Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2024) 32: 516 – 533

doi:10.55730/1300-0632.4085

Enrichment of Turkish question answering systems using knowledge graphs

Okan ÇİFTÇİ[1], **Fatih SOYGAZİ**[2*], **Selma TEKİR**[1]
[1]Department of Computer Engineering, Faculty of Engineering, İzmir Institute of Technology, İzmir, Turkiye
[2]Department of Computer Engineering, Faculty of Engineering, Aydın Adnan Menderes University, Aydın, Turkiye

Received: 26.11.2023 • **Accepted/Published Online:** 24.04.2024 • **Final Version:** 26.07.2024

Abstract: Recent capabilities of large language models (LLMs) have transformed many tasks in Natural Language Processing (NLP), including question answering. The state-of-the-art systems do an excellent job of responding in a relevant, persuasive way but cannot guarantee factuality. Knowledge graphs, representing facts as triplets, can be valuable for avoiding errors and inconsistencies with real-world facts. This work introduces a knowledge graph-based approach to Turkish question answering. The proposed approach aims to develop a methodology capable of drawing inferences from a knowledge graph to answer complex multihop questions. We construct the Beyazperde Movie Knowledge Graph (BPMovieKG) and the Turkish Movie Question Answering dataset (TRMQA) to answer questions in the movie domain. We evaluate our proposed question answering pipeline against a baseline study. Furthermore, we compare it with a question answering system built upon GPT-3.5 Turbo to answer the 1-hop questions from TRMQA. The experimental results confirm that link prediction on a knowledge graph is quite effective in answering questions that require reasoning paths. Finally, we provide insights into the pros and cons of the provided solution through a qualitative study.

Key words: Knowledge representation and reasoning, question answering systems, natural language processing, deep learning, graph embeddings

1. Introduction

Despite the remarkable advances achieved in developing question answering (QA) systems in the field of Natural Language Processing (NLP), particularly for widely spoken languages like English, there remains a considerable gap in the development of such systems for less frequently used languages. This gap is primarily due to insufficient training data, which poses a significant challenge in building effective QA models for low-resource languages like Turkish. The existing Turkish question answering (TRQA) systems are mostly designed to comprehend a given query, rank the documents based on their relevance to the query, and present the most relevant document to the user [1, 2].

The state-of-the-art QA systems rely on contextual language models, performing well in providing proper answers. However, their factuality suffers. Knowledge graphs, which represent facts as triplets, can be valuable for avoiding errors and inconsistencies with real-world facts.

Knowledge graphs are an essential tool for structured knowledge representations in artificial intelligence. In recent years, transformer-based methods have paved the way for representing nodes and relations in more powerful ways [3–5]. The present study introduces a knowledge graph-based approach to question answering for

*Correspondence: fatih.soygazi@adu.edu.tr

Turkish. The proposed approach aims to develop a methodology capable of drawing inferences from a knowledge graph to answer complex multihop questions.

We construct the Beyazperde Movie Knowledge Graph (BPMovieKG) using web data crawling techniques as part of the methodology. Furthermore, we introduce the TRMQA (Turkish Movie Question Answering) dataset, created by utilizing BPMovieKG to generate various question types and prepare question templates for $1-2$ and 3-hop reasoning. We adopt a deep learning architecture to perform question answering tasks on the graph. Inside the architecture, we compare the performance of different graph embedding methods and BPMovieKG against knowledge bases in literature. Furthermore, we evaluate question embedding techniques based on results from the question-answering system. Finally, we provide quantitative and qualitative analyses of TRMQA using GPT-3.5 Turbo [6] prompts.

One of the primary contributions of the study is to create a knowledge graph (BPMovieKG) in the movie domain for the Turkish language. Additionally, the study introduces the first question-answering system developed for Turkish on the knowledge graph using embeddings. In this context, various contemporary methods for question embedding and graph embedding, which are the two different stages aimed at improving the results over a baseline, were tested on the Turkish KG (BPMovieKG), and the results were compared. The comparisons focused on accessing the answer on the KG through a single edge (1-hop) and providing 1-hop, 2-hop, and 3-hop answers in the applicable cases. These results were then compared on the dataset we created (BPMovieKG) and English benchmark datasets (WN18, FB15k) to evaluate the success of the experimented pipeline. Instead of using a single KG embedding and question answering embedding method as in the baseline, we tested combinations of these methods on BPMovieKG and TRMQA to discuss the results of the most successful combination. Finally, we demonstrated the instances where ChatGPT-3.5 Turbo provided incorrect answers to 1-hop questions. In these situations, we explained how our proposed method yielded accurate results based on facts in the KG. We applied well-studied embedding approaches to achieve the best results in a pipeline where these approaches could be used together.

The contributions of this paper are as follows:

- We introduce the first Turkish QA system that utilizes knowledge graphs for multihop reasoning. To the best of our knowledge, there is no work on knowledge graphs or knowledge graph embeddings for Turkish question answering systems.

- We conduct a comparison between the OpenAI GPT-3.5 Turbo, utilizing the LlamaIndex [1] and a Turkish question answering system for the first time in the literature. Additionally, we compare the graph generated in this study with benchmark graphs commonly used in the literature, based on the results obtained from various graph embedding methods.

- We introduce two different datasets to contribute to Turkish question-answering systems. The first is a knowledge graph in the movie domain, and the second is a set of questions related to this knowledge graph that require $1-2$ and 3-hop reasoning.

The remaining parts of this paper are organized as follows. Section 2 describes the related work on knowledge graphs and question answering in literature. Section 3 begins by explaining knowledge graphs, followed by the construction of BPMovieKG. The section concludes with statistics regarding the BPMovieKG. Section 4 describes the creation of TRMQA from BPMovieKG. Section 5 elaborates on our method to capture

[1] https://github.com/jerryjliu/llama_index

answers from the knowledge graph. Section 6 presents our experiments and their results. Finally, Section 6 concludes the paper and presents future work. Relevant datasets and source codes are available at Github[2].

2. Related work

Extensive research has been conducted to develop methods for answering natural language questions in various languages. Within the wide range of approaches proposed for question answering, including rule-based, statistical, and neural methods, this study explicitly targets modern developments in neural question answering. Our research focuses on two distinct domains of inquiry, namely Turkish question answering and Knowledge Base question answering.

2.1. Turkish question answering

In Turkish question-answering literature, different approaches have been proposed. Derici et al. [7] construct a closed-domain question answering system operating in two phases: question analysis and information retrieval. The question analysis module extracts the focus to guide the information retrieval system. The information retrieval system utilizes a focused approach by posing a question and searching for related documents on search engines.

Celebi et al. [8] construct a similar pipeline except for finding focus. They utilize questions as queries and categorize them using named entity recognition and pattern matching. First, various preprocessing steps are performed on the documents. Each document is tokenized, and stemming is applied to obtain the base forms of words for each token. Following that, patterns are prepared to identify named entities and keywords. Subsequently, the extracted keywords and named entities are stored in the database for each document. After that, a ranking metric is proposed for the question answering system. Pronouns and entity names are extracted from the given questions, and based on this metric, the question answering system is constructed.

Derici et al. [9] propose a question-answering framework, HazırCevap, which gets the question in natural language, parses and converts it into a query to retrieve a set of relevant documents, and applies summarization to them to present potential answers. Their system uses reliable sources as the input document collection and can process sources in other languages through an integrated translation component. HazırCevap supports both factoid and open-ended question answering.

Tasar et al. [10] address the challenge of question answering by leveraging the semantic web as a valuable knowledge resource. This study uses various NLP techniques to understand the question, create relationships between extracted named entities, and convert the question into a SPARQL query.

Yigit and Amasyali [11] contribute to Facebook's bAbi dataset [12] for Turkish. The bAbi dataset consists of twenty tasks for text understanding and reasoning. The authors aim to improve dynamic memory networks' input and attention modules in their proposed work and demonstrate that their method improved accuracy in various tasks for both languages.

Soygazi et al. [13] released THQuAD, a Turkish reading comprehension dataset constructed from Turkish Wikipedia articles. The authors provided baseline performances due to some BERT variants.

Menevşe et al. [14] released the first spoken question-answering dataset in Turkish. In construction, they rely on a pretrained multilingual Transformer for question generation and speech recognition parts and perform fine-tuning with respective manually annotated data.

[2]https://github.com/okanvk/Enrichment-of-Turkish-Question-Answering-Systems-using-Knowledge-Graphs

Gemirter and Goularas [15] propose an approach for reading comprehension. The proposed method uses a given paragraph and question to predict the start and end indices of the answer within the given paragraph. In this study, a language model based on the transformer architecture is fine-tuned to find answers to various questions posed on banking sector documents. This study presents the first Turkish question answering system using a transformer-based architecture and also introduces the first machine-reading comprehension dataset designed explicitly for the Turkish language.

Akyön et al. [16] use a multilingual language model to predict the relevant span within a given passage for a given question or to generate a question based on a given passage and its corresponding answer for historical text data for the question generation and machine reading for question answering tasks. In this study, the generation of Turkish questions from Turkish texts has been performed for the first time in the literature.

2.2. Knowledge base question answering (KBQA)

Knowledge base question answering entails reconciling natural language questions with reasoning paths in knowledge bases. The former brings its open-domain knowledge through pretrained language models, while the latter contributes to the factuality of the provided answers by its structure. Thus, the methods in this area should bridge the gap between unstructured text and structured knowledge base.

Bordes et al. [17] propose a new approach for answering simple questions using a memory network architecture. With this approach, authors use a memory module to learn the relationship between a question and a knowledge base by embedding them in the same vector space.

Cui et al. [18] realize KBQA by employing templates to map questions to predicates in KBs. In preparing templates, they perform conceptualization (selecting upper-level concepts for entities) on KBs. They learn template-predicate mappings using the Yahoo! Answer corpus. Their system models complex questions as expanded predicates, paths of multiple edges from an entity to a specific value, and can answer complex and binary factoid questions.

Saxena et al. [19] propose a method to answer multihop questions on a given knowledge graph. In this study, ComplEx is used to learn graph embeddings. In addition, a neural network architecture was utilized to learn the representations of natural language questions.

Sorokin and Gurevych [20] are the first researchers to use gated graph neural networks (GGNN) for KBQA. Their system represents a question using deep convolutional neural networks and learns to construct a semantic graph for a given question by training pairs of questions and semantic graphs. The approach can answer complex questions significantly better compared to the nongraph baselines.

In their approach, De Cao et al. [21] use graph convolutional neural networks (GCNs) for multidocument question answering. In multidocument question answering, one must answer a given question by reasoning across a document collection. They frame the task as an inference problem on an entity-GCN where nodes are entities and edges are their mentions. Their system achieves state-of-the-art results on the WIKIHOP dataset.

Yan et al. [22] encode the natural language question along with the linearized KG paths using BERT and perform further fine-tuning in relation extraction, relation mapping, and relation reasoning (RR) tasks. RR deals with the issue of KB's incompleteness. The experimental results on the WebQSP dataset prove the approach's effectiveness, especially when the KB is incomplete.

Ravishankar et al. [23] propose a framework to decouple the semantic parse of a natural language question from its final SPARQL query in the task of KBQA to generalize over different knowledge graphs. In the first stage of the workflow, the authors adopt an encoder-decoder architecture to transform the question into a generic

query skeleton. After the entity linking and relation text to KG relation mappings at the succeeding stage, the KG-specific SPARQL query is constructed. Their system provides significant performance improvement in predicting unseen relation combinations.

Recent methods combine knowledge graphs with a text corpus [24, 25]. Such methods are advantageous when the knowledge graph or text corpus is incomplete. Both methods present a new approach for open-domain question answering systems that use two different sources, structured and unstructured data.

This work is mainly related to [19] since both use knowledge graph embeddings for multihop question answering over sparse KGs. The main distinction is that our work is a pipeline for Turkish question answering that operates on BPMovieKG and TRMQA. Furthermore, we test the effect of different question and KG embedding methods' on the performance. We also provide insights into the performances of these different KG embedding methods by running them on input KGs of varying sparsity.

In Turkish question answering, our work is close to [10] as both approaches process a question in natural language and return an answer based on a knowledge base. Their method converts the question into its SPARQL equivalent to query an ontology, and they test their framework on GEO-TR, a novel Turkish ontology in geography. However, our approach formulates the problem as an entity prediction on a graph where the task is to determine the closest entity in the knowledge graph given the embeddings for all entities and the question embedding. Moreover, we run our experiments on the Beyazperde Movie Knowledge Graph (BPMovieKG). As our approach follows the link prediction technique on BPMovieKG, it has the advantage of exploring potential connections and uncovering hidden relationships by utilizing the natural language understanding capabilities of question embedding and graph embedding modules. Additionally, our approach assists in handling a large-scale knowledge graph efficiently through link prediction to answer 1-hop, 2-hop, or 3-hop questions.

3. Dataset

In this work, we construct a knowledge graph, BPMovieKG, on the movie domain. Then, we form the TRMQA dataset that contains 1-2 and 3-hop questions related to BPMovieKG.

A knowledge graph is a structured knowledge representation consisting of a collection of entities and relations represented as a set of triples. Each triple in the knowledge graph represents a basic unit of information consisting of a subject, relation, and object. The notations and their descriptions about the knowledge graphs are listed in Table 1.

Table 1. Knowledge graph notations.

Notation	Definition
G	A knowledge graph
V	The set of nodes or entities in G
E	The set of edges or relations in G
F	The set of triples in G
e_i	An entity node in G
r_j	A relation edge in G
(e_i, r_j, e_k)	A triple in G connecting entity nodes e_i and e_k with a relation r_j
N_G	The number of nodes in G
R_G	The number of relations in G
T_G	The number of triples in G

3.1. BPMovieKG construction

BPMovieKG construction starts by obtaining the relevant information through crawling the famous Turkish movie website beyazperde[3]. Then, the preprocessing of crawled resources follows as illustrated step by step in Figure 1.

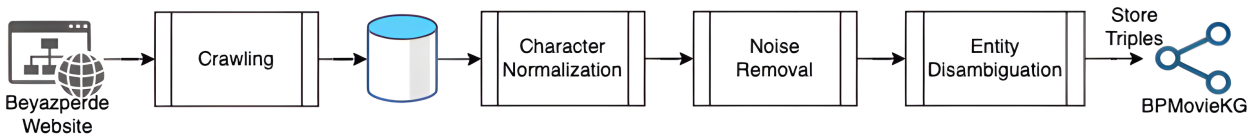


Figure 1. The construction process of BPMovieKG.

In the crawling process, we use the libraries bs4[4] and selenium[5]. We collect movie descriptions and metadata such as the release year, genre, language, budget, runtime, rating, actor, director names, professions, birth dates, and nationalities.

The preprocessing of crawled resources is performed as follows:

Character normalization The crawled data contains different typos, conflicts, and exceptions. For instance, though the nodes Yapımcı and Yapimci refer to the same entity, their forms are different, the latter with the English letter i rather than the Turkish ı. Another variation is that a movie name appears with or without the consecutive word filmi, such as Savaş and Savaş filmi. To resolve the entities correctly, we group the nodes by their types and edit them manually.

Noise removal As for the unknown nodes, we remove them automatically from the graph. The node types, such as budget, runtime, and birth dates, cause a decrease in graph density. Thus, we also drop them from the graph.

Entity disambiguation In the crawled data, we observe name conflicts in actor and director node types. To fix this issue, we concatenate node names with unique IDs. To resolve a similar name conflict in movies, we compare the movie nodes based on the in-degree centrality and remove the far less popular one from the graph. To calculate in-degree centrality, we used

$$C_{\text{in}}(e_i) = \frac{\deg_{\text{in}}(e_i)}{V-1} \tag{1}$$

(1) calculates the in-degree centrality of a node by dividing the number of incoming edges by the total number of nodes in the network minus one.

After crawling and preprocessing, we collect all remaining triples in a single graph and name it BP-MovieKG. Appendix 6 shows the subgraph that belongs to the movie Thor Karanlık Dünya.

The constructed BPMovieKG contains $317,992$ triples T_G, which relate to $36,489$ unique nodes N_G and 16 unique relations R_G. Each relation is created bi-directionally to ensure compatibility with graph embedding methods commonly relying on undirected graphs to learn embeddings.

[3] http://beyazperde.com
[4] https://pypi.org/project/beautifulsoup4/
[5] https://selenium-python.readthedocs.io/

Table 2 includes the node types with their counts. Here, actors and directors are merged into the node type Person.

Table 2. Distribution of node types in BPMovieKG.

Node Type	Node Count
Person	25,187
Movie	10,958
Nationality	100
Year	93
Language	59
Profession	54
Genre	28
Rating	10

3.2. TRMQA construction

TRMQA has 8 types of 1-hop, 19 types of 2-hop, and 14 types of 3-hop questions. Please refer to our GitHub repository for the exact question types and examples. To prepare the TRMQA dataset, we first design question types for each hop. For each question type, we wrote various questions manually to ensure the diversity of different types of questions. Each question type represents a path on the knowledge graph. We examined how many times each question type occurs on our knowledge graph. For each path we find, we assign the starting node of the path to the entity mentioned in the question, and the end node(s) of the path is provided as the answer(s) to the question. We perform uniform random sampling when we select questions from the question pool to construct TRMQA. The purpose is to avoid bias on the created partitions by randomly sampling questions from the question sets for each question type. TRMQA consists of diverse question types that exploit bidirectional knowledge through relations to infer an answer effectively.

In creating TRMQA, we are inspired by MetaQA benchmark [26, 27]. The purpose of publishing the MetaQA benchmark is to expand the existing WikiMovies dataset in the literature and to release a multihop question dataset in the movie domain to evaluate models with reasoning capabilities.

In preparing the questions, we examined the question types within MetaQA and created corresponding question types in TRMQA. Additionally, we developed new question types for relations existing in BPMovieKG but not covered in MetaQA. One author manually wrote each set of questions for the created question types; another reviewed and made necessary corrections. You can find all the question types with examples in Appendix 6.

This study inspired us to add similar question types in the movie domain. In contrast to MetaQA, wherein the knowledge graph includes distinct entities bearing identical names, resulting in inconsistencies between questions and their respective answers, we resolve such inconsistencies by generating our knowledge graph wherein a unique name designates each entity. For instance, in MetaQA, the following question is asked: 'Who is the writer of School for Scoundrels?' The method used in the baseline provides the answer 'Stephen Potter' to this question, but in the dataset, the answer is stated as 'Todd Phillips.' However, within the knowledge base, there are two different nodes with the same name, School for Scoundrels, and one is associated with Todd Phillips as the writer. In contrast, the other is associated with Stephen Potter as the writer.

Table 3 presents the statistics of the train, test, and development partitions of the TRMQA dataset. We stratified question types equally among the train, test, and development sets, using the split ratios of

$0.9 : 0.05 : 0.05$, $0.88 : 0.065 : 0.065$, and $0.91 : 0.045 : 0.045$. Appendices 6, 6, and 6 depict the examples of 1-hop, 2-hop, and 3-hop questions, respectively. Initially, we created the test and development datasets with the requirement that there should be at least one example from the existing question sets for each question type. This way, we utilized the test and development sets for every question we crafted. Additionally, we included 5% of the generated questions for each question type in the test set and another 5% in the development set. Due to variations in the number of questions for each question type, the ratios differed accordingly. After forming these datasets, we used the remaining questions in the training set.

Table 3. Statistics for TRMQA.

Dataset	1-hop	2-hop	3-hop
Train	250,296	287,081	214,688
Test	15,389	21,636	10,946
Dev	15,388	21,629	10,938

To answer a question like "Hangi kişi Karayip Korsanları Salazar'ın intikamı filminde yönetmendir?", it is sufficient to go through one edge on the graph. The associated reasoning path is shown in Appendix 6.

On the other hand, the answer to the question "Onur Saylak'ın oynadığı filmler hangi temalardadır?" requires a two-hop connection on the movie graph. First, we need to infer the movies by the given actor and then find the genres of the inferred movies. Appendix 6 illustrates the reasoning path for this example.

Finally, a three-step reasoning is necessary to answer the question "Eric Reed'in yönettiği filmlerin diğer yönetmenleri hangi uyruktan gelmektedir?". First, we need to list the movies by the given director. Next, we find the other directors of the listed movies. Finally, we ask for the nationalities of these directors. Appendix 6 depicts the associated reasoning path.

4. Method

This section presents our approach, consisting of three distinct modules: Knowledge Graph Embedding, Question Embedding, and Answer Selection. These modules are designed to handle different stages of the question answering process, from capturing semantic relationships in the KG to encoding natural language questions and selecting the best answers from a set of candidates. Figure 2 illustrates the structural design of our approach.

The baseline study (Saxena et al. [19]) relies on the available KG facts that cover the entities V and relations E in a knowledge graph G. It represents facts as triples $F \subseteq VxExV$. The problem formulation is to extract the entity $e_f \in V$ in a question q where the topic entity $e_t \in V$ should match the factual entity e_f.

Saxena et al. [19] name their work EmbedKGQA, which uses graph embeddings for responding to multihop natural language questions. Initially, it acquires a representation of the KG in an embedding space. Subsequently, when presented with a question, it generates a question embedding. Ultimately, it merges these embeddings to make predictions and determine the answer. Through these steps, we have generated our design similarly, where we have applied various question embedding and graph embedding techniques. The difference is that they use the ComplEx graph embedding approach with a BiLSTM and three fully connected layers on top to determine the correct answer. In our study, we extract the representations of questions using a fine-tuned SBERT model and then apply mean pooling along with three fully connected layers (Figure 2). In addition to the ComplEx model used in the baseline study, our study uses the DistMult and SimplE models.

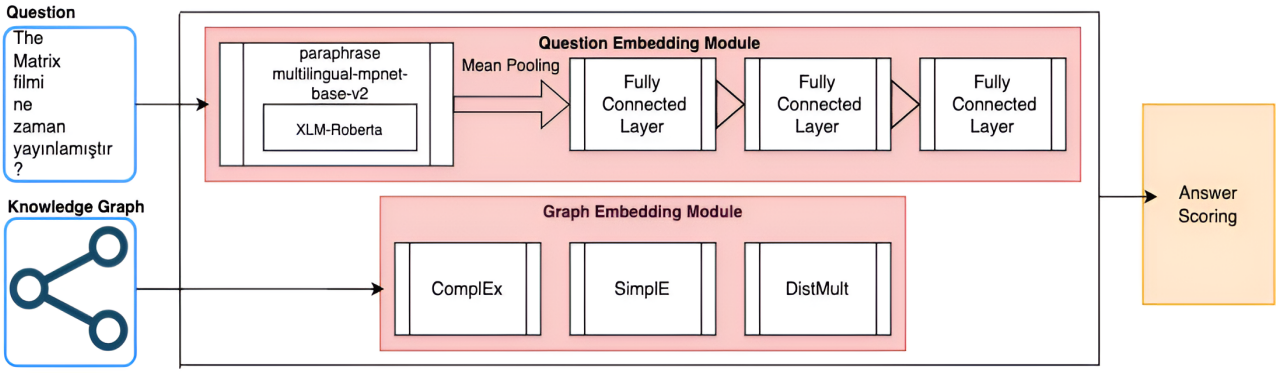


Figure 2. The structural design of our approach.

4.1. KG embedding module

Recently, researchers have proposed methods to embed structured data, such as graphs. We used KG embeddings, e.g., DistMult, ComplEx, and SimplE, to encode a large-scale knowledge graph rather than working with sub-graph encodings because constructing a question-specific sub-graph from the KG depends on the availability of relevant text corpora.

DistMult [28] proposes a solution for the link prediction task to represent entities in the head and tail positions with the same embedding. It measures head and tail entity compatibility using an entry-wise product. Since the entry-wise product in this work is symmetric, this technique is unsuitable for asymmetric and antisymmetric relations.

The authors use a link prediction task in ComplEx [29] to learn graph embeddings. They propose a tensor factorization-based model for learning embeddings. Instead of real vector space, they embed vectors in complex space. So, each entity has two imaginary parts and one real part, represented by a complex vector. The model is trained to optimize a scoring function for generating embeddings of $\mathbf{h}$, $\mathbf{r}$, $\mathbf{t}$ using the Hermitian dot product. The output of the scoring function should be greater than zero if triple is true; otherwise, it should be lower than zero. This method learns both symmetric and antisymmetric relations using the complex space.

SimplE [30] proposes a new embedding method to improve DistMult and ComplEx embedding methods since they have limitations in capturing complex relations between entities. They introduce a new scoring based on the element-wise product of two embedding vectors instead of a dot product.

KG Embedding module was used to obtain graph embeddings to provide as input to the Answer Selection module. In this module, we constructed node and relation embeddings from the constructed knowledge graph using DistMult, ComplEx, and SimplE methods. We trained the KG Embedding module independently, and the weights in this module were frozen during the training of the Question Answering System. Before training, triples are divided into train, test, and development sets based on $0.9 : 0.05 : 0.05$ split ratio, and the relationships were stratified during the process.

The training process was designed to ensure that all nodes in the BPMovieKG were included in the training data. This approach enabled the model to learn representations for each node because, during training, each node created a loss score that could be optimized using the available graph embedding methods.

4.2. Question embedding module

This module constructs a vector representation for a given question. We use SBERT [31] as the question embedding method. Specifically, we use a pretrained multilingual model called paraphrase-multilingual-mpnet-base-v2; from now on, we refer to this model as PMMBV2. PMMBV2 is a multilingual SBERT-variant appropriate for sentence representations, making it ideal for encoding questions in Turkish. It outperforms BERT, XLNet, and RoBERTa on the GLUE Benchmark and Question Answering (SQuAD) datasets.

PMMBV2 is trained using multilingual knowledge distillation [32]. Two distinct models are chosen in the training phase: teacher and student. The teacher produces sentence embeddings in one specific language. The student model is expected to imitate the teacher model. Training is based on parallel sentences with corresponding translations to ensure that the student model can handle multiple languages. In the training phase, paraphrase-mpnet-base-v2 [33] is used as the teacher model to generate embeddings for English sentences, and XLM-RoBERTa [34] serves as the student model to generate embeddings for English and other languages such as Turkish.

The PMMBV2 model generates contextualized word embeddings, which are then compressed into a 768-dimensional vector representation through mean pooling. The resulting vector is processed through three fully connected linear layers with the ReLU activation function.

4.3. Answer selection module

This module approaches the question answering as a link prediction problem. The answer selection module takes the question and entity embeddings as input. It treats the embedding representing the question as a relation embedding. On the other hand, the entity embedding serves as the source node. The module aims to accurately predict the target node incorporating the source and relation embeddings. During inference, the model evaluates the combination of the source node and the question against all possible entities in BPMovieKG. The answer is the entity with the highest likelihood, predicted by the link prediction models used in the architecture, such as DistMult, ComplEx, and SimplE.

5. Experimental results

In this section, we present the experimental results in three parts. Firstly, we compare the existing knowledge bases in [35] against BPMovieKG through graph embedding methods and highlight the peculiarities of BP-MovieKG. Secondly, we report the performance of the question-answering system on 1, 2, and 3 hop questions. Lastly, we ask the 1-hop questions from TRMQA to OpenAI GPT-3.5 Turbo and comment on its relative performance.

5.1. Metrics

We use Hits@1, Hits@3, Hits@10, and Mean Reciprocal Rank (MRR) to compare graph embedding methods, while we rely on Hits@1 to report the performance of question answering. **Hits@K** is a measure of model effectiveness that gives the ratio of predictions with a ranking equal to or below a threshold K. **Mean Reciprocal Rank** is the average of the reciprocals of the ranks of the ground-truth triples. The higher Hits@K and MRR are better.

5.2. Experimental setup

The hyperparameters for the graph embedding methods are the same as those used in the baseline study and are shown in Table 4A. Table 4B shows the hyperparameters and their values for the question answering system.

The BiLSTM embedding size is 256, and all the hyperparameters are the same as those in the baseline study. Additionally, during training, early stopping is applied. We monitor the progress of the training process by evaluating the Hits@1 metric on the validation set every 5 epoch. If the result does not surpass the best result obtained within the previous 5 evaluations, we terminate the training process and select the best model.

Table 4. Model hyperparameters.

Table 4A. Graph embedding model hyperparameters.

Hyperparameter	Value
Learning rate	0.0005
Batch size	128
Epoch	200
Entity-Relation Vector Size	200

Table 4B. Question answering model hyperparameters.

Hyperparameter	Value
Learning rate	0.0001
Batch size	256
Epoch	90
Validate Every	5
Patience	5
BiLSTM Dimension	256

5.3. Graph embedding methods experiments

We run three graph embedding methods on the WN18, FB15k, and BPMovieKG datasets and report the results in Table 5A. We used WN18 and FB15k as benchmarks for KG embeddings since there is no such benchmark in Turkish. Also, conceptual structuring is essential in structured data sources, and language dependency is not a big concern. Table 5B gives the summary statistics for these datasets.

Table 5. Performance results and statistics of various knowledge bases.

Table 5A. Performance comparison of ComplEx, DistMult, and SimplE on the WN18, FB15k and BPMovieKG datasets.

Dataset	WN18				FB15k				BPMovieKG			
Model	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR
ComplEx	0.440	0.631	0.790	0.559	0.183	0.330	0.545	0.296	0.332	0.473	0.582	0.420
DistMult	0.293	0.481	0.703	0.423	0.165	0.291	0.476	0.264	0.239	0.370	0.502	0.326
SimplE	0.440	0.631	0.794	0.559	0.181	0.326	0.538	0.293	0.333	0.474	0.585	0.421

Table 5B. Knowledge base statistics.

Knowledge Base	Entities	Relations	Training	Validation	Test
WN18	40,943	18	141,442	5000	5000
FB15k	14,951	1345	483,142	50,000	59,071
BpMovieKG	36,489	16	288,992	14,500	14,500

As can be seen from the results, the SimplE method provides the best results on BPMovieKG in all evaluation metrics. Consistent with the prior studies in the literature, the DistMult method has yielded the worst scores for all three datasets.

Furthermore, the ComplEx method is superior to SimplE on the FB15k dataset. In contrast, in the WN18 dataset, only in the Hits@10 metric, the SimplE method outperforms ComplEx, and both methods yield similar results in other metrics. As the number of relations within the datasets increases, the ComplEx method gives better results than the SimplE method.

Additionally, the worst-performing dataset for the methods is FB15k. Although FB15k has a significantly larger training set size, it also has 80 times more relations than others. This exponentially higher number of relations explains the poorer performance in the link prediction task.

The distribution of entity degrees for each dataset is shown in Appendices 6, 6, and 6. As observed from the distributions, the entity degrees in the FB15k dataset, which is the least generalizable, exhibit a wide range of values, including very high entity degrees compared to the other datasets. On the other hand, the entity degrees in the WN18 dataset, which is the most generalizable dataset, are spread over a narrower range with lower entity degrees. BPMovieKG has a narrow range of entity degrees with a more dispersed pattern than WN18. Based on the datasets and methods used in this study, one can conclude that link prediction performance lessens as the range of entity degrees widens and their values increase.

5.4. Question answering system experiments

5.4.1. Question answering results

We present the performance of the proposed method (graph embedding method-SBERT pairs) and the baseline architecture (graph embedding method-BiLSTM pairs) on TRMQA 1-2-3 questions in Table 6. We ran each method five times and averaged the results.

Table 6. Validation and testing performances (hits@1 %) on TRMQA involving 1-2-3 hop questions.

Dataset	Validation			Test		
Method	1-hop	2-hop	3-hop	1-hop	2-hop	3-hop
ComplEx-BiLSTM	**100%**	**97.30%**	73.36%	**100%**	**97.32%**	74.30%
SimplE-BiLSTM	**100%**	95.83%	73.97%	**100%**	96.05%	75.02%
DistMult-BiLSTM	**100%**	95.11%	73.17%	**100%**	95.37%	74.01%
ComplEx-SBERT	**100%**	97.21%	73.36%	**100%**	97.30%	74.30%
SimplE-SBERT	**100%**	95.82%	**74.13%**	**100%**	95.90%	**75.61%**
DistMult-SBERT	**100%**	94.02%	73.49%	**100%**	94.11%	74.05%

The results show that all methods can correctly answer all 1-hop questions in both partitions. Recent studies [36, 37] on the MetaQA dataset have demonstrated that they perform better on 2- and 3-hop questions than on 1-hop questions due to the entity disambiguation requirement in MetaQA. However, this issue did not arise in the BPMovieKG dataset, as each entity name is unique.

The baseline method produces superior results for 2-hop questions in both the validation and test sets, while SimplE+SBERT gives the best results for 3-hop questions. There is no statistically significant difference due to question embeddings (BiLSTM and SBERT).

Since the context of 3-hop questions is broader and the questions are longer and more detailed, models that can capture longer sequences tend to perform better. Since the number of question types in the 2-hop dataset is higher than the number of question types in the 3-hop dataset, and as stated in the comparison section of Graph Embedding methods, the higher performance of the ComplEx method in the 2-hop dataset could be related to the number of question types. On the other hand, in the 3-hop dataset with a lower number of question types, the SimplE method provides better results.

As observed in the results of embedding methods, DistMult performs worse in knowledge completion on BPMovieKG. The question answering system's performance confirms a similar behavior, where DistMult consistently yields the lowest scores in both 2-hop and 3-hop evaluations.

5.4.2. Evaluation of 1-Hop questions in TRMQA using GPT and LlamaIndex

In this subsection, we present OpenAI's GPT-3.5 Turbo model's performance in answering 1-hop questions from TRMQA to have a large language model baseline. The model is fine-tuned on GPT-3.5, a subclass of GPT-3.

Using the GPT-3.5 Turbo model, we have developed a question-answering system for the evaluation using the Hits@K metric. To achieve this, we first create a text document containing the names of all the nodes in BPMovieKG. After that, we partition it into text segments, each containing 4096 tokens. Using the LlamaIndex, we generate embeddings for each text segment using the GPT-3.5 Turbo model. The purpose of dividing the BPMovieKG graph into chunks of 4096 and then converting them into vectors is that GPT-3.5 Turbo can take a maximum of 4096 tokens as input. We concatenate all the embeddings of text segments to get a single embedding. This single embedding represents the overall context of the BPMovieKG. This way, all questions that can be asked through BPMovieKG can be answered.

To ask questions to GPT-3.5 Turbo, we create a prompt template. Within this template, we want a single entity name as the answer and provide an example to guide the model's response. Then, we keep the answer field empty for the model to complete when asking the question. Figure 3 shows an example prompt.

```
Prompt: Aşağıdaki soruya en yüksek olasılığa sahip tek bir varlık ismi ile cevap verebilir
misin. Verilen cevap içerisinde sadece varlık ismi yer alsın.
   Örnek Soru: Karanlıklar Ülkesi: Kan Savaşları filmi hangi yılda yayınlanmıştır?
   Örnek Cevap: 2016
Soru: Zor Baba 3 filmi hangi yılda yayınlanmıştır?
Cevap:
```

Figure 3. An example of a prompt asking for the release year of a movie.

As a result, we concatenate the embedding obtained from the prompt and the embedding to represent the BPMovieKG context and give it to GPT-3.5 Turbo to generate an answer. Thus, we receive a single entity name as an answer, which matches with one of the nodes within BPMovieKG, providing the answer in the Hits@1 metric.

We exclude the questions containing the relation "beyazperde_yildizi" in the evaluation as this information is specific to that website and not general knowledge.

Quantitative analysis In this section, we begin by quantitatively evaluating our dataset. We evaluate the answers provided by GPT-3.5 Turbo and compare them to our method using the Hits@1 metric.

In open-domain question answering systems such as GPT-3.5 Turbo, if the system does not have the necessary information to disambiguate a named entity for a particular question, even if it provides the correct answer for another entity with the same name, the answer may be incorrect for the entity we are asking about.

Thus, the entities without the disambiguation issue have been collected under the "Filtered" dataset, whereas the "All" version includes entities with and without disambiguation. Table 7 gives the two types of evaluations on the validation and test set partitions.

Table 7. Evaluation results of the 1-hop questions based on Hits@1.

Dataset	Validation		Test	
Relation type	All	Filtered	All	Filtered
mesleklerinden_birisidir	78.97%	93.29%	79.54%	93.55%
uyruklarindan_birisidir	83.36%	96.06%	81.81%	97.16%
yayinlanma_yili	54.61%	83.76%	53.10%	84.25%
dillerinden_birisidir	72.72%	91.93%	73.88%	92.57%
turlerinden_birisidir	43.04%	83.70%	40.47%	83.51%
oyuncularindan_birisidir	39.18%	72.57%	40.81%	76.50%
yonetmenlerinden_birisidir	48.14%	83.00%	46.00%	80.06%

To filter the dataset, we determined the frequency of movie or person entities within the movie domain for each question using the Cinemagoer[6] library. If the entity appeared only once, we used it in the evaluation process.

As seen from Table 7, the Filtered set results better reflect the actual performance of GPT-3.5 Turbo as there is no entity-related ambiguity.

In question types where the answer is a person ('oyuncularindan_birisidir' and 'yonetmenlerinden_birisidir'), GPT-3.5 Turbo has lower performance compared to other question types. Since 69% of the nodes in the knowledge graph are persons, the model struggles to correctly identify the right person from a more extensive search space.

Additionally, GPT-3.5 Turbo performs better in answering questions related to personal information, such as questions type 'mesleklerinden_birisidir' and 'uyruklarindan_birisidir' compared to questions about movie information. One can explain this performance difference because the training corpora for GPT-3.5 Turbo contain more information about persons than those related to movies.

To conclude, our system performs better than the fine-tuned GPT-3.5 Turbo model as it answers all 1-hop questions correctly.

Qualitative analysis The following qualitative analysis includes an error analysis of the performance of GPT-3.5 Turbo.

Question-1: Hangi yıl Trendeki Kız filmi yayınlanmıştır?
Ground-Truth: 2016
GPT-3.5 Turbo Answer: 2021
Discussion: There are multiple films titled 'Trendeki Kız.' One of these films was released in 2021, while another in 2016. GPT-3.5 Turbo provides an answer for the most recent film. However, the film present in BPMovieKG is the other one. The wrong answer is due to name ambiguity.

Question-2: Zehirli Element filmindeki aktörlerden biri kimdir?
Ground-Truth: Paddy Considine

[6]https://github.com/cinemagoer

GPT-3.5 Turbo Answer: Iron Man
Discussion: This question expects an actor from the film 'Zehirli Element.' In response, the GPT-3.5 Turbo model provides a film name. Here, the model struggles to understand the type of answer the question requires.

Question-3: Gilbert'in Hayalleri filminde hangi tema işlenmiştir?
Ground-Truth: Dramatik komedi
GPT-3.5 Turbo Answer: Dram
Discussion: This example asks for one of the genres of 'Gilbert'in Hayalleri.' Although the GPT-3.5 Turbo model could identify the correct genre, 'Dram,' it failed to find the more specific genre, 'Dramatik Komedi,' as the answer.

Question-4: Hangi yıl Yarın Asla Ölmez filmi vizyona girmiştir?
Ground-Truth: 1997
GPT-3.5 Turbo Answer: 1987
Discussion: This example asks about the release year of 'Yarın Asla Ölmez,' one of the James Bond films. GPT-3.5 Turbo returns the release year of another Bond film, 'Gün Işığında Suikast.' This error indicates that the model has difficulty in distinguishing sequel films.

6. Conclusion and future work

In this work, we developed a Turkish question answering system enriched with a knowledge graph. We constructed a knowledge graph (BPMovieKG) by crawling beyazperde.com to answer questions in the movie domain. Then, we created a collection of 1-2-3 hop questions (TRMQA) from BPMovieKG. As part of our experiments, we compared BPMovieKG with the existing knowledge bases using different graph embedding methods. The results confirmed that graph embedding methods performed better in knowledge bases with lower entity degree values and within lower ranges, as many relations complicate link prediction.

We evaluated our proposed question answering pipeline against a baseline study. Our finding is that the graph embedding method used in the question answering system affects the results. Furthermore, the question embedding module does not significantly impact the performance. While BiLSTM provides a better representation for 2-hop questions, the SBERT model was superior for 3-hop questions.

Additionally, we tested a question answering system built upon GPT-3.5 Turbo to answer the 1-hop questions from TRMQA. Our experiments showed that the GPT-3.5 Turbo model is more prone to making errors than the KG-enriched pipeline. This behavior may be due to insufficient information about the film domain in the corpora used to fine-tune the GPT-3.5 Turbo model. Another reason is that the GPT-3.5 Turbo model cannot distinguish the hierarchical relationships among named entities.

With today's large language models, question answering, extractive in nature, has been transformed into a generation task, supported with appropriate prompts. New approaches in this area should reconcile extraction and generation in their pipelines. Retrieval augmented generation (RAG) is a new theme where a context extraction phase precedes the generation of the answer based on a prompt extended with the retrieved context. Thus, knowledge-supported prompting solutions are a plausible research direction.

As a future work, the analyses performed in this study can be further developed to create reliable question-answering systems based on large language models that exploit structured knowledge in knowledge graphs. Such systems can prevent hallucinations and allow deeper reasoning paths.

Acknowledgment

We thank Serap Şahin for participating in our meetings during the earlier phases of this project. We also thank anonymous reviewers for their valuable comments.

References

[1] Pala Er N, Çiçekli I. A factoid question answering system using answer pattern matching. In: Sixth International Joint Conference On Natural Language Processing (IJCNLP 2013), Asian Federation of Natural Language Processing (ACL 2013); Nagoya, Japan; October 14-18, 2013; 854-858.

[2] Amasyali M, Biricik G, Solmaz S, Özdemir E. A Turkish automatic question answering system with question multiplexing: Ben bilirim. International Journal of Research in Information Technology (IJRIT) 2013; 1 (6): 46-51.

[3] Wang B, Shen T, Long G, Zhou T, Wang Y et al. Structure-augmented text representation learning for efficient knowledge graph completion. In: Proceedings of the Web Conference 2021 (WWW'21); April 19-23, 2021; 1737-1748. https://doi.org/10.1145/3442381.3450043

[4] Li D, Zhu B, Yang S, Xu K, Yi M et al. Multi-task pre-training language model for semantic network completion. ACM Transactions on Asian and Low-Resource Language Information Processing 2023; 22 (11): 1-20. https://doi.org/10.1145/3627704

[5] Bi Z, Cheng S, Chen J, Liang X, Xiong F et al. Relphormer: relational graph transformer for knowledge graph representation. Neurocomputing, 566, 127044, 2024. https://doi.org/10.1016/j.neucom.2023.127044

[6] Brown T, Mann B, Ryder N, Subbiah M, Kaplan J et al. Language models are few-shot learners. In: 33rd Advances In Neural Information Processing Systems (NeurIPS 2020); Virtual; December 6-12, 2020; 1877-1901.

[7] Derici C, Celik K, Kutbay E, Aydin Y, Gungor T et al. Question analysis for a closed domain question answering system. In Computational Linguistics and Intelligent Text Processing: 16th International Conference, CICLing 2015, Cairo, Egypt, April 14-20, 2015, Proceedings, (pp. 468-482). Springer International Publishing. https://doi.org/10.1007/978-3-319-18117-2_35

[8] Çelebi E, Günel B, Şen B. Automatic question answering for Turkish with pattern parsing. In: International Symposium On Innovations In Intelligent Systems And Applications 2011; 389-393. https://doi.org/10.1109/INISTA.2011.5946098

[9] Derici C, Aydin Y, Yenialaca Ç, Aydin NY, Kartal G et al. A closed-domain question answering framework using reliable resources to assist students. Nat Lang Eng 2018; 24: 725-762. https://doi.org/10.1017/S1351324918000141

[10] Tasar CO, Komesli M, Unalir MO. Semantic web enabled geographic question answering framework: GeoTR, 2023. https://doi.org/10.48550/arXiv.2301.04752

[11] Yigit G, Amasyali MF. Ask me: A question answering system via dynamic memory networks. In: Innovations In Intelligent Systems And Applications Conference (ASYU); 2019;1-5. https://doi.org/10.1109/ASYU48272.2019.8946411

[12] Weston J, Bordes A, Chopra S, Rush AM, Van Merriënboer B et al. Towards ai-complete question answering: A set of prerequisite toy tasks. In 4th International Conference on Learning Representations, ICLR 2016.

[13] Soygazi F, Çiftçi O, Kök U, Cengiz S. THQuAD: Turkish Historic Question Answering Dataset for Reading Comprehension. In: 2021 6th International Conference on Computer Science and Engineering (UBMK); 2021; 215-220. https://doi.org/10.1109/UBMK52708.2021.9559013

[14] Menevşe MU, Manav Y, Arisoy E, Özgür A. A Framework for Automatic Generation of Spoken Question-Answering Data. In: Findings of the Association for Computational Linguistics: EMNLP 2022;4659-4666. https://doi.org/10.18653/v1/2022.findings-emnlp.342

[15] Gemirter C, Goularas D. A Turkish question answering system based on deep learning neural networks. Journal Of Intelligent Systems: Theory And Applications 2021; 4 (2): 65-75. https://doi.org/10.38016/jista.815823

[16] Akyön F, Çavusoglu A, Cengiz C, Altinuc S, Temizel A. Automated question generation and question answering from Turkish texts. Turkish Journal of Electrical Engineering and Computer Science 2022; 30: 1931-1940. https://doi.org/10.55730/1300-0632.3914

[17] Bordes A, Usunier N, Chopra S, Weston J. Large-scale simple question answering with memory networks. CoRR 2015. https://doi.org/10.48550/arXiv.1506.02075

[18] Cui W, Xiao Y, Wang H, Song Y, Hwang S et al. KBQA: learning question answering over QA corpora and knowledge bases. In: Proceedings of the VLDB Endowment 2017; 10 (5):565-576. https://doi.org/10.14778/3055540.3055549

[19] Saxena A, Tripathi A, Talukdar P. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In: Proceedings Of The 58th Annual Meeting Of The Association For Computational Linguistics, ACL 2020; Online; July 5-10, 2020; 4498-4507. https://doi.org/10.18653/v1/2020.acl-main.412

[20] Sorokin D, Gurevych I. Modeling semantics with gated graph neural networks for knowledge base question answering. In: Proceedings of the 27th International Conference on Computational Linguistics; 2018. pp. 3306-3317.

[21] De Cao N, Aziz W, Titov I. Question answering by reasoning across documents with graph convolutional networks. 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; 2306-2317. https://doi.org/10.18653/v1/N19-1240

[22] Yan Y, Li R, Wang S, Zhang H, Daoguang Z et al. Large-scale relation learning for question answering over knowledge bases with pre-trained language models. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing 2021; 3653-3660. https://doi.org/10.18653/v1/2021.emnlp-main.296

[23] Ravishankar S, Thai D, Abdelaziz I, Mihidukulasooriya N, Naseem T et al. A two-stage approach towards generalization in knowledge base question answering. In: Findings of the Association for Computational Linguistics: EMNLP 2022; 5571-5580. https://doi.org/10.18653/v1/2022.findings-emnlp.408

[24] Sun H, Bedrax-Weiss T, Cohen W. PullNet: open domain question answering with iterative retrieval on knowledge bases and text. In: Proceedings Of The 2019 Conference On Empirical Methods In Natural Language Processing And The 9th International Joint Conference On Natural Language Processing, EMNLP-IJCNLP 2019; Hong Kong, China; November 3-7, 2019; 2380-2390. https://doi.org/10.18653/v1/D19-1242

[25] Sun H, Dhingra B, Zaheer M, Mazaitis K, Salakhutdinov R et al. Open domain question answering using early fusion of knowledge bases and text. In: Proceedings Of The 2018 Conference On Empirical Methods In Natural Language Processing; Brussels, Belgium; October 31 - November 4, 2018; 4231-4242. https://doi.org/10.18653/v1/D18-1455

[26] Zhang Y, Dai H, Kozareva Z, Smola A, Song L. Variational reasoning for question answering with knowledge graph. In: Proceedings Of The Thirty-Second AAAI Conference On Artificial Intelligence, (AAAI-18), The 30th Innovative Applications Of Artificial Intelligence (IAAI-18), And The 8th AAAI Symposium On Educational Advances In Artificial Intelligence (EAAI-18); New Orleans, Louisiana, USA; February 2-7, 2018; 6069-6076.

[27] Miller A, Fisch A, Dodge J, Karimi AH, Bordes A et al. Key-value memory networks for directly reading documents. In: Proceedings Of the 2016 Conference On Empirical Methods In Natural Language Processing, EMNLP 2016; Austin, Texas, USA; November 1-4, 2016. pp. 1400-1409. https://doi.org/10.18653/v1/D16-1147

[28] Yang B, Yih W, He X, Gao J, Deng L. Embedding entities and relations for learning and inference in knowledge bases. In: Conference Track Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015; San Diego, CA, USA; May 7-9, 2015.

[29] Trouillon T, Welbl J, Riedel S, Gaussier É, Bouchard G. Complex embeddings for simple link prediction. In: International Conference On Machine Learning 48; 2016. pp. 2071-2080.

[30] Kazemi SM, Poole D. Simple embedding for link prediction in knowledge graphs. In: 31st Advances In Neural Information Processing Systems (NeurIPS 2018); Montréal, Canada; December 3-8, 2018; 4289-4300.

[31] Reimers N, Gurevych I. Sentence-BERT: sentence embeddings using siamese BERT-networks. In: Proceedings Of The 2019 Conference On Empirical Methods In Natural Language Processing And The 9th International Joint Conference On Natural Language Processing, EMNLP-IJCNLP 2019; Hong Kong, China; November 3-7, 2019. pp. 3982-3992.

[32] Reimers N, Gurevych I. Making monolingual sentence embeddings multilingual using knowledge distillation. In: Proceedings Of The 2020 Conference On Empirical Methods In Natural Language Processing, EMNLP 2020; Online; November 16-20, 2020. pp. 4512-4525. https://doi.org/10.18653/v1/2020.emnlp-main.365

[33] Song K, Tan X, Qin T, Lu J, Liu T. MPNet: Masked and permuted pre-training for language understanding. In: 33rd Advances In Neural Information Processing Systems 33: Annual Conference On Neural Information Processing Systems 2020, NeurIPS 2020; December 6-12, 2020; Virtual. pp.16857-16867.

[34] Conneau A, Khandelwal K, Goyal N, Chaudhary V, Wenzek G et al. Unsupervised cross-lingual representation learning at scale. In: Proceedings Of The 58th Annual Meeting Of The Association For Computational Linguistics (ACL 2020); Online; July 5-10, 2020; 8440-8451. https://doi.org/10.18653/v1/2020.acl-main.747

[35] Bordes A, Usunier N, Garcia-Duran A, Weston J, Yakhnenko O. Translating embeddings for modeling multi-relational data. In: 26th Advances In Neural Information Processing Systems 2; Lake Tahoe, Nevada, United States; 2013. pp. 2787-2795.

[36] He G, Lan Y, Jiang J, Zhao WX, Wen J-R. Improving multi-hop knowledge base question answering by learning intermediate supervision signals. In: The Fourteenth ACM International Conference On Web Search And Data Mining (WSDM '21); Virtual Event; Israel; March 8-12, 2021. pp. 553-561. https://doi.org/10.1145/3437963.3441753

[37] Shi J, Cao S, Hou L, Li J, Zhang H. TransferNet: an effective and transparent framework for multi-hop question answering over relation graph. In: Proceedings Of The 2021 Conference On Empirical Methods In Natural Language Processing, EMNLP 2021; Virtual Event; Punta Cana, Dominican Republic, November 7-11, 2021. pp. 4149-4158. https://doi.org/10.18653/v1/2021.emnlp-main.341

Appendices

Appendix A. Subgraph of BPMovieKG.

Appendix B. Examples of 1-hop, 2-hop, and 3-hop questions.

Table S1. Examples of the 8 types of 1-hop questions.

Question type	Count	Example
Film - Yayınlanma Yılı	10,947	Hangi sene İhtiyarlara Yer Yok filmi yayınlandı?
Film - Dil	13,240	Hangi dil Çifte Hayatlar filminin seslendirilmesinde kullanılmıştır?
Aktör - Meslek	36,750	Jennifer Ulrich kişisi meslek olarak ne yapmaktadır?
Film - Aktör	40,238	Hicran Gecesi filminde rol alan aktörlerden biri kimdir?
Film - Tür	19,326	Straight Outta Compton filminin kategorisi nedir?
Film - Uyruk	19,184	Hangi uyruk Zoë Wanamaker kişisinin uyruğudur?
Film - Yönetmen	11,620	Hangi kişi Bıçağın İki Yüzü filminde yönetmendir?
Film - Yıldız Değeri	2510	Kevin Hakkında Konuşmalıyız filminin aldığı yıldız değeri kaçtır?

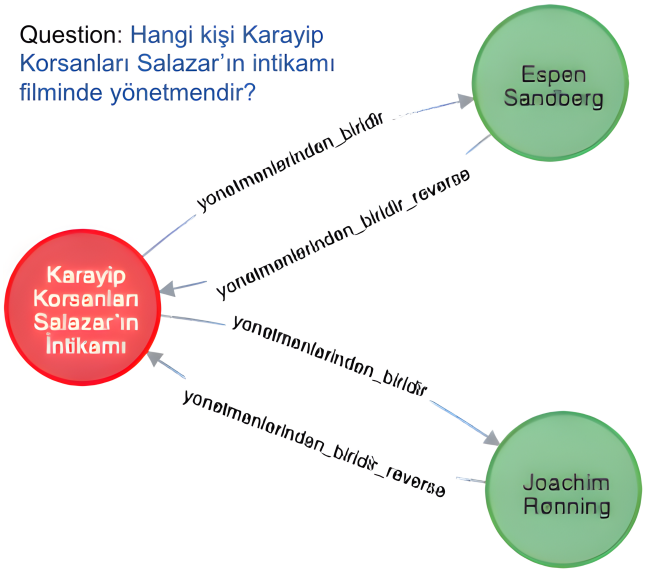
Table S2. Examples of the 19 types of 2-hop questions.

Question type	Count	Example
Film - Yayınlanma Yılı - Film	10,947	Yarının Sınırında filminin yayınlanma yılında hangi filmler vizyona girmiştir?
Film - Dil - Film	4377	13. Savaşçı filmi içerisinde konuşulan lisanlardan biri aynı zamanda hangi filmlerde kullanılmaktadır?
Film - Tür - Film	4382	Hangi filmlerin temaları The Company Men filminin temaları ile kesişmektedir?
Film - Yönetmen - Film	7324	Constantine filminin yönetmeninin yönettiği diğer filmler nelerdir?
Film - Aktör - Film	10,012	Fury filminde yer alan oyuncular aynı zamanda hangi filmde rol almıştır?
Film - Aktör - Uyruk	10,688	Parabellum filmindeki oyuncular hangi uyruktan gelmektedirler?
Film - Aktör - Meslek	10,957	Teksas Katliamı: Başlangıç filmindeki aktörler hangi meslek ile uğraşmaktadır?
Film - Yönetmen - Meslek	10,947	Ocean's 12 filminin yönetmenlerinin meslekleri nelerdir?
Film - Yönetmen - Uyruk	9470	Yeşil Sokak Holiganları filminin yönetmenleri hangi milliyettendir?
Aktör - Film - Yönetmen	19,632	Hangi yönetmenler Sari Lennick'in oynadığı filmleri yönetmiştir?
Aktör - Film - Tür	19,788	Mel Gibson'in oynadığı filmler hangi temalardadır?
Aktör - Film - Yayınlanma Yılı	19,758	Ethan Hawke'ın yer aldığı filmlerin yayınlanma yılı nedir?
Aktör - Film - Dil	19,737	Daniel Brühl'in oynadığı filmlerde kullanılan diller nelerdir?
Aktör - Film - Aktör	19,792	Hangi aktörler ile Nick Cave aynı filmde yer almıştır?
Yönetmen - Film - Aktör	6095	Bruce Lee hangi oyuncuların oynadığı filmleri yönetmiştir?
Yönetmen - Film - Yönetmen	977	Hakan Gürtop'in yönettiği filmlerde başka hangi yönetmenler bulunmaktadır?
Yönetmen - Film - Yayınlanma Yılı	6089	Hangi senelerde Katharine O'Brien'in yönettiği filmler yayınlanmıştır?
Yönetmen - Film - Tür	6094	Stuart Beattie hangi kategorilere ait filmleri yönetmiştir?
Yönetmen - Film - Dil	6082	Hangi diller Peter Weir'in çektiği filmlerde kullanılmıştır?

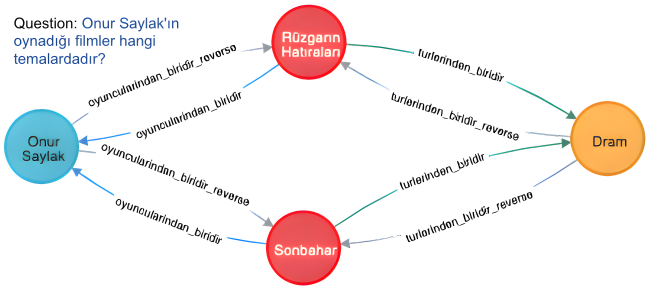
Table S3. Examples of the 14 types of 3-hop questions.

Question type	Count	Example
Aktör - Film - Aktör - Meslek	19,788	Hangi iş üzerine Amanda Peet oyuncusu yer aldığı filmlerde bulunan aktörler uğraşmaktadır?
Aktör - Film - Aktör - Uyruk	18,558	Hangi milletler Arielle Holmes oyuncusunun yer aldığı filmlerdeki oyuncuların milletleridir?
Yönetmen - Film - Yönetmen - Meslek	977	Hangi meslek Greg Strause yönetmeni ile aynı filmlerde yönetmen olan kişilerce yapılmaktadır?
Yönetmen - Film - Yönetmen - Uyruk	731	Zeynel Doğan yönetmeninin yönettiği filmlerin yönetmenleri hangi uyruktan gelmektedir?
Film - Aktör - Film - Yayınlanma Yılı	10,010	A Million Little Pieces filmi içerisinde yer alan oyuncuların diğer oynadığı filmlerin yayınlanma yılı ne zamandır?
Film - Aktör - Film - Dil	10,012	Mr. Bean Tatilde filminde rol alan kişilerin oynadığı filmlerde hangi dil konuşulmaktadır?
Film - Aktör - Film - Tür	10,012	Ip Man filminin oyuncularının oynadığı filmlerin içerisinde hangi kategoriler bulunmaktadır?
Film - Aktör - Film - Yönetmen	10,012	Hangi yönetmenler Yılanların Öcü filmi oyuncularının oynadığı filmleri yönetmektedir?
Film - Aktör - Film - Yıldız Değeri	7795	Hangi değerlendirme puanı Rocky 4 filmi oyuncularının oynadığı filmler için verilmiştir?
Film - Yönetmen - Film - Yayınlanma Yılı	7322	Hangi yıllar içerisinde Oslo, 31 Ağustos filminin yönetmeninin yönettiği filmler yayınlanmaktadır?
Film - Yönetmen - Film - Dil	7322	Hangi dil Zor Ölüm 2 filmini yöneten kişinin yönettiği filmlerde kullanılmıştır?
Film - Yönetmen - Film - Tür	7324	Siyah Giyen Adamlar 3 filminin yönetmeninin diğer yönettiği filmlerde hangi konular yer almaktadır?
Film - Yönetmen - Film - Aktör	7324	Hangi oyuncu Pearl Harbor filmi içerisinde yönetmen olarak rol alan kişinin filmlerinde oynamıştır?
Film - Yönetmen - Film - Yıldız Değeri	3858	Pasifik Savaşı filmi içerisinde yer alan yönetmenlerin diğer yönettiği filmlerin değerlendirmesi 5 üzerinden kaçtır?

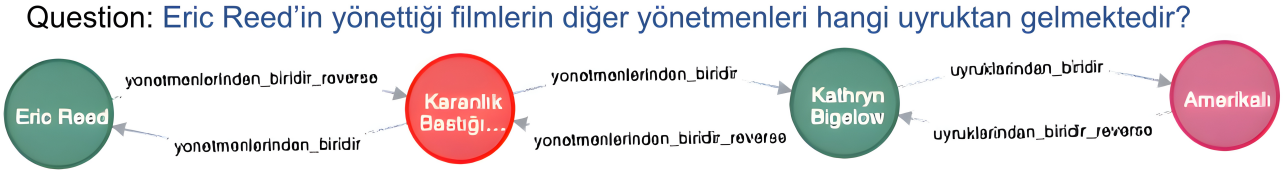
Appendix C. 1-hop reasoning example for "Movie to Director" question type.



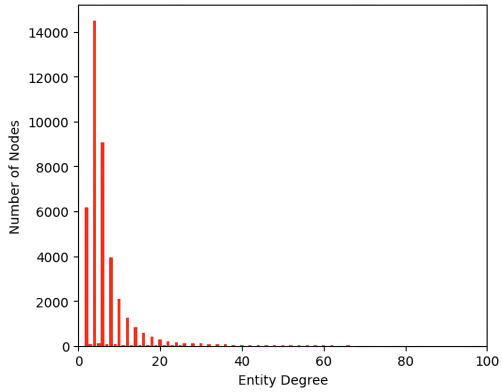
Appendix D. 2-hop reasoning example for "Actor to Movie to Genre" question type.



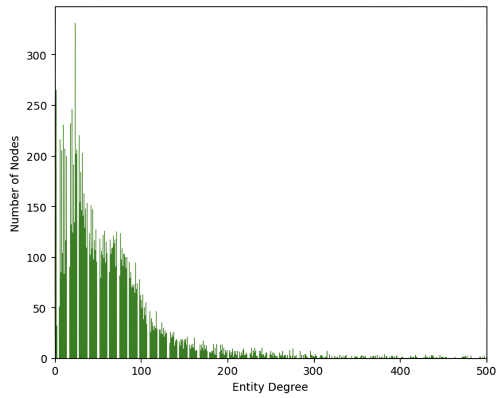
Appendix E. 3-hop reasoning example for "Director to Movie to Director to Nationality" question type.



Appendix F. Distribution of entity degrees in WN18.



Appendix G. Distribution of entity degrees in FB15K.



Appendix H. Distribution of entity degrees in BPMovieKG.