# Hellenic Mediterranean University

**School of Engineering**

**Department of Electrical and Computer Engineering**

**Thesis**
**Master of Science**

***Enhancing KBQA with Semantic Embeddings and LLMs***

**Giannakos Iakovos**
**MTP312**

**Supervisors: Dr. Papadakis Nikolaos**

## Abstract

Knowledge-Based Question Answering (KBQA) is one of the key tasks in Natural Language Processing (NLP), as it supports retrieval based on factual information using structured data sources. This capability broadens access to information by allowing individuals, regardless of their technical expertise, allowing them to query structured information without the need for experience in specialized query languages such as SPARQL and SQL. KBQA systems are increasingly important in explainability and accuracy-demanding applications such as intelligent assistants, academic search engines, enterprise systems, and health-care information retrieval.

However, conventional KBQA solutions, especially those dependent on semantic parsing or retrieval-oriented structures solely, are confronted by several well-documented challenges. These are lexical mismatch between questions and knowledge base entries, natural language ambiguity, processing incomplete or loosely organized questions, as well as maintaining scalability efficiency with growing knowledge base size. Besides, such systems are likely to be unable to cope with the diverse, informal ways users pose their questions as well as not be able to link related facts across multiple triples.

In order to overcome such limitations, this project introduces a hybrid KBQA system based on semantic embeddings coupled with the reasoning function of state-of-the-art Large Language Models (LLMs). The system begins with the use of Sentence-BERT (SBERT) in generating dense semantic embeddings of the users' input queries. These are matched with pre-processed Freebase knowledge base via approximate nearest neighbor search using Annoy indexing, making it possible for the system to fetch semantically similar candidate triples in an efficient manner. This embedding-based retrieval significantly improves the system's ability in identifying the underlying intent of paraphrased and diverse queries.

To further enhance the quality of the returned information, the top-k candidate triples are re-ranked based on the use of a Cross-Encoder model. The Cross-Encoder jointly encodes query-triple pairs in order to capture fine-grained contextual relationships and outputs relevance scores for selecting top-ranking ones. Finally, the query along with top-ranked triples is passed through a Large Language Model, i.e., ChatGPT-4o or DeepSeek-v3, for response generation in natural language. These LLMs include an additional generative reasoning layer, enabling the system to address ambiguity, perform light inferences, as well as make fluent contextual responses according to user intent.

The system was tested on the WebQuestions dataset, a standard benchmark for KBQA tasks. Evaluation was performed based on exact match and F1 measures for system answer comparison against gold-standard answers. Experimental results show that ChatGPT-4o achieved the best F1 score, performing better than DeepSeek-v3 and ChatGPT-4o-mini. This validates that the integration of SBERT-based retrieval, neural re-ranking, and LLM reasoning results in substantial gains in answer accuracy and language naturalness, particularly for non-canonical and diverse queries.

This project proposes an extensible and scalable KBQA system that can be equipped with structured knowledge retrieval combined with deep contextual understanding. Directions for the future include scaling it up towards multilingual and cross-lingual KBQA, as well as multi-hop reasoning for hard question answering that requires bridging multiple knowledge

entities. Future directions especially involve adding real-time feedback from users, evaluation on domain specialized KBs, and system robustness testing on new long-context LLMs.

# Περίληψη

Στο πεδίο της Επεξεργασίας Φυσικής Γλώσσας (Natural Language Processing - NLP), το πεδίο Knowledge-Based Question Answering (KBQA) αναγνωρίζεται ως θεμελιώδες, καθώς επιτρέπει την ανάκτηση έγκυρης πληροφορίας από δομημένες πηγές δεδομένων. Αυτή η δυνατότητα παρέχει έναν απλοποιημένο τρόπο πρόσβασης σε σύνθετη δομημένη πληροφορία, χωρίς να απαιτείται εξειδικευμένη γνώση σε επίσημες γλώσσες ερωτημάτων όπως οι SPARQL και SQL. Τα KBQA συστήματα καθίστανται ολοένα και πιο σημαντικά σε εφαρμογές που απαιτούν ερμηνευσιμότητα και ακρίβεια, όπως οι ευφυείς βοηθοί, οι ακαδημαϊκές μηχανές αναζήτησης, τα επιχειρησιακά συστήματα και τα πληροφοριακά συστήματα στον τομέα της υγείας.

Ωστόσο, οι παραδοσιακές λύσεις KBQA, ιδίως αυτές που βασίζονται αποκλειστικά σε semantic parsing ή σε retrieval-based αρχιτεκτονικές, αντιμετωπίζουν χρόνια προβλήματα. Ανάμεσα σε αυτά περιλαμβάνονται η λεκτική αναντιστοιχία μεταξύ των ερωτήσεων και των εγγραφών της knowledge base, η ασάφεια στη φυσική γλώσσα, η δυσκολία επεξεργασίας ημιτελών ή ασαφώς διατυπωμένων ερωτήσεων, καθώς και η πρόκληση σε συνθήκες μεγάλου όγκου δεδομένων. Επιπλέον, τέτοια συστήματα συνήθως αδυνατούν να διαχειριστούν τον ποικίλο και μη τυποποιημένο τρόπο με τον οποίο οι χρήστες διατυπώνουν ερωτήσεις, ενώ επίσης αποτυγχάνουν να συνδέσουν σχετικές πληροφορίες διασκορπισμένες σε πολλαπλά triples.

Για την αντιμετώπιση αυτών των περιορισμών, το παρόν έργο προτείνει ένα υβριδικό σύστημα KBQA, το οποίο βασίζεται σε semantic embeddings σε συνδυασμό με τις δυνατότητες reasoning που παρέχουν τα σύγχρονα Large Language Models (LLMs). Το σύστημα ξεκινά με τη χρήση του Sentence-BERT (SBERT) για την παραγωγή πυκνών εννοιολογικών embeddings των ερωτήσεων των χρηστών. Οι εν λόγω αναπαραστάσεις συγκρίνονται με ένα προ-επεξεργασμένο knowledge base από το Freebase, μέσω approximate nearest neighbor αναζήτησης, αξιοποιώντας την τεχνική Annoy indexing. Αυτό επιτρέπει την αποδοτική ανάκτηση triples με υψηλή σημασιολογική συνάφεια.

Για τη βελτίωση της ποιότητας των υποψηφίων απαντήσεων, εφαρμόζεται ένα Cross-Encoder μοντέλο που επανεκτιμά τα κορυφαία k-triples. Το Cross-Encoder επεξεργάζεται συνδυαστικά το query και κάθε triple για να καταγράψει τις λεπτές σημασιολογικές συσχετίσεις, αποδίδοντας σχετικές βαθμολογίες και επιλέγοντας τις πιο κατάλληλες εγγραφές. Τέλος, η αρχική ερώτηση και τα επαναβαθμολογημένα triples δίνονται ως είσοδος σε ένα Large Language Model όπως το ChatGPT-4o ή το DeepSeek-v3, για παραγωγή απάντησης σε φυσική γλώσσα. Τα LLMs προσφέρουν ένα επιπλέον επίπεδο συλλογιστικής (reasoning), επιτρέποντας στο σύστημα να διαχειρίζεται ασάφειες, να εκτελεί ελαφρού τύπου inference και να παρέχει συνεκτικές και απαντήσεις που βασίζονται στα συμφραζόμενα.

Το προτεινόμενο σύστημα αξιολογήθηκε στο σύνολο δεδομένων WebQuestions, που αποτελεί καθιερωμένο benchmark για KBQA έργα. Η αξιολόγηση βασίστηκε σε ακριβείς συνδυασμούς (exact match) και F1 μετρικές, συγκρίνοντας τις απαντήσεις του συστήματος με τις gold-standard απαντήσεις. Τα πειραματικά αποτελέσματα έδειξαν ότι το ChatGPT-4o πέτυχε τον υψηλότερο F1 score, ξεπερνώντας το DeepSeek-v3 και το ChatGPT-4o-mini. Αυτό επιβεβαιώνει ότι η ενσωμάτωση SBERT retrieval, νευρωνικού re-ranking και reasoning μέσω

LLM οδηγεί σε σημαντικές βελτιώσεις στην ακρίβεια και φυσικότητα των απαντήσεων, ειδικά σε μη κανονικές και ποικιλόμορφες ερωτήσεις.

Το έργο αυτό δείχνει ένα επεκτάσιμο και αποδοτικό KBQA σύστημα που συνδυάζει την ανάκτηση από δομημένη γνώση με βαθιά συμφραζόμενη κατανόηση. Μελλοντικές κατευθύνσεις περιλαμβάνουν την επέκτασή του σε πολυγλωσσικές και cross-lingual KBQA εφαρμογές, καθώς και την υλοποίηση συλλογισμού πολλαπλών βημάτων (multi-hop reasoning) για την απάντηση πολύπλοκων ερωτήσεων που απαιτούν σύνδεση πολλαπλών knowledge entities. Επιπλέον, προτείνεται η ενσωμάτωση ανατροφοδότησης σε πραγματικό χρόνο από τους χρήστες, η αξιολόγηση σε εξειδικευμένες βάσεις γνώσης και η δοκιμή της ανθεκτικότητας του συστήματος σε νέα LLMs με long-context δυνατότητες.

# Table of Contents

2 Related Research ..... **Error! Bookmark not defined.**
2.1 Embedding-Based Retrieval ..... **Error! Bookmark not defined.**
2.2 Neural Re-Ranking with Cross-Attention ..... **Error! Bookmark not defined.**
2.3 Generative and Pretrained Language Models ..... **Error! Bookmark not defined.**
2.4 Retrieval-Augmented Generation ..... **Error! Bookmark not defined.**
2.5 Gap and Motivation ..... **Error! Bookmark not defined.**
3 Background ..... 12
3.1 Introduction to KBQA (Knowledge-Based Question Answering) ..... 12
3.2 Knowledge Bases in KBQA ..... 12
3.3 Advancements in NLP and Deep Learning ..... 13
3.4 Semantic Representations for KBQA ..... 13
3.5 Leveraging Large Language Models (LLMs) ..... 16
3.6 Hybrid Approach in KBQA ..... 17
3.7 Challenges in KBQA ..... 18
4 Methodology ..... 19
4.1 Approach Overview ..... 19
4.2 Architecture of the System ..... 20
4.2.1 Knowledge Base Representation ..... 20
4.2.2 Retrieval Model (Embedding-Based Search) ..... 20
4.2.3 Neural Reranking Model (Cross-Encoder + LLM) ..... 21
4.2.4 Final Answer Selection ..... 22
4.3 Data Processing Pipeline ..... 22
4.4 Evaluation & Performance ..... 23
5 Experiment ..... 24
5.1 Problem definition ..... 24
5.2 Data ..... 24
5.3 Results ..... 25
6 Discussion ..... 27
7 Conclusions ..... 28
8 Appendix ..... 30
8.1 A SBERT-Based Semantic Evaluation ..... 30
8.2 Threshold Sensitivity and Optimal Performance Range ..... 33
9 Bibliography ..... 34

# Tables and Figures

Figure 2.1: Comparison of Competitive Approaches. ..... **Error! Bookmark not defined.**
Figure 3.1: SBERT architecture at inference, illustrating the process of computing similarity scores between sentence pairs using cosine similarity. This architecture efficiently encodes sentences in parallel and is utilized for semantic similarity tasks in KBQA. (Adapted from Reimers & Gurevych, 2019) [11]. ..... 14
Figure 3.2: A Cross-Encoder Architecture. (Adapted from Humeau, 2020) [12]. ..... 15
Figure 3.3: MMLU Benchmark Comparison of LLMs used in KBQA experiment. ..... 16
Figure 4.1: The architecture system for KBQA. The pipeline includes query embedding, processing Freebase triplets, retrieving with SBERT, re-ranking using a Cross-Encoder and LLM to obtain the final top-K entities. ..... 20
Table 5.1: Entity Predictor Results ..... 25
Table 5.2: LLM Models (Entity Predictor) Results ..... 25
Table 5.3: Each pipeline step process time for 289483 triples and 2032 Questions ..... 26
Figure 8.1: Average Precision, Recall, and F1 Score vs. SBERT Threshold for ChatGPT-4o LLM. ..... 31
Figure 8.2: Average Precision, Recall, and F1 Score vs. SBERT Threshold for DeepSeek-V3 LLM. ..... 31
Figure 8.3: Average Precision, Recall, and F1 Score vs. SBERT Threshold for ChatGPT-4o-mini LLM. ..... 32
Table 8.4: Average F1 Score vs. SBERT Threshold per LLM. ..... 32

# 1 Introduction

Knowledge Base Question Answering (KBQA) has emerged as a crucial and rapidly progressing branch within the broader domain of Natural Language Processing (NLP). These systems are engineered to automatically comprehend and address queries expressed in natural language by harnessing structured data resources, such as Knowledge Graphs (e.g., Freebase and Wikidata). This method of information retrieval is highly valuable because it enables users to access large volumes of organized data directly, without needing to master complex query languages or understand the intricacies of data organization.

Nonetheless, the development of robust and precise KBQA systems is hindered by several intrinsic challenges. Foremost among these is the lexical gap, which denotes the disparity between the terminology used in user queries and the corresponding entities or relations present in the knowledge base. This gap may be attributed to differences in vocabulary, the use of synonyms, paraphrasing variations, or divergent linguistic expressions. Furthermore, the inherent ambiguities and complexities associated with natural language queries, when compared to the vast scale and intricate structure of knowledge graphs, exacerbate these challenges.

The main aim of this research is to develop and thoroughly evaluate a novel approach to knowledge base question answering, utilizing advanced deep learning models in conjunction with effective indexing strategies. In particular, it introduces an architecture that integrates refined sentence embedding methods (SBERT) with semantic matching models (Cross-Encoders) and augments them with a highly efficient approximate nearest neighbor search mechanism, known as Annoy indexing. This innovative combination of methodologies is intended to achieve performance that exceeds that of current state-of-the-art KBQA systems.

# 2 Related Research

Prior work on Knowledge Base Question Answering (KBQA) can be broadly grouped into four strands: embedding-based retrieval, neural re-ranking with cross-attention, generative pretrained models and textual-evidence validation. Each trades off between scale, precision and fluency. We review these in turn and then highlight the remaining gap that motivates our three-stage hybrid architecture.

## 2.1 Embedding-Based Retrieval

Early IR-based KBQA methods learn joint vector embeddings of natural-language questions and KB triples so that simple nearest-neighbor lookup retrieves answers in milliseconds. Yang et al. (2014) [1] project questions and Freebase predicates into a shared low-dimensional space, greatly improving lexical coverage over handcrafted lexicons but still failing on unseen question phrases. Bordes et al. (2014) [2] embed question tokens alongside entities and relations and score each triple by cosine similarity. This one-pass retrieval hits sub-10 ms on million-scale indices but cannot refine its top-k candidates when subtle context matters. These works show that dense similarity search can scale to millions of facts yet expose a clear weakness in semantic precision.

## 2.2 Neural Re-Ranking with Cross-Attention

To correct retrieval errors, subsequent work adds a neural re-ranking stage over the top-k candidates. Dong et al. (2015) [3] apply a multi-column CNN with separate channels for the answer path, answer context and expected answer type by learning which question fragments matter most, they markedly boost top-1 accuracy over pure embedding baselines. In a similar gain in accuracy Xu et al. (2016) [4] retrieved support passages from Wikipedia for each candidate triple, and then filtered out some superficially similar but inaccurate answers based on both Wikipedia relation scores and textual matching scores. Hao et al. (2017) [5] go in a new direction. They proposed a two-way cross-attention transformer based system that jointly attends over question tokens and KB features to re-score each candidate triple, achieving double digit improvements on multiple benchmarks. Chen et al. (2019) [6] augment this with an explicit memory for better interpretability but still stop at selecting a single triple rather than generating full-sentence answers. These pipelines correct many retrieval mistakes yet fall short of producing fluent, multi-fact responses.

## 2.3 Generative Pretrained Models

Japa and Rekabdar (2020) [7] fine-tune BERT with two-way attention to compare questions and candidate triples, improving ranking metrics over earlier baselines but still only selecting triples. While this brings KBQA into the realm of pretrained language models, it does not yet yield full, fluently generated answers.

## 2.4 Textual-Evidence Validation

Other KBQA pipelines supplement the structured retrieval with verification against unstructured text for candidate triples. Xu et al. (2016) [4] return supporting Wikipedia passages for each triple and merge KB-relation scores with text-matching-based scores to rule out shallowly alike but incorrect answers. While this enhances precision, these methods fail at triple selection rather than generating full, multi-fact answers.

## 2.5 Gap and Motivation

Existing pipelines either:

- scale to millions of facts but lack deep contextual precision (Section 2.1),
- apply powerful cross-attention models but handle only a few hundred candidates and stop at triple selection (Section 2.2), or
- generate fluent answers yet risk hallucination when untethered from structured retrieval (Sections 2.3–2.4).

No current system unifies all three critical functions for robust KBQA:

1. Millisecond-scale retrieval over millions of structured facts
2. Deep contextual re-ranking via transformer-based cross attention
3. Fluent multi-fact answer synthesis with explicit provenance linking back to KB triples

This gap motivates our three-stage hybrid architecture, which combines SBERT-Annoy retrieval, a transformer cross-encoder for precise candidate re-scoring and a generation module that produces grounded, human-quality answers end to end.

| Method | Description | Limitation |
|---|---|---|
| Embedding-Based Retrieval (Yang et al., 2014 [1]; Bordes et al., 2014 [2]). | Jointly embed questions and KB triples for one-pass similarity search. | Fails on unseen phrases, no refinement of top k candidates. |
| Neural Re-Ranking and Cross-Attention (Dong et al., 2015 [3]; Xu et al., 2016 [4]; Hao et al., 2017 [5]; Chen et al., 2019 [6]) | Use CNN or transformer cross-encoders over top k triples for precise re-ranking. | Handles only hundreds of candidates, stops at triple selection. |
| Generative Pretrained LMs (Japa and Rekabdar, 2020 [7]) | Fine-tune large language models to rank triples or generate fluent answers. | Can hallucinate facts, lacks structured grounding. |
| Textual-Evidence Validation (Xu et al. 2016 [4]) | Retrieve Wikipedia passages to validate and filter candidate triples. | Still stops at triple selection, no answer synthesis |

| | | |
|---|---|---|
| Ours: Three-Stage Hybrid | SBERT and Annoy retrieval, transformer cross-encoder re-ranking, grounded answer generation. | More complex pipeline. Can hallucinate facts. Slightly higher end-to-end latency. |

*Figure 2.1: Comparison of Competitive Approaches.*

# 3 Background

## 3.1 Introduction to KBQA (Knowledge-Based Question Answering)

Knowledge-Based Question Answering (KBQA) is a branch of natural language processing concerned with answering questions using structured data from Knowledge Bases (KB). Knowledge Base (KB) such as Freebase holds huge amounts of factual information that is organized in a structured format for efficient retrieval of answers to natural language queries. Freebase has been widely used in KBQA research for its rich repository of entities and relationships, forming a crucial backbone for natural language understanding tasks. [6]

Knowledge-based Question Answering (KBQA) is a crucial sub-task for various applications, which a KBQA system is able to answer user questions told to automated reasoning (e.g. in an automated scientific research application). Such systems typically parse the input question in natural language and create a structured query that can be run against a KB to obtain a correct answer.

Nevertheless, there are some challenges for KBQA. A significant challenge is formulating an understanding of complicated questions, which require multi-use processing and interlinking reasoning, among others. Moreover, inevitably mapping the natural language query to the structured knowledge representation appropriately is often difficult as the query could be 'fuzzy' and/or 'ambiguous'. It is important to ensure that relations and entities from the KB are correctly matched by the system, as this will produce accurate answers.

KBQA algorithms, however the structure of knowledge graphs means that a question may not have a direct answer even if the relevant information is there, and poorly formulated questions can refer to clusters of entities, making it almost impossible to correlate the data needed. For example, models such as BERT (Devlin et al., 2019) [8], have achieved new records in natural language understanding, which facilitated the utilization of more comprehensive knowledge to advance related works such as KBQA systems. The recent past saw the advent of approaches that incorporated structured data with contextual reasoning to answer questions more accurately and thoroughly.

## 3.2 Knowledge Bases in KBQA

Knowledge Bases (KB) are structured information storehouses for large factual datasets. They typically include entities, relationships, and attributes which can queried to quickly get to the most relevant information. Due to the structured nature of KBs, this is a very appealing task for question answering, where queries expressed in natural language need to be mapped to specific, structured answers.

A popular example of Knowledge Base used in KBQA research is Freebase. Freebase is a graph database that was collaboratively constructed and stores millions of entities and their

relationships that has been built in a certain way to enable it query-able (Bollacker et al., 2008) [9]. Freebase has been extensively used as a trustworthy KB to answer natural language questions in KBQA, offering a large-scale, structured database. This facilitates the retrieval of factual information and allowing queries to be correctly linked with the respective entities.

Structured data like that contained in knowledge bases (such as Freebase) is shown to be required for efficient question answering due to the observations that systems could map the question to the tables and found answers could be presented directly to the user. In contrast, in unstructured text-based approaches, the system has to implicitly discern relationships. Utilizing the explicit relations in KBs would allow KBQA systems for more accurate and quicker query resolution. Thus, the most critical component of successful KBQA systems is the utilization of structured data from KBs.

## 3.3 Advancements in NLP and Deep Learning

Deep learning models are the key drivers of recent advancement in NLP. These breakthroughs have transformed many NLP applications, such as question answering, information retrieval, and language comprehension. The development of transformer-based models has been one of the prominent advances in this area.

In NLP, transformer models like BERT (Bidirectional Encoder Representations from Transformers) have established new benchmarks allowing for deep contextual understanding of text. It has been particularly successful for tasks requiring some level of directedness or about fairly recent, contextualized representations of words. Sentence-BERT (SBERT) takes BERT a step further when it comes to semantic textual similarity by providing it with a siamese network structure and directly optimizing it towards that task (Reimers & Gurevych, 2019) [10].

Such advances have direct repercussions on KBQA systems, where generating high-quality semantic embeddings enhances the matching of questions to relevant answers (triples). However, introducing transformer models in KBQA pipeline, enables to more appropriately interpreted natural language query and contributes to providing much more accurate answers from structured data. As a consequence, deep learning technologies, especially transformer-based methods have played a significant role in the advancement of KBQA systems.

## 3.4 Semantic Representations for KBQA

For Knowledge-Based Question Answering (KBQA), semantic relationships in Knowledge Base (KB) play an important role in determining an accurate answer. The real challenge is understanding natural language and connecting it to structured data complete with embeddings that preserve semantic similarity across entities and attributes. For embedding generation and

similarity scoring, you feed the text into models such as Sentence-BERT (SBERT) and Cross-Encoder.

A modification of BERT architecture for sentence-level semantic similarity tasks is SBERT `Sentence-BERT` (Reimers & Gurevych, 2019) [10]. Although BERT is very powerful for contextual word representations, it is less effective for sentence embeddings. SBERT improves on this and uses a siamese network structure and encodes sentences in parallel to compute the result in much lesser time. The dual encoder setup where we encode each sentence separately and compare with cosine similarity allows SBERT to perform well for matching natural language questions with KB entries.

SBERT is trained on semantically similar sentence pairs. It is suitable for comparing meanings of sentences, which is a major necessity of KBQA in obtaining relevant answers. SBERT provides reliable handling of large numbers of candidate answers due to its high level of encoding efficiency when it comes to semantic information. In addition, the model comes out excellent in real world applications where the form of the queries and structuring of the questions differ significantly.

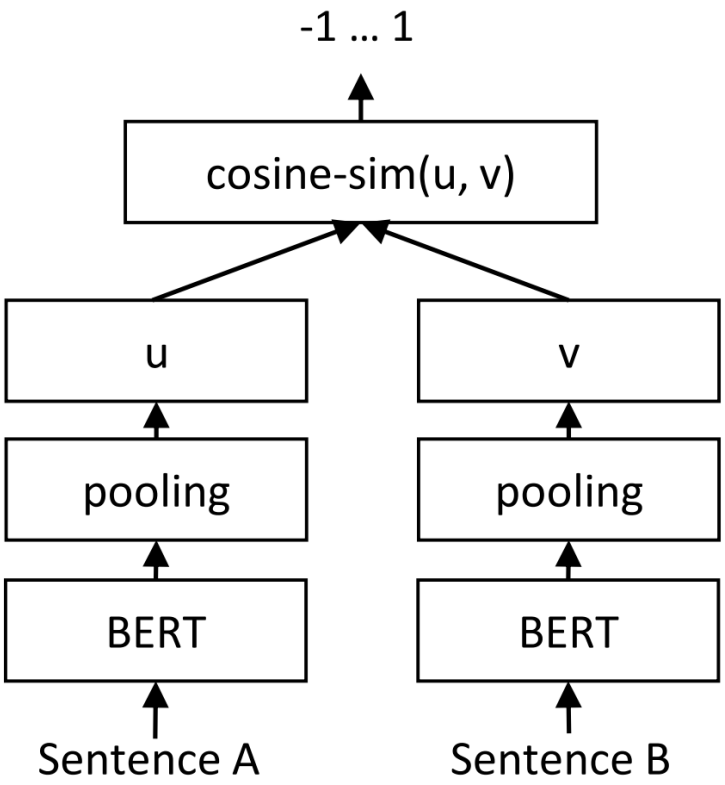


*Figure 3.1: SBERT architecture at inference, illustrating the process of computing similarity scores between sentence pairs using cosine similarity. This architecture efficiently encodes sentences in parallel and is utilized for semantic similarity tasks in KBQA. (Adapted from Reimers & Gurevych, 2019) [10].*

Cross-Encoder, by contrast, encodes input pairs together (Humeau et al., 2019) [11]. This shared encoding step captures complex relationships between the texts and provides more

accurate similarity scores. While the process of generating the embeddings for the text might make it appear similar to SBERT, the Cross-Encoder, on the other hand, looks at the connectedness of paired texts to capture the nuance of the relationship better and understand the context. This task is important when a query must be matched closely with its answer. For multi-hop logic or detailed reasoning questions which require long caretaking of the prior context, the Cross-Encoder's one-end approach has an edge as it preserves subtle contextual cues during the retrieval of the relevant documents.

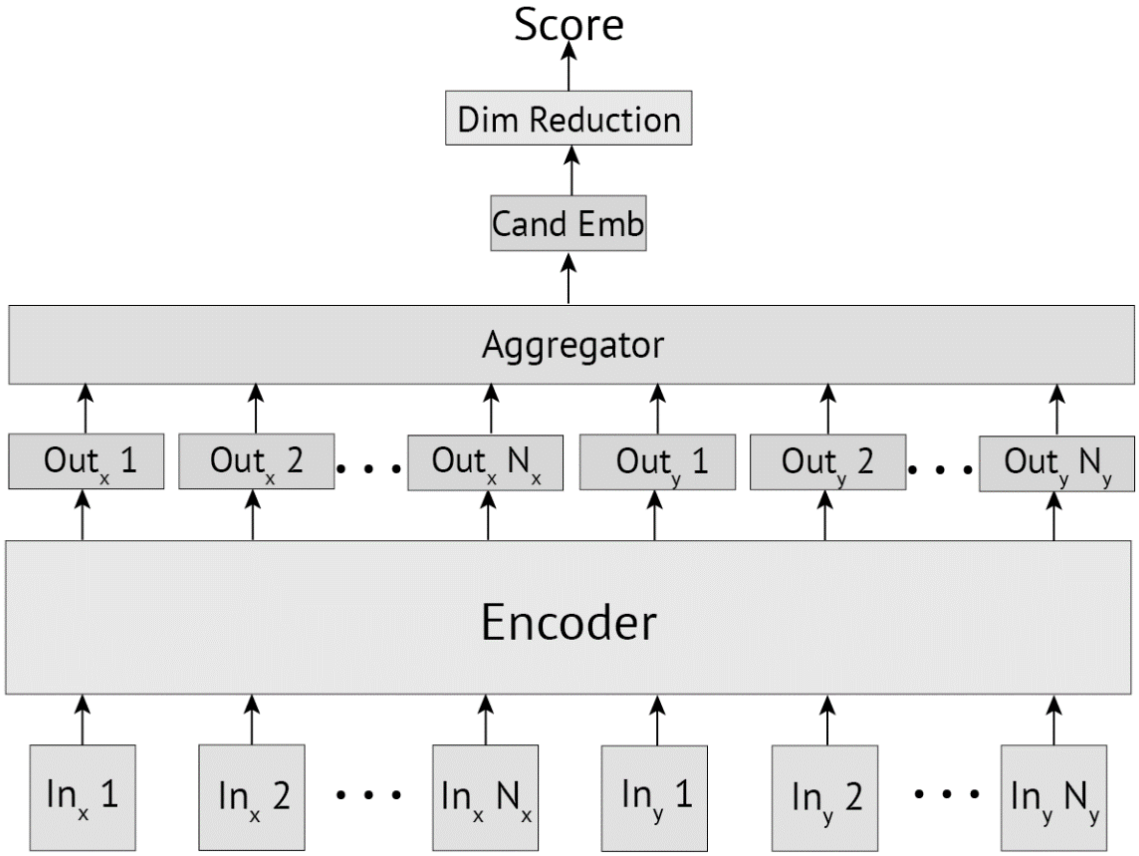


*Figure 3.2: A Cross-Encoder Architecture. (Adapted from Humeau, 2020) [11].*

A major benefit of SBERT and Cross-Encoder fusion in KBQA is their complementary properties. The strong point of SBERT is the fast embedding generation and the quick filtering of candidates in a large corpus. Once the most promising candidates are selected, the Cross-Encoder can refine the similarity calculation so that the final answer is not just relevant but also contextually appropriate. Utilizing such stepwise filtering and refinement results in much higher quality and response speeds, especially for real-time or user-interactive systems for question answering.

These KBQA approaches combine the efficiency of SBERT for obtaining embedding of queries with the accuracy of Cross-Encoder for saturated queries. SBERT allows for much faster embedding generation and filtering of initial candidates, and the Cross-Encoder filters these candidates further, performing detailed similarity scoring. It incorporates speed with

accuracy, making it fit for real-time and complex query resolution. The powerful combination of these two neural network models produces a KBQA system that is able to answer questions from more than one possible schema.

## 3.5 Leveraging Large Language Models (LLMs)

This is especially useful in Knowledge Based Question Answering (KBQA), where large Language Models (LLMs) such as ChatGPT-4o and DeepSeek-v3 have gained popularity, demonstrating enhanced performance compared to rule-based KBQA methods. With their incredible fluency, these models' capabilities of natural language comprehension and generation allow them to respond directly to user questions across a diverse range of topics.

ChatGPT-4o, released in 2024, is a fully multimodal model that combines text, vision, and audio processing in a single architecture [12]. It has a 128k token window for long-context interactions and a new 200k tokenizer for improved efficiency in multilingual and complex input. Such features are particularly helpful for KBQA problems in which a question depends on previous information mentioned earlier in a dialogue or covers many sub-topics. GPT-4o can keep track of long conversations and understand subtle changes in context and it's a major improvement over its predecessors.

ChatGPT-4o has been evaluated on multiple benchmarks. It scored 88.7% in MMLU and 90.2% in HumanEval, outperforming previous versions, such as GPT-4 and GPT-3. 5 in both reasoning accuracy and speed. [12],[13] These scores indicate its strength in both factual recall and logical reasoning, the two skills that are the undercurrent of high-quality KBQA.

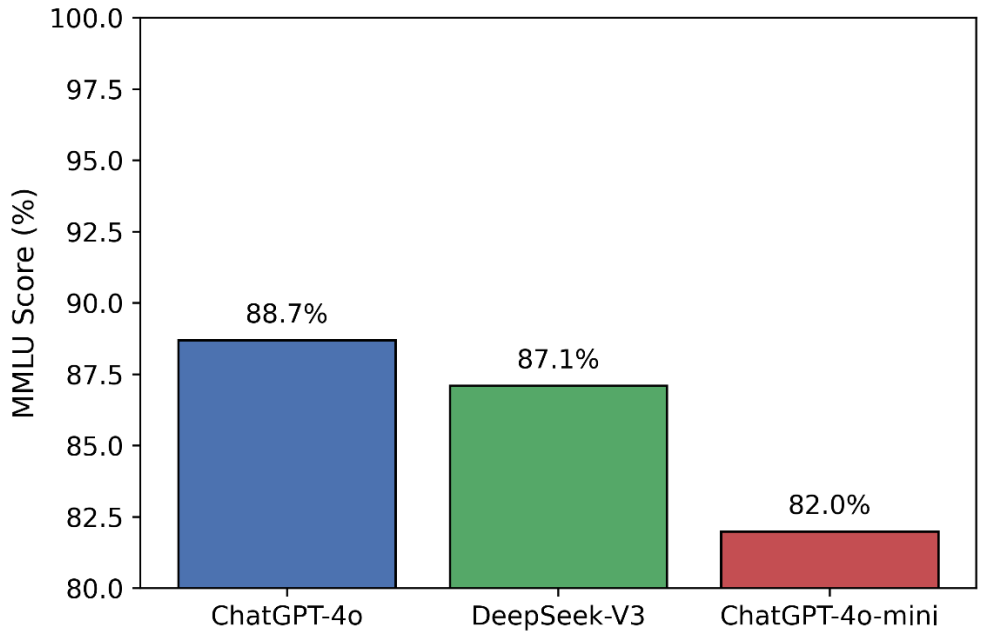


*Figure 3.3: MMLU Benchmark Comparison of LLMs used in KBQA experiment.*

Alternative models like DeepSeek-v3, on the other hand, target precision reasoning in specific domains, such as math and programming [14]. It works well in those areas but its architecture will not be as generalized as GPT-4o for generalized, open-domain question answering. This is why when building a KBQA system that needs to handle more unpredictable and natural user language, GPT-4o is a better choice.

In general, LLMs like GPT-4o come equipped with the versatility and contextual awareness needed to improve traditional KBQA systems. Since they have no issue understanding the subtleties of language and following lengthy, developing conversations, they can handle off highly complex and diverse user queries with improved accuracy than their predecessors. The condition of not being restricted by the fixed schemas of structured KBs allows these models to provide responses regardless of whether a direct answer is notionally stored. Additionally, their coupling with retrieval mechanisms allows them to draw on supplementary evidence in real time, which is especially useful in fields where information is constantly changing or is far too vast to be compiled manually into a knowledge base (KB). With each iteration of LLMs, their utility in KBQA is bound to increase, enabling systems that are not just going to be more accurate, but also more dynamic and user centric.

## 3.6 Hybrid Approach in KBQA

Current KBQA systems employ a hybrid approach that combines structured reasoning with contextual understanding, achieving both precision as well as flexibility in answer generation. The reasoning is structured and based on knowledge bases (KB) such as Freebase, enabling systems to provide factually consistent and accurate results from an entity-relation graph. Nevertheless, this approach might fail when users ask questions in the natural, conversational, less formal style or when the information is either missing or not available in the KB.

To overcome these issues a hybrid approach has been proposed which combines symbolic and neural modules. For this, our system uses semantic embedding techniques (SBERT), similarity-based re-ranking (Cross-Encoder) and generative contextual reasoning using LLMs like ChatGPT-4o and DeepSeek-v3. Such a layered methodology aims to leverage the strengths of both paradigms.

For both user query and KB triples, SBERT is used to create dense sentence embeddings. These embeddings are subsequently indexed for fast retrieval with an approximate nearest neighbor search. After the candidate answers are retrieved, a Cross-Encoder is then utilized to re-rank them by jointly encoding the question with each candidate, allowing the model to achieve greater alignment and contextual matching. In the final stage, LLMs can serve the role of interpreting the query within the context of a larger set of documents, generating natural language explanations for results, or validating and adding to structured results.

The result is a synergistic pipeline where efficient, accurate entity-based retrieval is augmented with the versatility and depth of understanding of LLMs. This enables the system to accommodate a variety of query formulations, whether explicit, partial or conversational while still being able to ground its answers in structured knowledge. The fusion of these elements enables the hybrid approach to effectively address KBQA challenges, providing a harmonious solution that balances scalability, accuracy, and contextual depth.

## 3.7 Challenges in KBQA

There are several ongoing challenges in Knowledge Based Question Answering (KBQA) systems that undermine their scalability as well as precision. This problem is aggravated by the high dimensionality of semantic embeddings generated from queries and entries in the knowledge base. These dense vector representations, typically generated with transformer-based encoders (e.g. SBERT or BERT), are extremely resource-constrained to create and search at scale efficiently [10]. This can be particularly challenging with large datasets, where hundreds of millions of triples, must be processed in rapid succession.

Data, of course, is another major barrier to using AI. Known as large-scale KBs like Freebase or Wikidata they contain huge amounts of entities and relations, which need advanced methods for indexing and retrieval. The difficulty is creating systems with high recall and precision that respond within an acceptable latency [9]. Additionally, in real-life scenarios user queries are usually incomplete, ambiguous, or in informal language which complicates retrieval from a formal KB.

Another major problem is how to properly encode knowledge into neural architectures. While symbolic logic functions by determining a relationship based on the ordering of data, more accurately known as comparison, neural models interpret meaning based on the contextual pattern of the data, also with long-tail or multi-hop querying the response could be misrepresented. Models such as ChatGPT-4o and DeepSeek-v3 make this attempt by integrating long context windows, increased parameter magnitudes, and retrieval-augmented generation principles to boost factual consistency and reasoning depth [12] [14]. Nonetheless, aligning retrieved facts and generated responses remains a serious problem.

These challenges inspire the hybrid methodology presented in this thesis that incorporates retrieval-based semantic embeddings, cross-encoder re-ranking, and generative LLM reasoning. This method aims to balance structured reasoning with context, all the while tackling scale, retrieval quality as well as representation fidelity.

# 4 Methodology

This chapter describes the methodology followed in our research which integrates Knowledge Bases (Freebase) with state-of-the-art neural models to improving question understanding and knowledge retrieval. The goal is efficiently retrieving relevant knowledge and provide accurate answers using a hybrid pipeline of vector search indexing, large-scale neural reranking and LLM.

## 4.1 Approach Overview

The KBQA system presented here consists of a three-stage pipeline for combining knowledge retrieval with language understanding in context. This involves several stages of candidate retrieval, followed by re-ranking and finally selection that utilize large language models.

For candidate retrieval, we start with the SBERT (Sentence-BERT) first encodes the user's natural language question into a dense vector space. Each triple in the Freebase knowledge base, which is made up of a subject, relation, and object, already comes with its own pre-computed vector embedding. These are stored in an Annoy-based nearest neighbor index [15], which facilitates similarity search. Given a query vector, the system scores and retrieves the k-nearest triples from the knowledge graph according to cosine similarity, and generates a ranked list of candidates that are semantically similar to the user input.

Once the candidate triples have been retrieved, a second stage occurs: re-ranking. For this purpose, we use a Cross-Encoder model [11], which takes a single query and candidate triple and encodes it to evaluate how well they match semantically. Instead of separately calculating embeddings, the Cross-Encoder takes the query-triple pair as a single input and outputs a score of relevance representing how likely the triple answers the question. It then sorts the candidates by the scores to list the most relevant information on top.

Finally, in the last step, the system uses some large language models (LLM), such as DeepSeek-V3 or ChatGPT-4o to understand both the re-ranked triples and the user original question. The LLM does not just take the top-ranked triple, but considers the surrounding context of the retrieved knowledge and the wording of the query. It then uses that information to produce a coherent and contextually suited entities response consistent with the user's request. This improves not just fluency and completeness of response but also strengthens the system's efficacy against differently or loosely worded questions.

As a result, this hybrid approach of multi-tasking enables fast semantic retrieval, accurate re-ranking, and flexible generative reasoning in a single model, making the KBQA system solid and powerful.

## 4.2 Architecture of the System

The system is designed as a structured pipeline for Knowledge Base Question Answering (**KBQA**) combining retrieval, semantic ranking and neural re-ranking.

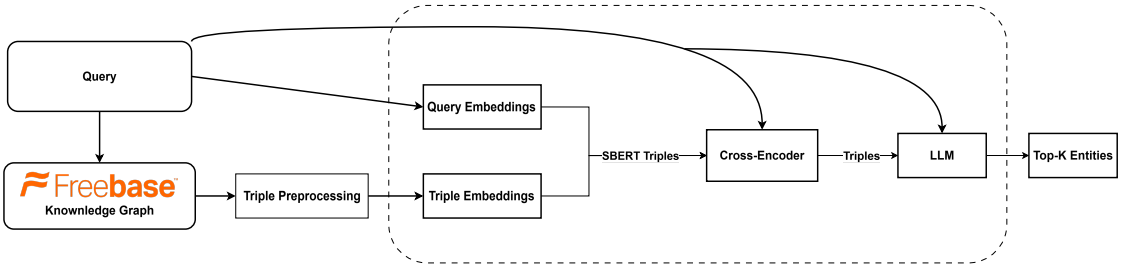


*Figure 4.1: The architecture system for KBQA. The pipeline includes query embedding, processing Freebase triplets, retrieving with SBERT, re-ranking using a Cross-Encoder and LLM to obtain the final top-K entities.*

The architecture consists of the following key components:

### 4.2.1 Knowledge Base Representation

The system uses Freebase dataset, a structured Knowledge Graph (KG) where is formatted as (subject, relation, object) triples. Before retrieval, the triplets undergo preprocessing, removing duplicates, and unimportant data, ensuring clean structured data is stored in TSV format.

### 4.2.2 Retrieval Model (Embedding-Based Search)

In order to retrieve relevant information from the knowledge base, an embedding-based retrieval model is utilized by the system, where user queries and entries (triples) of the knowledge base are both represented in the same semantic vector space. It makes semantic comparison of natural language input and structured data possible via dense representations, such that we can map onto other data more flexibly and accurately compared to the symbolic querying techniques used so far.

The query embedding stage is the first step of the retrieval process. In this, the user query in natural language is converted to a high-dimensional vector using SBERT (Sentence-BERT), a model fine-tuned specifically to capture semantic similarity at the sentence level. This embedding is a reduced representation of the asking intentions and context of a question, which can be computed and compared with the entries of the pre-computed knowledge base.

Simultaneously, each triple from the Freebase knowledge base (a subject, relation and object) is processed independently and embedded into this vector space. Triplet embeddings are usually created once during training and stored for fast access during inference. The main

objective is to have semantically similar queries and triples will be represented by nearby vectors.

For performing the actual retrieval, the cosine similarity between the embedded query vector and the stored triple embeddings is calculated. Present-day knowledge bases are so massive in scale that exact search either is not computationally feasible. The system uses an Approximate Nearest Neighbor (ANN) search using Annoy [15] as it is designed to work for the high-dimensional, large-scale indexing. Annoy is an approximate nearest neighbor search algorithm that retrieves the top-k most relevant knowledge triples based on their vector proximity to the query in an efficient manner.

This pairs dense semantic encoding with a lightweight indexing process which is the main component of the initial retrieval stage in the system. This guarantees that during subsequent KBQA stages, the most contextually relevant and semantically accurate candidates are chosen for additional progression and refinement.

### 4.2.3 Neural Reranking Model (Cross-Encoder + LLM)

To improve the correctness and context of the first retrieved candidate triples we implement a two-stage neural reranking. This step is built upon the semantic retrieval layer and it focuses on improving the results by capturing more fine-grained semantic dependency between the user query and the entries in the knowledge base. The reranking module has two main components: a Cross-Encoder for semantic relevance and a LLM (Large Language Model) to choose the final candidates and response formatting.

The first stage is Cross-Encoder reranking of the top-k triples retrieved by an embedding-based similarity search. In contrast to Bi-Encoders, which encode the query and candidate in isolation, the Cross-Encoder codes each query-triple pair as one input sequentially. This enables the model-to-model interactions from the query and candidate at the token level, capturing more nuanced contextual relationships. For each pair, the model assigns a relevance score that indicates the likelihood of the triple being a correct match for the question. Using these scores, triples are reprioritized so that triples that align the most are at the top of the candidate list. While this leads to much more accurate rankings, it is also significantly more computationally costly since each query-triple pair must be evaluated independently.

After the Cross-Encoder, a last reasoning step with a large language model (LLM), like DeepSeek-v3 or ChatGPT-4o, is included in the pipeline. The triples re-ranked as per our evaluation are concatenated with the original user query, and this combination is passed to the LLM to perform a contextual analysis from a rich perspective. The model analyzes the relations between the question and the retrieved facts, deduces the answer based on the information at hand, and generates a natural language response as a list of entities corresponding with the context. The last step assists in filling salient holes, enabling the model to give a final answer that is logical and meaningful, even if the query is poorly worded or not an actual question.

But although LLMs reasoning capabilities, they also open a measure of uncertainty. You are always at risk of hallucinating, creating content that, although linguistically correct, is

not factually true sometimes. The system leverages the structural grounding obtained in previous retrieval and reranking phases to constrain the generative output and to reduce the factual harm.

### 4.2.4 Final Answer Selection

At the last step of the pipeline, the LLM identifies the entities that are most pertinent to the user's inquiry and creates a response that most appropriately addresses the question. This phase reevaluates the candidate triples and employs the LLM's contextual reasoning to comprehend the question and generate a coherent, logical response. We rely on the Knowledge Base (KB) for structured retrieval to improve answer accuracy while the LLM's generative reasoning helps handle queries that require deeper semantic understanding and recognition of relationships between entities. This combined approach ensures that the final response is not only based on retrieved knowledge but also closely reflects the intended meaning behind the user's question a crucial feature in KBQA, where question phrasing can vary significantly.

## 4.3 Data Processing Pipeline

The approach relies on a three-step data processing pipeline consisting of (1) data pre-processing, (2) indexing, and (3) selecting the final answer.

The first stage of the pipeline consists of loading the Freebase Knowledge Base (KB) for data preprocessing. The freebase_extracted is the local extracted and parsed data file (.tsv). The first stage involves cleaning the dataset i.e., removing duplicates, irrelevant data, etc. Normalization is performed for readability. This means that underscores in relation names are replaced by space characters for readability. After the data is cleaned and formatted each Freebase triple, consisting of subject, relation, and object is encoded for dense vector representation using SBERT (Sentence-BERT) [10]. These embeddings are inserted into an Annoy index, which allows for fast nearest neighbor search when processing queries.

So, the second stage is all about indexing. An Annoy-based approximate nearest neighbor index is built on the SBERT representations of all Freebase triples. This allows efficient and scalable similarity search during inference. Another user submits a query, which is then also encoded into a vector using the same SBERT model. The Annoy index is queried to return the most semantically similar top-k triples based on the cosine similarity between the query vector and the embedding of stored triples.

The last step is the correct answer selection. The retrieved top-k triples from the Annoy index are treated as initial candidate knowledge. All these candidates are then fed into a Cross-Encoder model to perform re-ranking. [11] The Cross-Encoder encodes every (query, triple) pair together and predicts the relevance between them while sorting them accordingly. The highest-ranked triples are then fed as contextual input to a Large Language Model (LLM) like ChatGPT-4o or DeepSeek-v3. The LLM then considers both the query and the ranked, relevant knowledge to find the most relevant entities and to produce a fluent, context-aware answer that matches the user's intent.

## 4.4 Evaluation & Performance

In order to assess the performance of our KBQA system, we performed experiments on WebQuestions dataset which is a standard dataset for knowledge base question answering. We used official evaluation scripts offered by Chen et al. [6] to determine performance using the F1 score. This method imposes strict exact string matching, in that the predicted answer must exactly match the gold answer in order for it to be counted as correct. This provides a more conservative and exact measure of performance than some fuzzy matching techniques.

The data evaluation results indicate that ChatGPT-4o recorded the highest F1 score, surpassing the performance of the other two AIs, DeepSeek-v3 and ChatGPT-4o-mini. This result demonstrates the model's impressive accuracy to interpret the query and return the most relevant information from the knowledge base.

By leveraging SBERT for semantic embedding and indexing with Annoy, we successfully and efficiently retrieved candidate triples to our query, avoiding the limitations of symbolic search methods. Additionally, detailing a Cross-Encoder in the re-ranking stage improved the selected triples' accuracy according to better alignment with user query.

The main evaluation was based on exact match F1, but we also had another evaluation that looked at using SBERT to calculate the semantic similarity of predicted and gold answers. Now, this does offer a softer comparison and takes into consideration paraphrasing and conceptual similarity, where a strict matching might miss. Results from this supplemental assessment were reported in the appendix.

# 5 Experiment

## 5.1 Problem definition

The objective of this experiment is to assess the efficacy of our Knowledge Base Question Answering (KBQA) system. Upon receiving a natural language query from WebQuestions dataset specific from testset, the system seeks to determine the most relevant entities from Freebase through a process that includes semantic retrieval, neural reranking and the application of large language models (LLMs). The main challenges include:

**Lexical gap**: Users might express questions in a way that differs from how information is recorded in Freebase.

**Scalability**: Efficiently accessing relevant information from an extensive knowledge base.

**Ranking accuracy**: Making certain that the highest-ranked entities are genuinely corresponds to the query.

To solve these challenges our system employs SBERT-based retrieval, Annoy indexing, Cross-Encoder re-ranking, and LLMs for the final selection.

## 5.2 Data

For evaluation we use the WebQuestions dataset as mention by Wu et al. (2020) [16], which is a commonly used standard for KBQA. WebQuestions comprises natural language inquiries matched with structured responses obtained from Freebase. The dataset comprising 3778 training questions and 2032 questions for testing is especially valuable as it mirrors actual user inquiries and evaluates a system capability to connect them to structured knowledge base information.

Also, the validation set of WebQuestions was formed by splitting the training data into an 80% training portion (3023 samples) and a 20% validation portion (755 samples). Interestingly, approximately 85% of the inquiries can be resolved using just one Freebase predicate and there may be several valid answers for every question.

The Freebase knowledge base functions as a repository of factual information, structured in the form of (subject, relation, object) triples. Before it can be used the dataset undergoes preprocessing, which includes:

- Standardization to ensure consistency in the representation of queries.
- Adjusting the format to correspond with the retrieval and ranking process implemented in our system.

## 5.3 Results

The evaluation of model performance was performed using of F1-score, Precision and Recall where collectively measure the accuracy of the retrieved entities among gold-answers.

- **F1-score** provides a balance between precision and recall, ensuring that retrieved entities are both correct and comprehensive.
- **Precision** measures how many retrieved answers are actually correct.
- **Recall** indicates how many correct answers were successfully retrieved.

To assess the effectiveness of different models, we performed evaluations on ChatGPT-4o, DeepSeek-v3, and ChatGPT-4o-mini. The table below encapsulates their performance outcomes.

| Methods | F1 Score |
|---|---|
| **Bordes 2014b** [2] | 29.7 |
| **Bordes 2014a** [17] | 39.2 |
| **Yang 2014** [1] | 41.3 |
| **Dong 2015** [3] | 40.8 |
| **Bordes 2015** [18] | 42.2 |
| **Xu 2016** [4] | 42.2 |
| **Hao 2017** [5] | 42.2 |
| **Chen 2019 (topic entity predictor)** [6] | 51.8 |
| **Japa 2020** [7] | 52.7 |
| **Our Approach** | **52.74** |

*Table 5.1: Entity Predictor Results*

| Model | F1 Score |
|---|---|
| **ChatGPT-4o-mini** | 46.43 |
| **DeepSeek-V3** | 49.22 |
| **ChatGPT-4o** | **52.74** |

*Table 5.2: LLM Models (Entity Predictor) Results*

| **Pipeline Step** | **Time (minutes:seconds)** |
|---|---|
| **Indexing SBERT (Annoy)** | **2:47** |
| **Query Discovery Triplets for 2032 Questions** | **1:18** |
| **Gold-Query Discovery Triplets for 2032 Questions** | **2:35** |
| **LLM Models** | |
| **ChatGPT-4o-mini** | **30:24** |
| **ChatGPT-4o** | **32:44** |
| **DeepSeek-V3** | **121:15** |

*Table 5.3: Each pipeline step process time for 289483 triples and 2032 Questions*

# 6 Discussion

In this thesis we build a Knowledge-Based Question Answering (KBQA) system using Knowledge Bases (KB) and state-of-the-art NLP techniques. Structured data in Knowledge Bases like Freebase are tremendously helpful in constructing answers to natural language queries. Few-shot performance on NLP tasks like question answering and information retrieval have improved remarkably due to deep learning and transformer-based model architectures, which have shown state-of-the-art results in recent years.

To the best of our knowledge, this is the first study to systematically compare KBQA performance across a variety of combinations of semantic representations of KBs and state-of-the-art deep learning models including SBERT, Cross-Encoder (with Annoy Indexing), and Large Language Models (LLMs), namely DeepSeek and ChatGPT. We hypothesized that by leveraging these models together, we could achieve a model that produced more accurate and efficient question answering than typical models based on symbolic reasoning or rule-based models.

We performed one main experiment to assess the proposed KBQA system. The experiment dealt with semantic embeddings that were generated using the SBERT and Cross-Encoder models and judged based on their semantic capturing capabilities and structure representation in the KB. It also included assessing the ability of LLMs to perform direct question answering, exploring the trade-offs between utilizing large pre-trained language models and structured knowledge bases (KBs). Additionally, the experiment explored combining both methods to improve the accuracy and efficiency of tackling complex questions.

During the experiment, we tested different embedding qualities and found that the quality of embeddings significantly improves answer retrieval. When we generate embeddings using SBERT, we obtain a more robust representation. After applying Annoy Indexing, we achieved a significant speed improvement compared to traditional approaches. However, Annoy primarily helps with query times rather than retrieval recall or accuracy. Despite this, embedding dimensionality and data volume still posed challenges, especially in scenarios requiring instant results.

The experiment further revealed that while LLMs such as ChatGPT-4o and DeepSeek-v3 are capable of returning accurate answers for questions of lower and medium complexity, they often fail to provide specificity for questions that require precise entity linking or multi-hop reasoning. This limitation underlines the value of hybrid approaches that utilize knowledge base (KB) embeddings for more structured reasoning.

The results show that combining LLMs with KB embeddings outperforms benchmark data such as WebQuestions significantly, particularly in terms of accuracy and response time. This combined method alleviates the inherent weaknesses of each individual model, resulting in a more consistent and efficient KBQA system.

In summary, our results show that a hybrid KBQA system combining deep learning embeddings and LLM capabilities outperforms standalone methods. Future work should focus on optimizing this integration so that LLM reasoning with KB-based entity linking can become even more efficient for answering complex natural language questions.

# 7 Conclusions

In this section, a summary of the thesis is given, focusing on the most important aspects covered in the developed KBQA system and the experiments performed. In addition, we cover possible improvements and future research directions.

The aim of this thesis was to built a highly qualitative, robust Knowledge Based Question Answering (KBQA) system using KB and advanced Natural Language Processing (NLP) techniques. Freebase, a structured KB, had an enriched data source that could be leveraged to process natural language questions. With the emergence of transformer architecture and deep learning models, tasks such as question answering and information retrieval have seen significant advancements with state-of-the-art results.

We believed that adding semantics from KBs to contextualized representations from deep learning models will lead to better quantitative question and answering performance compared to deep neural network symbolic reasoning models. To do so, we have incorporated models (SBERT, Cross-Encoder) as well as large language models (LLMs) such as DeepSeek and ChatGPT-4o. We designed a system that is well able to integrate structured data with contextual reasoning. As part of this research, we proposed a hybrid framework that incorporated reasoning from LLMs and embeddings from knowledge bases. This approach enabled us to combine the structured information in the KB with contextual reasoning through the LLMs. Benchmark tests on datasets like WebQuestions showed that our hybrid system outperforms in topic entity predictor [7], showing the advantages of having a hybrid system.

For KBQA, we performed an experiment on comparing the SBERT-based embeddings with the LLM capabilities to see the performance. We used SBERT to obtain semantic embeddings because they were well able to capture relationships in the KB. To further evaluate the capabilities of deep learning models we examined LLMs, including ChatGPT-4o and DeepSeek-v3, and ChatGPT-4o-mini for direct question answering. Secondly, we found that ChatGPT-4o achieves the highest accuracy across both simple and moderately complex questions, along with DeepSeek-v3 and ChatGPT-4o-mini. All models struggled on questions that needed a more sophisticated knowledge representation.

While the outcomes are promising, they are not without challenges. Some of the most important issues identified include the management of embedding dimensionality and dealing with large amounts of data, all of which impact how good your system will be. Higher accuracy in KBQA would also require mindful augmentation of the representation of knowledge in LLMs.

As future work is important to improve the current KBQA pipeline. For a promising direction among cautious, one can enrich reasoning capabilities with multi-hop reasoning as many questions show relational features and require complex reasoning operations. We argue that multi-hop reasoning could be of critical use in this area, since presented questions are more and more relationally complex, in terms of logistical reasoning. It is also important to improve knowledge representation within LLMs in order to achieve accurate and relevant question answering. Improving the interfacing between LLMs and KB embeddings will be a key factor to guarantee systems robustness. We will need also to optimize the generation of embeddings as the system scales out. Overcoming these hurdles will help push the field onward. We thus believe that KBQA systems with hybrid KBQA methods using deep learning methods along

with other KB concept recognition methods and structured reasoning-based factual knowledge will achieve good results.

# 8 Appendix

## 8.1 A SBERT-Based Semantic Evaluation

Alongside the main evaluation that follows the official WebQuestions script with the exact string matching, we also performed a secondary evaluation with Sentence-BERT (SBERT) that focuses on the semantic similarity between predicted answers and gold answers. It articulates a more nuanced and flexible view of answer quality allowing, for example in cases where the system output is semantically similar but otherwise slightly different in terms of wording or formatting from the gold answer.

We calculated cosine similarity between SBERT embeddings of each predicted answer and the corresponding gold answer. If this similarity was greater than a predefined threshold for instance 0.80, the prediction was seeming a semantically acceptable match. This enabled us to obtain partial and paraphrased answers that would otherwise be penalized according to exact match evaluation.

This semantic evaluation has two-part targets:

1. To highlight few instances in which the model shows partial understanding or paraphrasing capability. Valuable in real KBQA scenarios.

2. To support the strict F1 score evaluation and provide additional context to help inform model behavior.

The outcome of this semantic evaluation is shown in the figures and table below:

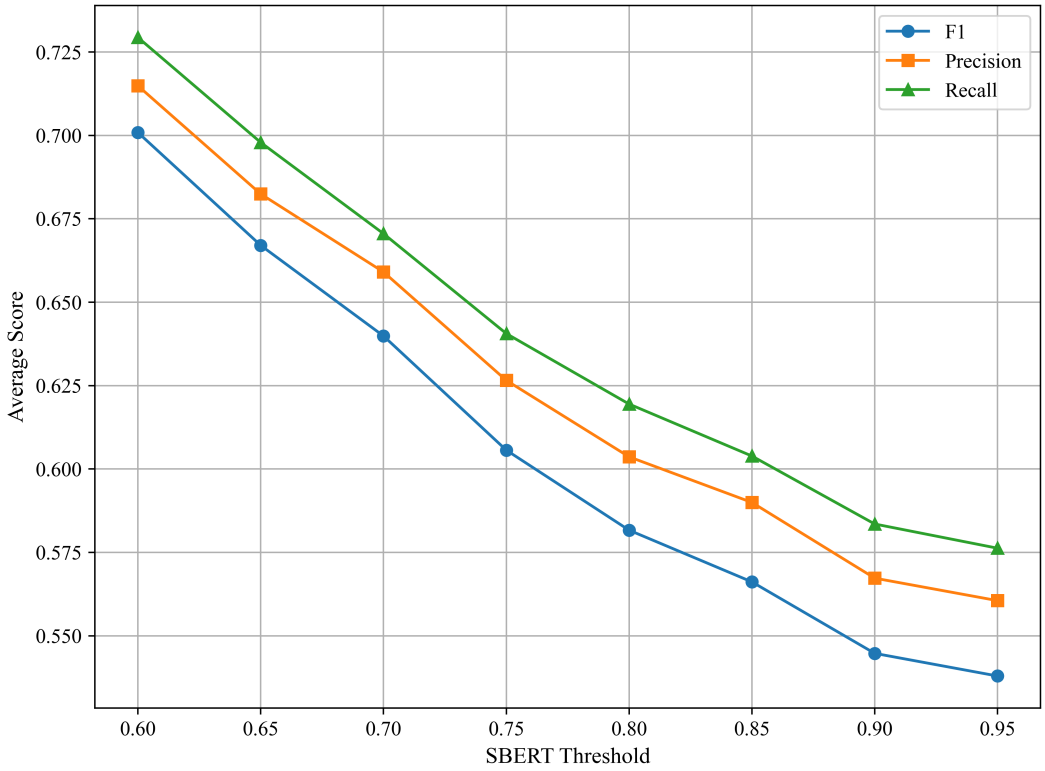


*Figure 8.1: Average Precision, Recall, and F1 Score vs. SBERT Threshold for ChatGPT-4o LLM.*

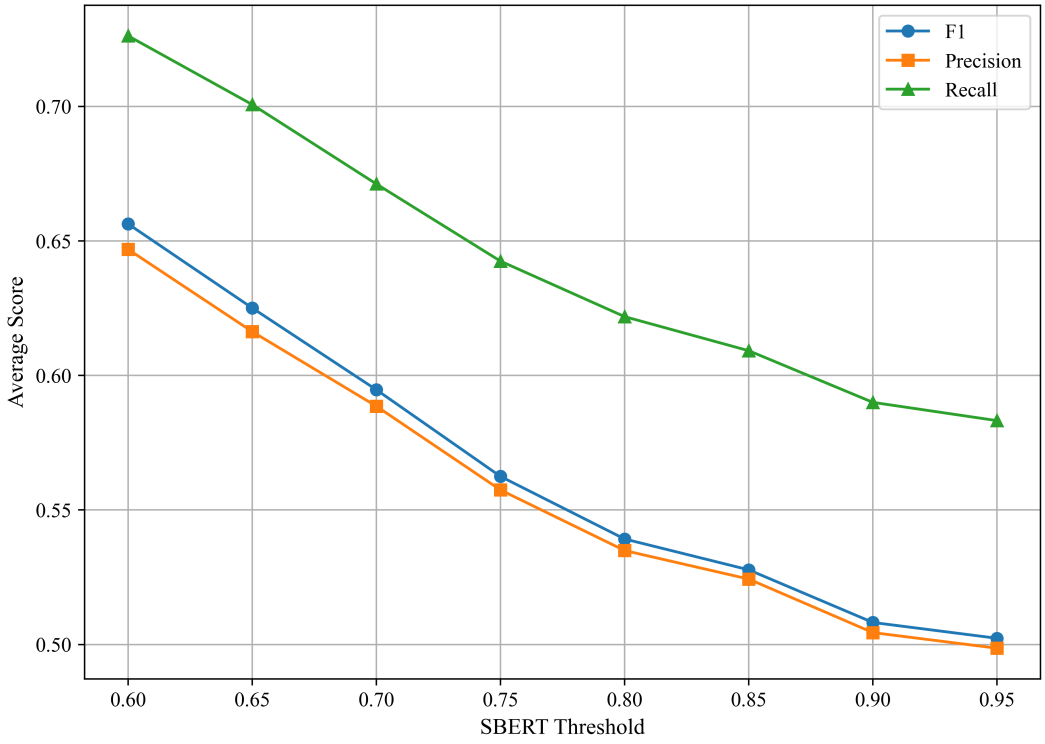


*Figure 8.2: Average Precision, Recall, and F1 Score vs. SBERT Threshold for DeepSeek-V3 LLM.*

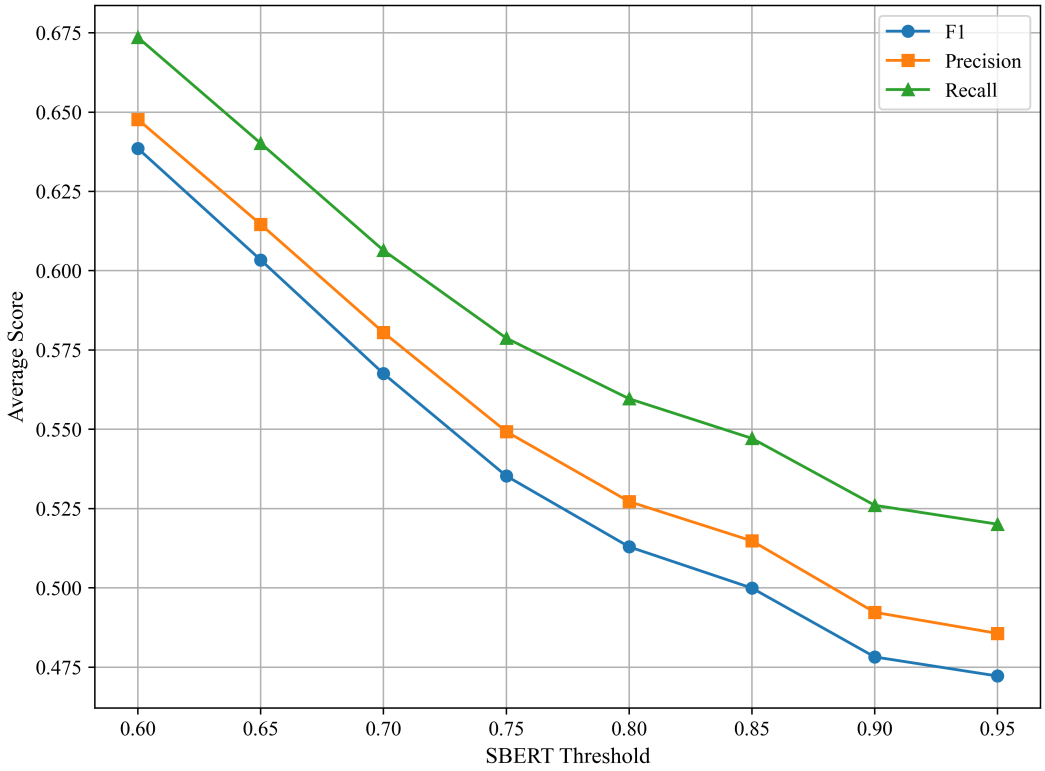


*Figure 8.3: Average Precision, Recall, and F1 Score vs. SBERT Threshold for ChatGPT-4o-mini LLM.*

| Threshold | ChatGPT-4o Average F1 | DeepSeek-V3 Average F1 | ChatGPT-4o-mini Average F1 |
|---|---|---|---|
| **0.6** | **0.700** | 0.656 | 0.638 |
| **0.65** | **0.667** | 0.625 | 0.603 |
| **0.7** | **0.639** | 0.594 | 0.567 |
| **0.75** | **0.605** | 0.562 | 0.535 |
| **0.8** | **0.581** | 0.539 | 0.512 |
| **0.85** | **0.566** | 0.527 | 0.499 |
| **0.9** | **0.544** | 0.508 | 0.478 |
| **0.95** | **0.538** | 0.502 | 0.472 |

*Table 8.4: Average F1 Score vs. SBERT Threshold per LLM.*

## 8.2 Threshold Sensitivity and Optimal Performance Range

The evaluations utilizing the SBERT approach for various thresholds of cosine similarity have yielded comparable results. The trend of F1 scores is consistently declining as the degree of specified threshold changes from strict to lenient. This implies a difficulty in ensuring the adequate balance between semantic similarity and flexibility in recognition of alternative phrasing that are still acceptable linguistic output.

In summary, the overall F1 scores for all three models, ChatGPT-4o, DeepSeek-V3 and ChatGPT-4o-mini, exhibit peak values at a threshold of 0.60. Specifically, F1 scores show that ChatGPT-4o scores the highest at 0.70, following is DeepSeek-V3 with 0.656 and ChatGPT-4o-mini in third position with a score of 0.638. All of these scores will be considered the identified peak scores based on the evaluation conducted on all three models.

As shown and indicated as follows, the F1 scores shows a downward trend from thresholds greater than 0.60, at a threshold of 0.65, the scores begin to demonstrate a cumulative downward trend. At the final, highest threshold of 0.95, the F1 scores decrease to 0.538 for ChatGPT-4o, 0.502 for DeepSeek-V3, and 0.472 for ChatGPT-4o-mini, overall demonstrating and indicating that stricter similarity specifications tend to improve accuracy, but can also exclude targeting still correct answers which provide greater flexibility in phrasing from the decisively perceived gold standard.

The results of the overall evaluations show potential tradeoff between specificity and accuracy particularly at the threshold between 0.60 and 0.65, in part due to all three models demonstrating strong performance compared to each other, however the total performance of the models in terms of overall score based on the answers with all three approaches. This is to say that balance is important in real world KBQA systems and application as multiple correct phrasings may occur delivering the same answer across multiple answer phrases depending on the phrasing structure and subject of the question posed on the KBQA task.

The results from these additional metrics are not considered the primary form of measure in this thesis, but provided here as a means to perceive possible benefits or drawbacks from the models than cannot be detected by simple exact match methods only.

# 9 Bibliography

[1] M. C. Yang, N. Duan, M. Zhou, and H. C. Rim, "Joint Relational Embeddings for Knowledge-based Question Answering," in *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, Association for Computational Linguistics (ACL), 2014, pp. 645–650. doi: 10.3115/V1/D14-1071.

[2] A. Bordes, J. Weston, and N. Usunier, "Open Question Answering with Weakly Supervised Embedding Models," in *Machine Learning and Knowledge Discovery in Databases*, T. Calders, F. Esposito, E. Hüllermeier, and R. Meo, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 165–180.

[3] L. Dong, F. Wei, M. Zhou, and K. Xu, "Question Answering over Freebase with Multi-Column Convolutional Neural Networks," in *ACL-IJCNLP 2015 - 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Proceedings of the Conference*, Association for Computational Linguistics (ACL), 2015, pp. 260–269. doi: 10.3115/V1/P15-1026.

[4] K. Xu, S. Reddy, Y. Feng, S. Huang, and D. Zhao, "Question Answering on Freebase via Relation Extraction and Textual Evidence," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Katrin Erk and Noah A. Smith, Eds., Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 2326–2336. doi: 10.18653/v1/P16-1220.

[5] Y. Hao *et al.*, "An End-to-End Model for Question Answering over Knowledge Base with Cross-Attention Combining Global Knowledge," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Regina Barzilay and Min-Yen Kan, Eds., Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 221–231. doi: 10.18653/v1/P17-1021.

[6] Y. Chen, L. Wu, and M. J. Zaki, "Bidirectional Attentive Memory Networks for Question Answering over Knowledge Bases," 2019. Accessed: Mar. 06, 2025. [Online]. Available: https://arxiv.org/abs/1903.02188

[7] S. S. Japa and R. Banafsheh, "Question Answering over Knowledge Base using Language Model Embeddings," in *2020 International Joint Conference on Neural Networks (IJCNN)*, Glasgow, UK: IEEE, Jul. 2020, pp. 1–8.

[8] J. Devlin, M.-W. Chang, K. Lee, K. T. Google, and A. I. Language, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 4171–4186. doi: 10.18653/V1/N19-1423.

[9] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2008, pp. 1247–1249. doi: 10.1145/1376616.1376746.

[10] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks,” in *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, Association for Computational Linguistics, Aug. 2019, pp. 3982–3992. doi: 10.18653/v1/d19-1410.

[11] S. Humeau, K. Shuster, M.-A. Lachaux, and J. Weston, “Poly-encoders: Transformer Architectures and Pre-training Strategies for Fast and Accurate Multi-sentence Scoring,” Apr. 2019, Accessed: Apr. 07, 2025. [Online]. Available: https://arxiv.org/abs/1905.01969v4

[12] OpenAI *et al.*, “GPT-4o System Card,” Oct. 2024, Accessed: Apr. 09, 2025. [Online]. Available: https://arxiv.org/abs/2410.21276v1

[13] R. Islam and O. M. Moushi, “GPT-4o: The Cutting-Edge Advancement in Multimodal LLM,” *Authorea Preprints*, Jul. 2024, doi: 10.36227/TECHRXIV.171986596.65533294/V1.

[14] DeepSeek-AI *et al.*, “DeepSeek-V3 Technical Report,” Dec. 2024, Accessed: Apr. 09, 2025. [Online]. Available: https://arxiv.org/abs/2412.19437v2

[15] E. Bernhardsson, “Annoy: Approximate Nearest Neighbors in C++/Python.” Accessed: Apr. 09, 2025. [Online]. Available: https://github.com/spotify/annoy

[16] W. Wu, Z. Zhu, Q. Lu, D. Zhang, and Q. Guo, “Introducing External Knowledge to Answer Questions with Implicit Temporal Constraints over Knowledge Base,” *Future Internet 2020, Vol. 12, Page 45*, vol. 12, no. 3, p. 45, Mar. 2020, doi: 10.3390/FI12030045.

[17] A. Bordes, S. Chopra, and J. Weston, “Question Answering with Subgraph Embeddings,” in *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, Association for Computational Linguistics (ACL), Jun. 2014, pp. 615–620. doi: 10.3115/v1/d14-1067.

[18] A. Bordes, N. Usunier, S. Chopra, and J. Weston, “Large-scale Simple Question Answering with Memory Networks,” Jun. 2015, Accessed: Mar. 29, 2025. [Online]. Available: https://arxiv.org/abs/1506.02075v1