo, D., H. Jung, W.-K. Sung, S. Kim, and S.-H. Lee. Development of Korean spine database and ontology for realizing e-Spine. *Cluster Computing*, Vol. 17, No. 3, 2014, pp. 1069–1080. doi:10.1007/s10586-013-0344-x.
41. Clark, T., P. Ciccarese, and C. Goble. Micropublications: a semantic model for claims, evidence, arguments and annotations in biomedical communications. *Journal of biomedical semantics*, Vol. 5, No. 1, 2014, pp. 28–28. doi:10.1186/2041-1480-5-28.
42. Sharma, A., M. Witbrock, and K. Goolsbey. Controlling search in very large commonsense knowledge bases: a machine learning approach. *arXiv preprint arXiv:1603.04402*.
43. Mahmoud, A. and G. Williams. Detecting, classifying, and tracing non-functional software requirements. *Requirements Engineering*, Vol. 21, No. 3, 2016, pp. 357–381. doi:10.1007/s00766-016-0252-8.
44. Liao, Y.-C. and P. H. Phan. Internal capabilities, external structural holes network positions, and knowledge creation. *The Journal of Technology Transfer*, Vol. 41, No. 5, 2015, pp. 1148–1167. doi:10.1007/s10961-015-9415-x.
45. Yılmaz, O., E. Yakıcı, and M. Karatas. A UAV location and routing problem with spatio-temporal synchronization constraints solved by ant colony optimization. *Journal of Heuristics*, Vol. 25, No. 4, 2018, pp. 673–701. doi:10.1007/s10732-018-9389-6.

46. Geminiani, A., A. Pedrocchi, E. D'Angelo, and C. Casellato. Extended generalized leaky integrate and fire neuron for cerebellum modeling. *BMC Neuroscience*, Vol. 18, No. 1, 2017, pp. 28–29. doi:10.1186/s12868-017-0371-2.
47. Xu, J., S. Gan, L. Song, Z. Ruan, S. Chen, Y. Wang, C. Gui, and B. Wan. Dish layouts analysis method for concentrative solar power plant. *SpringerPlus*, Vol. 5, No. 1, 2016, pp. 1850–1850. doi:10.1186/s40064-016-3540-3.
48. Wen, J., Z. Zhou, F. Lei, and J. Zhang. Basic and personalized pattern-based workflow fragments discovery. *Personal and Ubiquitous Computing*, Vol. 25, No. 6, 2019, pp. 1091–1111. doi:10.1007/s00779-019-01276-3.
49. Zhang, Y., Y. Huang, F. Kun, J. Song, and X. Qi. TextInsight: A new text visualization system based on entropy and GMap. *Journal of Electronics (China)*, Vol. 31, No. 5, 2014, pp. 453–464. doi:10.1007/s11767-014-4061-2.
50. Wei, S., Y. Zhao, T. Yang, Z. Zhou, and S. Ge. Enhancing heterogeneous similarity estimation via neighborhood reversibility. *Multimedia Tools and Applications*, Vol. 77, No. 1, 2017, pp. 1437–1452. doi:10.1007/s11042-017-4347-0.
51. Ravikumar, K. E., H. Liu, J. D. Cohn, M. E. Wall, and K. Verspoor. Literature mining of protein-residue associations with graph rules learned through distant supervision. *Journal of biomedical semantics*, Vol. 3, No. 3, 2012, pp. 1–23. doi:10.1186/2041-1480-3-s3-s2.
52. Pyysalo, S., T. Ohta, R. Rak, A. Rowley, H.-W. Chun, S.-J. Jung, S.-P. Choi, J. Tsujii, and S. Ananiadou. Overview of the Cancer Genetics and Pathway Curation tasks of BioNLP Shared Task 2013. *BMC bioinformatics*, Vol. 16, No. 10, 2015, pp. 1–19. doi:10.1186/1471-2105-16-s10-s2.
53. Bai, M., J. Xin, G. Wang, R. Zimmermann, and X. Wang. Skyline-join query processing in distributed databases. *Frontiers of Computer Science*, Vol. 10, No. 2, 2015, pp. 330–352. doi:10.1007/s11704-015-4534-y.
54. McCarthy, T. W., H.-C. Chou, and V. Brendel. SRAssembler: Selective Recursive local Assembly of homologous genomic regions. *BMC bioinformatics*, Vol. 20, No. 1, 2019, pp. 371–371. doi:10.1186/s12859-019-2949-4.
55. Mao, F., M. Ji, and T. Liu. Mining spatiotemporal patterns of urban dwellers from taxi trajectory data. *Frontiers of Earth Science*, Vol. 10, No. 2, 2015, pp. 205–221. doi:10.1007/s11707-015-0525-4.
56. Li, X., Z. Hu, and H. Wang. Combining Non-negative Matrix Factorization and Sparse Coding for Functional Brain Overlapping Community Detection. *Cognitive Computation*, Vol. 10, No. 6, 2018, pp. 991–1005. doi:10.1007/s12559-018-9585-6.
57. Chen, F., Z. Fu, and Z. Yang. Wind power generation fault diagnosis based on deep learning model in internet of things (IoT) with clusters. *Cluster Computing*, Vol. 22, No. 6, 2018, pp. 14013–14025. doi:10.1007/s10586-018-2171-6.
58. Zou, Q., Z. Wang, S. Luo, Y. Huang, and M. Tian. A computational coding model for saliency detection in primary visual cortex. *Chinese Science Bulletin*, Vol. 57, No. 30, 2012, pp. 3943–3952. doi:10.1007/s11434-012-5402-x.
59. Li, Z., Y. Gao, and Y. Lu. Efficient Evaluation of Monitoring Top-t Most Influential Places. *Wuhan University Journal of Natural Sciences*, Vol. 17, No. 1, 2012, pp. 25–30. doi:10.1007/s11859-012-0799-2.
60. Irawan, C. A. and D. Jones. Formulation and solution of a two-stage capacitated facility location problem with multilevel capacities. *Annals of Operations Research*, Vol. 272, No. 1, 2018, pp. 41–67. doi:10.1007/s10479-017-2741-7.
61. Shen, Y., K. Yuan, J. Dai, B. Tang, M. Yang, and K. Lei. KGDDS: A System for Drug-Drug Similarity Measure in Therapeutic Substitution based on Knowledge Graph Curation. *Journal of medical systems*, Vol. 43, No. 4, 2019, pp. 92–92. doi:10.1007/s10916-019-1182-z.
62. Harbi, R., I. Abdelaziz, P. Kalnis, N. Mamoulis, Y. Ebrahim, and M. Sahli. Accelerating SPARQL queries by exploiting hash-based locality and adaptive partitioning. *The VLDB Journal*, Vol. 25, No. 3, 2016, pp. 355–380. doi:10.1007/s00778-016-0420-y.
63. Wu, Y., C. Wang, Y. qing Zhang, and J. jun Bu. Unsupervised feature selection via joint local learning and group sparse regression. *Frontiers of Information Technology & Electronic Engineering*, Vol. 20, No. 4, 2019, pp. 538–553. doi:10.1631/fitee.1700804.
64. Yang, T., G. Jin, and J. Zhu. Automated design of freeform imaging systems. *Light, science & applications*, Vol. 6, No. 10, 2017, pp. e17081–e17081. doi:10.1038/lsa.2017.81.
65. Saleem, M., S. S. Padmanabhuni, A.-C. N. Ngomo, A. Iqbal, J. S. Almeida, S. Decker, and H. F. Deus. TopFed: TCGA tailored federated query processing and linking to LOD. *Journal of biomedical semantics*, Vol. 5, No. 1, 2014, pp. 47–47. doi:10.1186/2041-1480-5-47.
66. Graham, S. and S. Weingart. The Equifinality of Archaeological Networks: an Agent-Based Exploratory Lab Approach. *Journal of Archaeological Method and Theory*, Vol. 22, No. 1, 2014, pp. 248–274. doi:10.1007/s10816-014-9230-y.
67. Savir, Y., J. J. Kagan, and T. Tlusty. Binding of Transcription Factors Adapts to Resolve Information-Energy Tradeoff. *Journal of Statistical Physics*, Vol. 162, No. 5, 2015, pp. 1383–1394. doi:10.1007/s10955-015-1388-5.
68. Wang, Z., J. Zhu, Y. Xue, C. Song, and N. Bi. Cell recognition based on topological sparse coding for microscopy imaging of focused ultrasound treatment. *BMC medical imaging*, Vol. 15, No. 1, 2015, pp. 46–46. doi:10.1186/s12880-015-0087-7.
69. Popic, V. and S. Batzoglou. A hybrid cloud read aligner based on MinHash and kmer voting that preserves privacy. *Nature communications*, Vol. 8, No. 1, 2017, pp. 15311–15311. doi:10.1038/ncomms15311.
70. He, L., B. Liu, G. Li, Y. Sheng, Y. Wang, and Z. Xu. Knowledge Base Completion by Variational Bayesian Neural Tensor Decomposition. *Cognitive Computation*, Vol. 10, No. 6, 2018, pp. 1075–1084. doi:10.1007/s12559-018-9565-x.
71. Sharma, A. and K. M. Goolsbey. Learning Search Policies in Large Commonsense Knowledge Bases by Randomized

Exploration.

72. Li, X., S. Zhang, R. Huang, B. Huang, C. Xu, and Y. Zhang. A survey of knowledge representation methods and applications in machining process planning. *The International Journal of Advanced Manufacturing Technology*, Vol. 98, No. 9, 2018, pp. 3041–3059. doi:10.1007/s00170-018-2433-8.
73. Comoglio, F., C. Sievers, and R. Paro. Sensitive and highly resolved identification of RNA-protein interaction sites in PAR-CLIP data. *BMC bioinformatics*, Vol. 16, No. 1, 2015, pp. 32–32. doi:10.1186/s12859-015-0470-y.
74. Chodpathumwan, Y., A. Vakilian, A. Termehchy, and A. Nayyeri. Cost-effective conceptual design using taxonomies. *The VLDB Journal*, Vol. 27, No. 3, 2018, pp. 369–394. doi:10.1007/s00778-018-0501-1.
75. Yang, Z., Q. Li, W. Liu, Y. Ma, and M. Cheng. Dual graph regularized NMF model for social event detection from Flickr data. *World Wide Web*, Vol. 20, No. 5, 2016, pp. 995–1015. doi:10.1007/s11280-016-0405-1.
76. Dash, J. K. and S. Mukhopadhyay. Similarity learning for texture image retrieval using multiple classifier system. *Multimedia Tools and Applications*, Vol. 77, No. 1, 2016, pp. 459–483. doi:10.1007/s11042-016-4228-y.
77. d'Acierno, A., M. Esposito, and G. D. Pietro. An extensible six-step methodology to automatically generate fuzzy DSSs for diagnostic applications. *BMC bioinformatics*, Vol. 14, No. 1, 2013, pp. 1–19. doi:10.1186/1471-2105-14-s1-s4.
78. Catena, M. and N. Tonellotto. Energy-Efficient Query Processing in Web Search Engines. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 29, No. 7, 2017, pp. 1412–1425. doi:10.1109/tkde.2017.2681279.
79. Khan, F. H., U. Qamar, and S. Bashir. Multi-Objective Model Selection (MOMS)-based Semi-Supervised Framework for Sentiment Analysis. *Cognitive Computation*, Vol. 8, No. 4, 2016, pp. 614–628. doi:10.1007/s12559-016-9386-8.
80. Yu, Q. QoS-aware service selection via collaborative QoS evaluation. *World Wide Web*, Vol. 17, No. 1, 2012, pp. 33–57. doi:10.1007/s11280-012-0186-0.
81. Bell, A. S., J. Bradley, J. R. Everett, J. Loesel, D. McLoughlin, J. E. J. Mills, M.-C. Peakman, R. E. Sharp, C. Williams, and H. Zhu. Plate-based diversity subset screening generation 2: an improved paradigm for high-throughput screening of large compound files. *Molecular diversity*, Vol. 20, No. 4, 2016, pp. 789–803. doi:10.1007/s11030-016-9692-9.
82. Nguyen, H. H., J. Park, S. Hwang, O. S. Kwon, C.-S. Lee, Y.-B. Shin, T. H. Ha, and M. Kim. On-Chip Fluorescence Switching System for Constructing a Rewritable Random Access Data Storage Device. *Scientific reports*, Vol. 8, No. 1, 2018, pp. 337–337. doi:10.1038/s41598-017-16535-7.
83. Jiang, Z., C. Qian, K. Zhao, S. Chen, R. Li, X. Wang, C. He, and J. Du. VariSecure: Facial Appearance Variance based Secure Device Pairing. *Mobile Networks and Applications*, Vol. 26, No. 2, 2019, pp. 870–883. doi:10.1007/s11036-019-01330-7.
84. Sharma, A., K. M. Goolsbey, and D. Schneider. Disambiguation for Semi-Supervised Extraction of Complex Relations in Large Commonsense Knowledge Bases. In *7th Annual Conference on Advances in Cognitive Systems*. 2019.
85. Zhao, J., N. Sun, and W. Cheng. Logistics forum based prediction on stock index using intelligent data analysis and processing of online web posts. *Journal of Ambient Intelligence and Humanized Computing*, Vol. 11, No. 9, 2019, pp. 3575–3584. doi:10.1007/s12652-019-01520-x.
86. Mohammed, S., E.-S. M. El-Alfy, and A. F. Barradah. Improved selectivity estimator for XML queries based on structural synopsis. *World Wide Web*, Vol. 18, No. 4, 2014, pp. 1123–1144. doi:10.1007/s11280-014-0311-3.
87. Kung, S.-Y. Discriminant component analysis for privacy protection and visualization of big data. *Multimedia Tools and Applications*, Vol. 76, No. 3, 2015, pp. 3999–4034. doi:10.1007/s11042-015-2959-9.