SQUIRE+LLM: An LLM-Assisted ESA-Based Unsupervised Semantic Query Interpretation and Ranking Engine

Avijeet Shil
Department of Computer Science and Engineering
University of North Texas
Denton, Texas
avijeetshil@my.unt.edu

Wei Jin
Department of Computer Science and Engineering
University of North Texas
Denton, Texas
wei.jin@unt.edu

***Abstract*—Mapping and translating user queries into relevant entities is a core problem in the information retrieval field. This paper introduces SQUIRE+LLM, an unsupervised framework capable of processing natural language queries. The framework uses a large language model (LLM) to normalize the user query. Then it decomposes the user query into n-gram phrases and matches them against Wikipedia titles. It involves constructing query-term combinations and ranking the top 3 Wikipedia entities using Explicit Semantic Analysis (ESA) based on each query-term combination. Then, within these top 3 entities, the system ranks the top 3 non-overlapping query-term combinations. Unlike prior work, SQUIRE+LLM does not depend on query expansion, syntactic parsing, or supervised learning. Instead, it leverages Wikipedia's concept space, containing titles, redirects, and disambiguation entities, to normalize and disambiguate entities. Redirected entities are resolved by recursively following redirect links until the canonical entity is reached, while disambiguation is resolved by selecting the related Wikipedia entity based on cosine similarity in the ESA space. The evaluation of SQUIRE+LLM is done using TREC questions, with SQUIRE+LLM performance compared to two baselines and a context-based question translation model (CBQT) and its expansion variant CBQE using Google search outputs. SQUIRE+LLM consistently outperforms both baseline methods, with notable improvements in complex question categories. It reduces zero-score MRR cases by 75% and improves MRR by up to 400% compared to baseline on "Why" questions. Compared to the Greedy baseline, it achieves significant gains of up to 170% for "Who" and 184% for "Name the" questions—without explicitly specifying question types. The results show that ESA-based semantic ranking, enhanced by LLM-guided query normalization, is a powerful and interpretable approach for constructing transparent and domain-independent natural language interfaces.**

***Index Terms*—Semantic Query Interpretation, Explicit Semantic Analysis, Information Retrieval, Wikipedia**

I. Introduction

Following the COVID-19 pandemic, the world has dramatically shifted its reliance on digital platforms for information, education, and remote services. Users now frequently use multiple-entity natural language queries complex enough to cover a range of topics, from health searches to academic and technical searches. As a result, they expect systems to produce results that are correct and semantically relevant. This shift has once again highlighted the weaknesses of traditional keyword search models.

Historically, search engines used a bag-of-words (BOW) model in which the searched documents and user queries are viewed as unordered collections of keywords. In practice, the bag-of-words model works well when users need short, exact matches of keywords; however, BOW models don't take into consideration the semantics, context, and order of words, which are critical in interpreting written natural language.

To address these challenges, various academic approaches have incorporated structured external knowledge, such as Wikipedia, into query understanding. One such approach is the CBQT-based context graph framework [1], which extracts Wikipedia entities from questions and organizes them using Wikipedia's link structure. However, this method depends heavily on graph connectivity heuristics and does not explicitly model the semantic coherence of query interpretations.

To incorporate semantic coherence, we propose a fully unsupervised, Wikipedia concept-based approach to natural language query understanding using Explicit Semantic Analysis (ESA). Unlike most previous works that limit themselves to adding related terms to the original query as an query expansion method [2], our method builds and evaluates n-gram combinations of query term phrases, then selects the most semantically coherent query interpretation in a high-dimensional Wikipedia concept space.

Our approach involves the following steps:

1) Query Normalization With LLM: We apply a large language model (LLM), specifically ChatGPT (GPT-4o), to clean and normalize user-generated queries. This includes correcting spelling, punctuation, and spacing errors. This step ensures more consistent downstream processing, especially when dealing with noisy or user-generated input, and increases the accuracy and coverage of entity linking and retrieval.
2) Query Term Extraction and Wikipedia Entity Linking: We identify n-gram phrases from the normalized query and attempt to map each phrase to a corresponding

Wikipedia article. This step filters out irrelevant phrases by ensuring that only those with valid Wikipedia entity links are retained. Redirects and disambiguation pages are handled to improve Wikipedia entity linkage coverage and accuracy.

3) Disambiguation Handling: If a phrase maps to a disambiguation page, we retrieve all linked Wikipedia candidate entities. Then we compute the TF-IDF-based cosine similarity score between the normalized query and the title of each Wikipedia candidate entity. The entity with the highest score is selected as the most relevant.
4) Query Interpretation Generation: Based on the retrieved linked Wikipedia entities, we build all valid non-overlapping combinations of phrases found in the original query. These combinations represent alternative interpretations of user intent.
5) Semantic Ranking with ESA: Each query-term combination is converted to a semantic vector using ESA and ranked by cosine similarity to relevant Wikipedia semantic interpretation vectors associated with matched Wikipedia entities. The best interpretation of the query is determined by averaging the similarity scores of the highest-ranked matching Wikipedia entities, followed by considering non-overlapping, highly ranked query-term combinations. Notably, all potential query interpretations are derived from the normalized query and aim to preserve the semantic intent of the user's original query.
6) Document Retrieval: The selected query interpretation is used to feed into an external search engine, Google, to retrieve relevant documents from a refined and semantically grounded version of the user query. It increases the likelihood of retrieving accurate and contextually appropriate search results.

This framework offers several key advantages:

1) It is fully unsupervised, requiring no annotated training data or deep model fine-tuning.
2) It robustly resolves ambiguity using semantic context rather than heuristic graph traversal.
3) It enables interpretability by grounding results in Wikipedia concepts.
4) It supports scalable retrieval via inverted indexes and sparse ESA vectors.
5) It improves input robustness by using a large language model (ChatGPT) to correct spelling errors and normalize user queries, enhancing Wikipedia entity matching precision without requiring handcrafted rules.

We evaluate our approach on downstreaming tasks on TREC QA dataset and demonstrate that it consistently outperforms standard keyword-based baselines and heuristic graph approaches in terms of Wikipedia entity resolution accuracy and Mean Reciprocal Rank (MRR). These results affirm that ranking query interpretations by semantic coherence, rather than adding isolated expansion terms, provides a more effective foundation for natural language interfaces.

II. Related Work

Our work bridges the areas of semantic query interpretation, entity disambiguation and linking, concept-based representation, user-centric ranking, and preference modeling. Unlike approaches based on dense embeddings, we use ESA-based representations and Wikipedia matching to resolve and rank entity interpretations directly.

A. Entity Disambiguation and Linking.

Entity disambiguation has seen a range of approaches, from supervised learning [3]–[7] to large-scale, weakly supervised linking [8]. Recent efforts incorporate language models like BERT for contextualized entity linking [9], [10], especially with spelling correction [11]. While effective, these models often require large training datasets and lack interpretability. In contrast, we use Wikipedia title and redirect matching combined with ESA-based disambiguation, allowing transparent, domain-independent resolution without supervised training.

B. Concept-Based Representation via ESA.

ESA [12] introduced a sparse, interpretable semantic space using Wikipedia concepts. Although neural embeddings like Word2Vec [13], graph embedding [14] or BERT [15], [16] have become dominant, ESA remains attractive for systems where concept-level reasoning and interpretability are required. Though ESA is a classic method, multiple studies have revisited its use in semantic retrieval settings [17]. For example, Bogdanova et al. [18] applied ESA to job-profile relevance in LinkedIn to evaluate a candidate's skillset. It has also been used in concept-based ranking and semantic filtering tasks, highlighting its continued relevance. In our work, it enables direct similarity computation between user queries and Wikipedia-derived documents, providing an interpretable and domain-independent framework for semantic matching.

C. Semantic Query Interpretation.

Research in query understanding has shifted from lexical to semantic modeling. Approaches such as query rewriting using large language models [19]–[21] and context-aware query reformulation [22], [23] often rely on dense neural retrieval. Our method avoids query rewriting altogether. Instead, we generate and rank all valid query-term combinations via ESA-based coherence and offer an interpretable and explainable approach alternative to deep learning-based black-box approaches.

D. User-Centric Ranking and Preference Modeling

Modeling user preferences in ranking has gained increasing attention, particularly in interactive and context-aware information retrieval [24], [25]. Jiang [26] compared directional and linear top-k queries, showing that users in lifestyle-oriented domains prefer more balanced trade-offs across attributes, while users in personal-interest domains exhibit more subjective and varied preferences. These findings align with our observation that semantic coherence, as computed via ESA, more accurately captures user-perceived relevance than literal keyword overlap.

III. METHODOLOGY

We propose SQUIRE+LLM, an LLM-assisted Semantic Query Interpretation and Ranking Engine, a fully unsupervised framework for understanding and interpreting natural language questions by identifying Wikipedia entity-aligned query terms. Figure 1 shows the end-to-end approach for extracting search results. Given a user question, the system begins with zero-shot query normalization, fixing the query using LLM and preprocessing steps (removing stop words, lemmatization, etc.), and then identifying potential Wikipedia entities through phrase-level matching with Wikipedia article titles. In contrast to traditional named entity recognition pipelines based on a pre-computed taxonomy (e.g., Person, Location, Organization) or graph presentation [1], [4], our method operates in Wikipedia's large and dynamically evolving concept space. Each Wikipedia article serves as a proxy for a real-world entity and encapsulates its most important attributes and contexts. The framework is designed to enable the system to detect and disambiguate entities across diverse domains, ranging from geopolitical regions and historical events to scientific theories, without relying on handcrafted rules or supervised training data. The set of concept-based Wikipedia entities ultimately provides a set of semantically based labels that can and will allow for downstream interpretation, disambiguation, and development by using ESA.

SQUIRE mainly consists of the following five steps:

1) Query Normalization with LLM
2) Wikipedia Entity Matching and Disambiguation Handling.
3) Query Interpretation Generation.
4) ESA-Based Semantic Representation.
5) Generation of Non-overlapping Top-Ranked Query Term Combinations.
6) Document Retrieval

A. Query Normalization with LLM

This step is stage 0 of the pipeline and serves as a critical part of preprocessing. It focuses on query normalization to enhance the quality of user input before semantic processing. We employ a large language model (LLM), specifically ChatGPT (GPT-4o), in a zero-shot setting to avoid introducing bias in the refined query. The role of LLM is a context-aware spell corrector, addressing spelling errors, inconsistent spacing or punctuation, and reformatting informal or unstructured input.

Unlike traditional rule-based spelling correction approaches, such as TextBlob, based on edit distance and dictionaries, the LLM leverages sentence-level context to make more accurate and semantically informed corrections. This normalization is useful for reducing lexical mismatches between user queries and Wikipedia article titles. For example, a query like "Where is Los Vegas?" is corrected to "Where is Las Vegas?", enabling accurate entity recognition downstream. Additional examples are illustrated in table III. The normalization process is fully unsupervised and does not rely on handcrafted dictionaries, domain-specific constraints, or labeled training data.

Following this, we apply token-level normalization to prepare queries for Wikipedia entity matching. Each token is POS-tagged using NLTK and then processed as follows: Stop words are removed; ordinal phrases ("21st," "3rd") are turned into their numeric root form using regular expression matching; proper names and special words are preserved in lowercase without lemmatization for Wikipedia entity-specific signal preservation, and other tokens are lemmatized according to WordNet, with POS tags being mapped correctly. For certain cases, these rules play a crucial role in finding Wikipedia entities that represent the user's intent. This preprocessing ensures consistent, semantically rich input that supports accurate disambiguation and ranking in later stages.

B. Wikipedia Entity Matching and Disambiguation Handling

After the preprocessing step, this process begins with processing the processed query to extract all possible n-gram sequences up to a predefined maximum length (typically, three). Each n-gram phrase is then mapped to the Wikipedia concept space that includes article information such as titles, contents, redirect pages, and disambiguation pages.

To enhance the semantic relatedness and increase the number of Wikipedia entities, the system incorporates redirect pages (which often point to canonical entities) and disambiguation page titles (DPTs). If any term refers to any potential Wikipedia entity, we consider the redirected Wikipedia entity for later computation, as the redirected entity contains the most relevant information about the potential entity. Redirects are recursively resolved to their canonical titles, ensuring the normalization of variant or synonymous expressions (e.g.,"Oscars" →"Academy Awards"). A disambiguation page on Wikipedia is a page that lists multiple articles that could be associated with the same term or phrase. This is used when a particular word or title has more than one meaning or refers to multiple topics. Such pages typically have a title like "X (disambiguation)," where X is the title of the Wikipedia title. For a disambiguation page, the content of the entity is a list of entities that may match the entity in the different contexts. For example, in "Flower (disambiguation),"[1] the Wikipedia entity refers to multiple entities in different contexts, such as a people surname, a musical band[2], or a community[3]. Hence, it is important to identify which "Flower" entity must be considered as a candidate Wikipedia entity.

To resolve this, for phrases matching a disambiguation page, all linked Wikipedia entities are extracted, and ambiguity is resolved using TF-IDF-based cosine similarity score. To determine the most contextually appropriate interpretation, we compute the cosine similarity between the TF-IDF vector of the user query and the vector representation of each Wikipedia article's title. The most semantically similar entities are retained, whereas the rest are discarded. If the highest similarity score falls below a predefined threshold (e.g., 0.2), we default to selecting the first Wikipedia title listed on the

[1]https://en.wikipedia.org/wiki/Flower_(disambiguation)
[2]https://en.wikipedia.org/wiki/Flower_(American_band)
[3]https://en.wikipedia.org/wiki/Flower,_West_Virginia

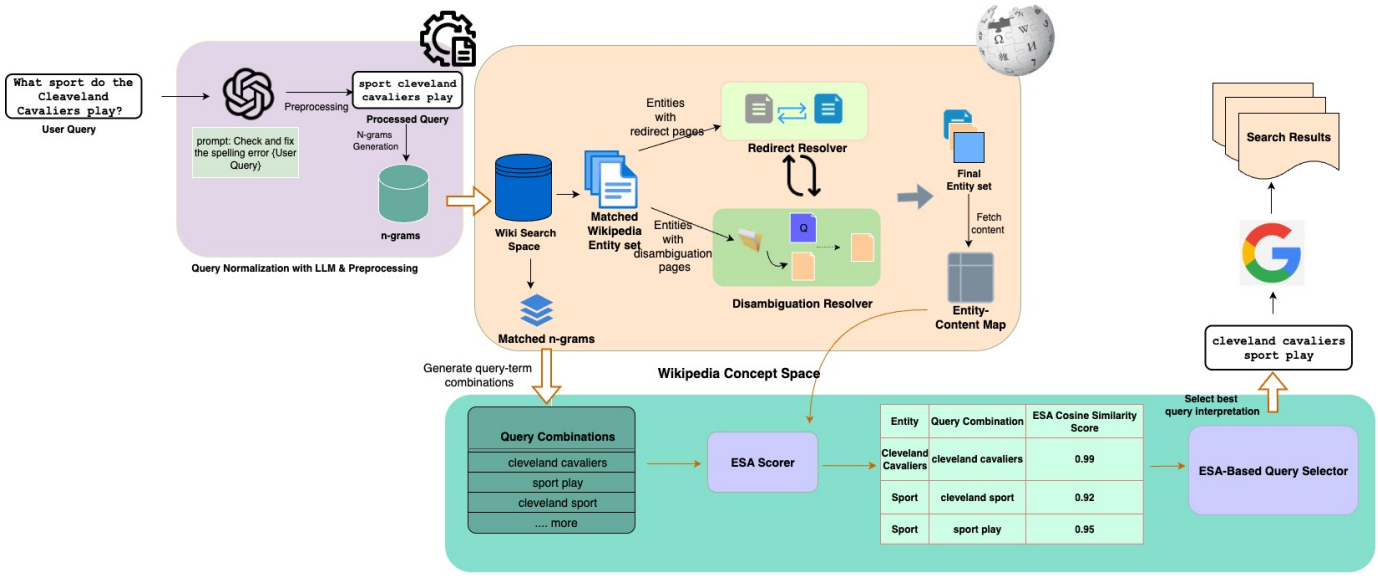


Fig. 1. The SQUIRE+LLM pipeline for processing a natural language query. The illustrated example shows a real output generated by the system, demonstrating query normalization, entity linking, semantic interpretation generation, and ranking.

disambiguation page, assuming it to be the most common usage of the phrase. Notably, only a few cases fell below this threshold, and we manually evaluated the relevance of the selected entities during threshold calibration. This step ensures that only contextually relevant Wikipedia entities are passed forward, which is particularly important when dealing with ambiguous terms such as "Flower."

In real-world scenarios, a page may serve both as a redirect and have links to a disambiguation page, depending on the context. For example, "United States" is the canonical Wikipedia entity for several redirects, such as "US" and "USA," and is also linked from "America (disambiguation)." These are resolved by recursively identifying the final canonical Wikipedia entity while retaining all potentially linked Wikipedia entities for disambiguation, if applicable. The system ensures and handles hybrid cases effectively while maintaining both correctness and coverage.

C. Query Interpretation Generation

This step is crucial for refining the phrase set and retaining the top-ranked Wikipedia entities that contribute to the major concepts of the user query. As described in Step III-B, we filter out unnecessary n-grams to construct a focused set of matched Wikipedia entities at both the word and phrase levels.

Following entity matching, we generate all valid combinations of query-terms/phrases that have corresponding matched Wikipedia entities. A query-term combination refers to a contiguous sequence of words (an n-gram) extracted from the input query. For instance, for the normalized query after LLM, *"What sport do the Cleveland Cavaliers play?"*, some examples of query-term combinations can be "cleveland cavaliers" or "cleveland play." A query interpretation, in turn, is defined as a set of non-overlapping query-term combinations that together capture the semantic meaning of the original query. The query may yield interpretations such as $\{$*cleveland cavaliers, sport play*$\}$. Each interpretation represents a different potential structure for understanding the query's core concepts and is evaluated independently in later steps.

D. ESA-Based Semantic Representation

ESA [12] is a concept-based semantic representation technique that maps text, for our problem, user query, into a high-dimensional space of Wikipedia concepts. Each dimension corresponds to a Wikipedia article, and the relevance of that concept to the input text is quantified using TF-IDF weighting based on the shared term set.

To determine which query-term combination most accurately reflects the user's intent, the semantic coherence of each combination is evaluated using the semantic interpretation vectors generated by ESA.

Unlike dense neural embeddings, ESA produces human interpretable, making it well-suited for tasks such as entity disambiguation and semantic ranking of documents. In our framework, ESA is used to encode both the query-term combinations and the corresponding Wikipedia entity documents. Each combination is represented as a sparse ESA vector that captures its distribution over Wikipedia concepts. Unlike Jaccard Similarity, which is token-based and fails to capture deeper semantic relationships, ESA provides a semantic coherence evaluation between query phrases and Wikipedia entities.

To formulate ESA for our problem, let T denote a query-term combination, and let $D = \{d_1, d_2, \ldots, d_N\}$ be the set of Wikipedia articles used as concepts, where each d_j corresponds to a distinct article. We maintain the full content of each article, as using only the title may not provide sufficient context. Let w_i be a word in T, and let $\vec{v}_T \in \mathbb{R}^N$ denote the ESA vector representation of T.

TABLE I
STATISTICS OF QUERY-TERM COMBINATIONS GENERATED PER QUESTION.

Statistic	#Entities/Question	#Query Terms/Question	#Query–Doc Pairs/Question
Average	3.11	3.11	2414.50
Maximum	14	14	917,266
Minimum	1	1	0

Let TFIDF(w_i) represent the TF-IDF weight of word w_i in the query term combination T. Let k_j be an inverted index entry for word w_i, where k_j quantifies the strength of association between word w_i and Wikipedia article/concept d_j, with $d_j \in D(i.e., d_1, d_2, ..., d_N)$, where N is the total number of Wikipedia concepts. Then, the ESA semantic interpretation vector $\vec{v}_T$ is computed as

$$\vec{v}_T[j] = \sum_{w_i \in T} \text{TFIDF}(w_i) * (k_j) \quad (1)$$

$$\vec{v}_T = \frac{\vec{v}_T}{\|\vec{v}_T\|} \quad (2)$$

Each component $\vec{v}_T[j]$ reflects the relevance of the j^{th} Wikipedia article to the text ofthe query-termm combination T. Since most articles contain only a small subset of vocabulary terms, the resulting vector $\vec{v}_T$ is typically sparse.

E. Generation of Top-Ranked Non-overlapping Query-Term Combinations

To evaluate semantic coherence, we compute the cosine similarity between possible query-term combinations and each matched Wikipedia entity; we use Explicit Semantic Analysis (ESA). ESA represents both textual fragments and Wikipedia concepts as high-dimensional concept vectors derived from Wikipedia. The cosine similarity between these vectors represents their semantic relatedness.

For each query-term combination, we compute an ESA vector $\vec{v}_Q$. Similarly, for each matched Wikipedia entity, we compute its ESA vector $\vec{v}_D$. We calculate cosine similarity between each query-term combination and each matched Wikipedia entity using:

$$\text{Sim}(\vec{v}_Q, \vec{v}_D) = \frac{\vec{v}_Q \cdot \vec{v}_D}{\|\vec{v}_Q\| \cdot \|\vec{v}_D\|}$$

For each query-term combination, we compute the cosine similarity scores between the combination and all matched Wikipedia entities, recording these scores in an ESA Cosine Similarity Score Table. For example, the Wikipedia entity "Cleveland Cavaliers" and the query-term combination "cleveland sport" are compared. The final step involves a two-step sorting process: first sorting at the entity level and then sorting at the query-term combination level. After calculating the similarity scores, we sort them and select the top-3 Wikipedia entities. Next, from the highly ranked Wikipedia entities and their corresponding query-term combinations, we aggregate the similarity scores. This is typically done by averaging the scores across all query-term combinations. To further refine the results, we sort the combinations by phrase length, prioritizing longer combinations to enhance specificity. Lastly, we generate top-3 non-overlapping query-term combinations, which are more precise and semantically relevant. The purpose of this step is to capture the best query interpretation that reflects the overall context of the original query.

F. Document Retrieval

This query then is passed to an external search engine (in our case, Google), which retrieves relevant documents or search results. The search engine is independent and robust enough to evaluate the query interpretation and return results that closely align with the original query's intent.

IV. EXPERIMENT DETAILS

A. Data

We use the English Wikipedia dump[4] as our knowledge base. We process the dump in a way to retrieve all article metadata, including titles, full content, redirect information, and disambiguation page flags, which are indexed in SQLite, a free and open-source relational database engine. Additionally, internal page links are collected. During preprocessing, we specifically remove non-article content such as templates (e.g., Template:), user pages (e.g., User:), user talk pages (e.g., User talk:), Wikipedia management pages (e.g., Wikipedia:), category pages (e.g., Category:), and help documents (e.g., Help:). This ensures that only relevant article data is retained.

As our task is in the context of open-domain question answering, we evaluate our system using the TREC[5] question collection on a random set of 893 questions, which contains natural language queries. These questions are manually classified into ten categories based on their question types, with the distribution as shown in table II. A few outdated questions are excluded from the dataset, typically $1-2$ per category.

To assess the effectiveness of our query translation pipeline, we remove the question type, reduce biases, and aim to accurately map natural language queries to their semantically equivalent keyword queries and evaluate performance based on the system's ability to retrieve correct answers. We use Google as the backend retrieval engine. Our system selects the highest-ranked query interpretation and submits it to the search engine, from which we extract the top-ranked results.

We apply an n-gram-based Wikipedia entity matching strategy, extracting all candidate phrases up to 3-grams from each

[4]https://dumps.wikimedia.org/

[5]https://trec.nist.gov/data/qa.html

TABLE II
STATISTICS OF THE TREC QUESTION COLLECTION.

Question Type	#Questions	Example
"What"	284	"What is the tallest building in Japan?"
"What XX"	168	"What time of day did Emperor Hirohito die?"
"Who"	165	"Who wrote 'Hamlet'?"
"Where"	92	"Where is Logan Airport?"
"How XX"	80	"How many people live in Tokyo?"
"When"	67	"When was Nostradamus born?"
"Name a"	20	"Name a symptom of mononucleosis."
"Name the"	8	"Name the highest mountain."
"How"	5	"How many islands does Fiji have?"
"Why"	4	"Why is Jane Goodall famous?"

TABLE III
QUERY NORMALIZATION EXAMPLES: SPELLING AND ENTITY LINKING CORRECTIONS

Original Question	Refined Question	Entities	Impact
Where did bocci originate?	Where did bocce originate?	Bocce	Corrects key verb spelling; enables proper Wikipedia article match.
What sport do the Cleaveland Cavaliers play?	What sport do the Cleveland Cavaliers play?	Sport, Cleveland, Cavaliers	Team/city name error; affects correct sports info retrieval.
What is Pittsburg's baseball team called?	What is Pittsburgh's baseball team called?	Baseball, Pittsburgh	Different city; prevents matching with actual team (Pirates).
Where is McCarren Airport?	Where is McCarran Airport?	McCarran International Airport	Misspelling affects major airport name and searchability.
Who was Samuel Johnsons's friend and biographer?	Who was Samuel Johnson's friend and biographer?	Samuel Johnson	Possessive correction enables proper entity linking and historical relation retrieval.

TABLE IV
MEAN RECIPROCAL RANK (MRR) ACROSS DIFFERENT QUESTION TYPES FOR GREEDY APPROACH, BASELINE, CBQT, CBQT+CBQE(LDA), SQUIRE, AND SQUIRE+LLM. THE BEST SCORE FOR EACH QUESTION TYPE IS HIGHLIGHTED IN BOLD.

Method	"What"	"What XX"	"Who"	"Where"	"How XX"	"When"	"Name a"	"Name the"	"How"	"Why"	All Questions
Greedy baseline	0.29	0.36	0.23	0.33	0.13	0.38	0.30	0.25	0.25	0.17	0.29
Baseline [1]	0.37	0.40	0.51	0.39	0.16	0.29	0.38	0.55	0.27	0.05	0.38
CBQT [1]	0.45	0.48	0.56	0.47	0.13	0.42	0.44	0.52	0.51	0.11	0.45
CBQT+CBQE (LDA) [1]	0.46	0.50	0.60	**0.53**	0.12	0.42	0.47	0.58	0.51	0.11	0.46
SQUIRE	0.56	0.51	**0.62**	0.47	0.21	**0.51**	**0.48**	**0.71**	**0.54**	**0.25**	**0.51**
SQUIRE+LLM	**0.57**	**0.53**	**0.62**	0.48	**0.22**	**0.51**	**0.48**	**0.71**	**0.54**	**0.25**	**0.51**

query. Each resulting query-term combination is evaluated and ranked. For every original question, we retrieve the top $N = 10$ search results and analyze the returned titles and snippets to evaluate whether the correct answer appears. TREC provides gold-standard answers and enables a direct and quantitative evaluation of the translation pipeline's accuracy.

B. Experimental Strategies

We evaluate our system using search results as the document set. A document is considered relevant if it contains the correct answer to the given natural language question. Our goal is to identify and retain query terms from highly ranked non-overlapping query-term combinations that lead to the most prominently ranked Wikipedia entities in the results. This ensures both semantic specificity and minimal redundancy.

In our initial observation, we design a pipeline where we consider the sentence may contain spelling errors. We find 2.2% of the questions contain spelling errors; some of them are listed in table III, for instance, "Where is McCarren Airport?". Table III illustrates a reasonable explanation of the need for spelling correction in the early stage.

To evaluate the impact of the LLM component on the performance of SQUIRE, we conducted an ablation study comparing the framework with (SQUIRE+LLM) and without (SQUIRE) the ChatGPT-based spelling corrector. The results are presented in Table IV. We observed an improvement in the Mean Reciprocal Rank (MRR) for four categories ("What", "What XX", "Where", "How XX"), as well as for all questions containing spelling errors. Specifically, the relative gain in mean MRR was 0.4, while the median MRR showed a substantial increase of 1.0. The inclusion of the LLM significantly enhances SQUIRE's performance, lifting the overall MRR and reducing zero-score MRR cases by up to 75%.

During preprocessing, we store all Wikipedia titles that match any extracted n-gram from the user query for future

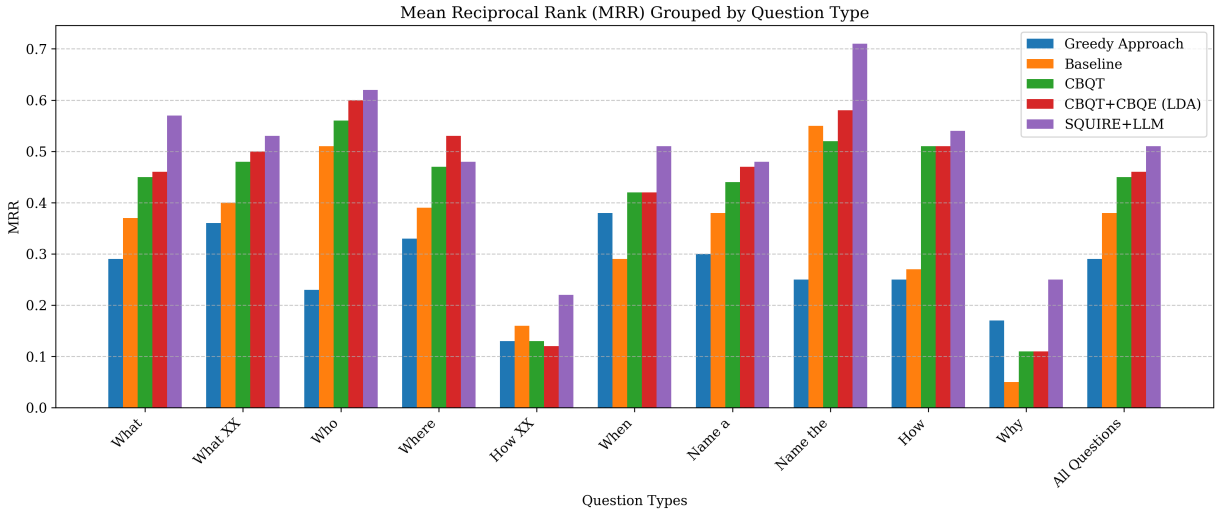


Fig. 2. MRR comparison across question types. **SQUIRE+LLM** consistently outperforms the greedy approach, CBQT, CBQT+CBQE (LDA), and the baseline in nearly all categories. We use the paired t-test with significance at $p < 0.01$.

ESA-based computation. However, given the size and scale of Wikipedia, treating every article as a candidate Wikipedia entity for ESA matching is computationally expensive and bandwidth-intensive. We cache the titles and full text of Wikipedia articles while indexing as well as semantic vectors of each wikipedia article for faster computation and most ESA-heavy computations are offline and cached to support scalability. Therefore, rather than applying ESA over the entire Wikipedia concept space, we constrain our candidate set to only those entities whose titles match n-grams from the original query case-insensitively. This significantly improves efficiency without sacrificing relevance. Furthermore, we intentionally preserve the naturalness of query terms to reflect realistic user input patterns. We observe that the best query interpretation derived from the highest-ranked Wikipedia entities is not only semantically coherent but also strongly aligned with the actual answer, thereby improving search effectiveness. For relevance judgment, a document (i.e., a search result) is considered correct if either its title or snippet matches the TREC-provided answer pattern, defined as a regular expression. We adopt Mean Reciprocal Rank (MRR) as our evaluation metric, computed over the top 10 search results per query. This cutoff is based on common user behavior and prior studies, as performance gain diminishes beyond the first page of Google search results. To note, final retrieval performance also depends on Google's proprietary ranking, which is outside the scope of our system capacity.

We have also designed and implemented a comparatively greedy baseline approach; in this case, we consider the smallest subset of n-grams that covers the entire question as much as possible without overlapping each other and keeps as many of the longest as possible and feed into the search engine.

We compare our approaches against the greedy approach, baseline strategy, CBQT, and its LDA-based expansion approach CBQE [1], and report the respective performance in Table IV, enabling a direct evaluation of our improvements in query interpretation and semantic matching.

C. Experimental Results

We evaluate the effectiveness of our proposed approach, SQUIRE, and its enhanced variant SQUIRE+LLM, using the TREC question dataset. Table IV presents Mean Reciprocal Rank (MRR) comparisons against several baseline approaches and comparative approaches, Context-Based Query Translation (CBQT) and its extension with LDA-based Context-Based Query Expansion (CBQE) using Google search results.

As shown, SQUIRE+LLM outperforms all baselines and comparative approaches across nearly all question categories. It achieves the highest MRR in 9 out of 10 categories, including significant gains in complex questions such as "How", "Why" and "Who". For "How", MRR increases from 0.27 (baseline in Table IV) to 0.54, demonstrating the ability to capture complex queries. Notably, for "Why" questions, it demonstrates its robustness by representing a 400% relative gain. Across all question types, SQUIRE+LLM achieves an average MRR of 0.51, compared to 0.46 for CBQT+CBQE (LDA), 0.45 for CBQT, and 0.38 for the baselines. Figure 2 provides a visual summary of the comparative performance across question categories.

Moreover, our approach achieves this performance without relying on any explicit query expansion techniques, highlighting the strength of ESA-based semantic interpretation without the need for query expansion. Overall, these results demonstrate SQUIRE+LLM's ability to generate semantically coherent and retrieval-effective query representations where surface-level keyword matching falls short.

V. Conclusion

In this work, we proposed **SQUIRE+LLM**, an unsupervised, concept-based framework for interpreting and ranking entity-term combinations resulting from natural language

queries. Our framework allows us to produce semantically coherent interpretations of entity combinations without query expansion, supervised learning, or black-box embedding.

The results show that SQUIRE+LLM outperforms both the baseline approaches and the context-based question translation model (CBQT) and its expansion variant CBQE, especially for semantically rich query types, like "Who", "What", "Name the," and "How" questions. The useful, interpretable architecture and open domain knowledge from Wikipedia make SQUIRE+LLM scalable and domain independent.

One of our key contributions is to examine semantically rich query understanding models that do not rely solely upon lexical overlap or dense embeddings like other models discussed in the literature. It demonstrates query understanding by operating from a core concept vector sourced from Wikipedia without predefined models of training data or normalized concepts. By grounding entity disambiguation and rankings directly on Wikipedia's knowledge and only using Wikipedia as an unsupervised search space, we can improve query interpretations without needing labeled or supervised training data or explicitly defined question-type keywords. Remarkably, our approach outperforms across nearly all question categories, making it a versatile and scalable approach for a wide range of user queries, well beyond just question answering.

However, in specialized or emerging domains, relying solely on Wikipedia for semantic grounding may limit performance because Wikipedia's coverage can become outdated. To address this, future work will look into adding domain-specific knowledge bases to support Wikipedia and expand the framework's usefulness. Furthermore, we plan to expand the role of large language models beyond query normalization by incorporating them as re-ranking components. This would allow us to combine the interpretability of ESA-based semantic scoring with the contextual reasoning capabilities of LLMs—an integration that is out of scope of our current work in order to preserve the interpretability and transparency of the system. Although our current system is entirely unsupervised, it is worth exploring how integrating pretrained embeddings and LLM-based reasoning can further enhance semantic understanding—without compromising transparency.

REFERENCES

[1] H. Wang, S. M. M. H. Chowdhury, and W. Jin, "Wikipedia empowered natural language interface for web search," in *Web Information Systems Engineering – WISE 2024: 25th International Conference, Doha, Qatar, December 2–5, 2024, Proceedings, Part I*. Berlin, Heidelberg: Springer-Verlag, 2024, p. 14–25. [Online]. Available: https://doi.org/10.1007/978-981-96-0579-8_2

[2] V. Gupta and A. Dixit, "Recent query reformulation approaches for information retrieval system-a survey," *Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science)*, vol. 16, no. 1, pp. 94–107, 2023.

[3] N. Kolitsas, O. Ganea, and T. Hofmann, "End-to-end neural entity linking," in *CoNLL*, 2018.

[4] G. Pons, B. Bilalli, and A. Queralt, "Knowledge graphs for enhancing large language models in entity disambiguation," in *International Semantic Web Conference*. Springer, 2024, pp. 162–179.

[5] D. Liu and S. Nag, "Query brand entity linking in e-commerce search," *arXiv preprint arXiv:2502.01555*, 2025.

[6] F. Mitsuji, S. Chakraborty, T. Morita, S. Egami, T. Ugai, and K. Fukuda, "Entity linking for wikidata using large language models and wikipedia links," in *2024 Twelfth International Symposium on Computing and Networking Workshops (CANDARW)*. IEEE, 2024, pp. 144–149.

[7] Y. Ding, A. Poudel, Q. Zeng, T. Weninger, B. Veeramani, and S. Bhattacharya, "Entgpt: Linking generative large language models with knowledge bases," *arXiv preprint arXiv:2402.06738*, 2024.

[8] L. e. a. Logeswaran, "Zero-shot entity linking by reading entity descriptions," in *ACL*, 2019.

[9] L. e. a. Wu, "Scalable zero-shot entity linking with dense entity retrieval," in *EMNLP*, 2020.

[10] F. Yan, P. Yu, and X. Chen, "Ltner: Large language model tagging for named entity recognition with contextualized entity marking," in *International Conference on Pattern Recognition*. Springer, 2025, pp. 399–411.

[11] E. Whittaker and I. Kitagishi, "Large language models for simultaneous named entity extraction and spelling correction," *arXiv preprint arXiv:2403.00528*, 2024.

[12] E. Gabrilovich and S. Markovitch, "Computing semantic relatedness using wikipedia-based explicit semantic analysis," in *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, ser. IJCAI'07. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007, p. 1606–1611.

[13] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[14] Z. Ye, Y. J. Kumar, G. O. Sing, F. Song, and J. Wang, "A comprehensive survey of graph neural networks for knowledge graphs," *IEEE Access*, vol. 10, pp. 75 729–75 741, 2022.

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, 2019, pp. 4171–4186.

[16] T. M. Tashu, E.-R. Kontos, M. Sabatelli, and M. Valdenegro-Toro, "Cross-lingual document recommendations with transformer-based representations: Evaluating multilingual models and mapping techniques," in *Proceedings of the Second Workshop on Scaling Up Multilingual & Multi-Cultural Evaluation*, 2025, pp. 39–47.

[17] L. Mazzola, P. Siegfried, A. Waldis, F. Stalder, A. Denzler, and M. Kaufmann, "A domain specific esa method for semantic text matching," in *Intelligent Systems: Theory, Research and Innovation in Applications*. Springer, 2020, pp. 25–49.

[18] D. Bogdanova and M. Yazdani, "Sesa: Supervised explicit semantic analysis," *arXiv preprint arXiv:1708.03246*, 2017.

[19] S. Kazi, S. Khoja, and A. Daud, "Bridging the gap: A survey of document retrieval techniques for high-resource and low-resource languages," *Computer Science Review*, vol. 57, p. 100756, 2025.

[20] J. Liu and B. Mozafari, "Query rewriting via large language models," *arXiv preprint arXiv:2403.09060*, 2024.

[21] X. Ma, Y. Gong, P. He, H. Zhao, and N. Duan, "Query rewriting in retrieval-augmented large language models," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 5303–5315.

[22] M. Li, M. Li, K. Xiong, and J. Lin, "Multi-task dense retrieval via model uncertainty fusion for open-domain question answering," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, Eds. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 274–287. [Online]. Available: https://aclanthology.org/2021.findings-emnlp.26/

[23] A. Anand, V. Setty, A. Anand *et al.*, "Context aware query rewriting for text rankers using llm," *arXiv preprint arXiv:2308.16753*, 2023.

[24] S. Uchida, T. Yamamoto, M. P. Kato, H. Ohshima, and K. Tanaka, "Entity ranking by learning and inferring pairwise preferences from user reviews," in *Information Retrieval Technology: 13th Asia Information Retrieval Societies Conference, AIRS 2017, Jeju Island, South Korea, November 22-24, 2017, Proceedings 13*. Springer, 2017, pp. 141–153.

[25] S. Gu, X. Luo, H. Wang, J. Huang, and S. Huang, "Abstract concept instantiation with context relevance measurement," *Journal of Web Engineering*, vol. 19, no. 5-6, pp. 575–602, 2020.

[26] X. Jiang, "Understanding user preference–comparison between linear and directional top-k query results," *arXiv preprint arXiv:2501.12337*, 2025.