ENRICHMENT OF TURKISH QUESTION ANSWERING SYSTEMS USING KNOWLEDGE GRAPHS

A Thesis Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

in Computer Engineering

by
Okan ÇİFTÇİ

July 2023
İZMİR

We approve the thesis of **Okan ÇİFTÇİ**

Examining Committee Members:

Prof. Yalın BAŞTANLAR

Department of Computer Engineering, Izmir Institute of Technology

Assoc. Prof. Selma TEKİR

Department of Computer Engineering, Izmir Institute of Technology

Asst. Prof. Alper DEMİR

Department of Computer Engineering, Izmir Economy University

19 July 2023

Assoc. Prof. Selma TEKİR

Supervisor, Department of Computer Engineering

Izmir Institute of Technology

Asst. Prof. Fatih SOYGAZİ

Co-Supervisor, Department of Computer Engineering

Aydın Adnan Menderes University

Prof. Cüneyt F. BAZLAMAÇCI

Head of the Department of Computer Engineering

Prof. Dr. Mehtap EANES

Dean of the Graduate School of Engineering and Sciences

ACKNOWLEDGMENTS

First and foremost, I express my sincere gratitude to my thesis advisor, Associate Professor Dr. Selma Tekir, for her invaluable support, guidance, and patience throughout my master's education.

I would also like to extend my sincerest appreciation to Assistant Professors Dr. Serap Şahin and Dr. Fatih Soygazi for their insightful contributions and invaluable guidance.

ABSTRACT

ENRICHMENT OF TURKISH QUESTION ANSWERING SYSTEMS USING KNOWLEDGE GRAPHS

In the era of digital communication, the ability to effectively process and interpret human language has become a key research area. Natural Language Processing (NLP) has emerged as a field that enables machines to better understand and analyze human language. One of the most important applications of NLP is the development of question answering systems, which are essential in various domains such as customer service, search engines, and chatbots. To answer incoming queries, question answering systems rely on knowledge graphs as a reliable source.

This thesis proposes a Turkish Question Answering (TRQA) system that utilizes a knowledge graph. The research focuses on the automatic construction of a knowledge graph specific to the film industry, as well as the creation of a multi-hop question-answering dataset that can be queried from this graph. Building upon these constructions, we develop a deep learning based method for answering questions using the constructed knowledge graph.

The constructed knowledge graph is compared with various knowledge graphs presented in the literature using DistMult, ComplEx and SimplE methods for the link prediction task. Additionally, the proposed question answering system is compared with the baseline study and compared with a generative large language model through quantitative and qualitative analyses.

ÖZET

TÜRKÇE SORU CEVAPLAMA SİSTEMLERİNİN BİLGİ ÇİZGELERİ İLE ZENGİNLEŞTİRİLMESİ

Dijital iletişim çağında, insan dilini etkili bir şekilde işleme ve yorumlama yeteneği önemli bir araştırma alanı haline gelmiştir. Doğal Dil İşleme, makinelerin insan dilini daha iyi anlamalarını ve analiz etmelerini sağlayan bir alan olarak ortaya çıkmıştır. Doğal Dil İşleme'nin en önemli uygulamalarından biri, müşteri hizmetleri, arama motorları ve sohbet botları gibi çeşitli alanlarda önemli olan soru cevap sistemlerinin geliştirilmesidir. Gelen sorgulara cevap verebilmek için soru cevaplama sistemleri, güvenilir bir kaynak olarak bilgi çizgelerinden yararlanır.

Bu tez, bilgi çizgesi kullanan bir Türkçe soru-cevap sistemini önermektedir. Çalışma, film endüstrisi ile ilgili bir bilgi çizgesinin otomatik olarak oluşturulmasına odaklanmakta, ayrıca bu bilgi çizgesi üzerinden sorgulanabilen çoklu adımlı bir soru-cevap veri kümesi oluşturulmasını kapsamaktadır. Bu iki oluşuma dayanarak, oluşturulan bilgi çizgesi kullanılarak derin öğrenme tabanlı bir soru-cevap yöntemi geliştirilmiştir.

Oluşturulan bilgi çizgesi, bağlantı tahmini görevi için DistMult, ComplEx ve SimplE yöntemlerini kullanarak literatürde yer alan çeşitli bilgi çizgeleri ile karşılaştırılmıştır. Ek olarak, önerilen cevap sistemi, hem referans alınan çalışma ile karşılaştırılmış olup, hem de üretken büyük dil modeli ile nicel ve nitel analizler yoluyla karşılaştırılmıştır.

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

Table **Page**

LIST OF ABBREVIATIONS AND SYMBOLS

NLP Natural Language Processing

QA Question Answering

LLM Large Language Model

KG Knowledge Graph

TRQA Turkish Question Answering

KGQA Knowledge Graph based Question Answering

BPMOVIEKG Beyazperde Movie Knowledge Graph

TRMQA Turkish Movie Question Answering Dataset

RNN Recurrent Neural Network

LSTM Long Short-Term Memory

BiLSTM Bidirectional Long Short-Term Memory

BERT Bidirectional Encoder Representations from Transformers

MRR Mean Reciprocal Rank

NSP Next Sentence Prediction

MLM Masked Language Modelling

SBERT Sentence-BERT

PMMBV2 paraphrase-multilingual-mpnet-base-v2

CHAPTER 1

INTRODUCTION

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

The information contained within documents may contain errors or inconsistencies with real-world facts, this may not be suitable to use as a reliable source of information to answer questions accurately. Relying on a knowledge graph as a source of information to answer questions may be more appropriate because the triples contained within knowledge graphs represent set of facts. With the increase in methods of knowledge representation, knowledge graphs became a more important topic in artificial intelligence. In recent years new techniques are presented to represent nodes and relations in more powerful ways with transformer-based methods.[3–5]

This research introduces an approach to question answering for the Turkish language that is based on knowledge graph. The proposed approach aims to develop a methodology that is capable of drawing inferences from a knowledge graph to answer complex multi-hop questions.

This study presents the construction of the Beyazperde Movie Knowledge Graph (BPMovieKG) through the use of web data crawling techniques. Furthermore, the study introduces the TRMQA (Turkish Movie Question Answering) dataset, which is created by utilizing BPMovieKG to generate various question types and preparing question templates for 1-2 and 3-hop reasoning. To perform question answering tasks on the graph, a deep learning architecture is proposed. Firstly the study evaluates and compares the

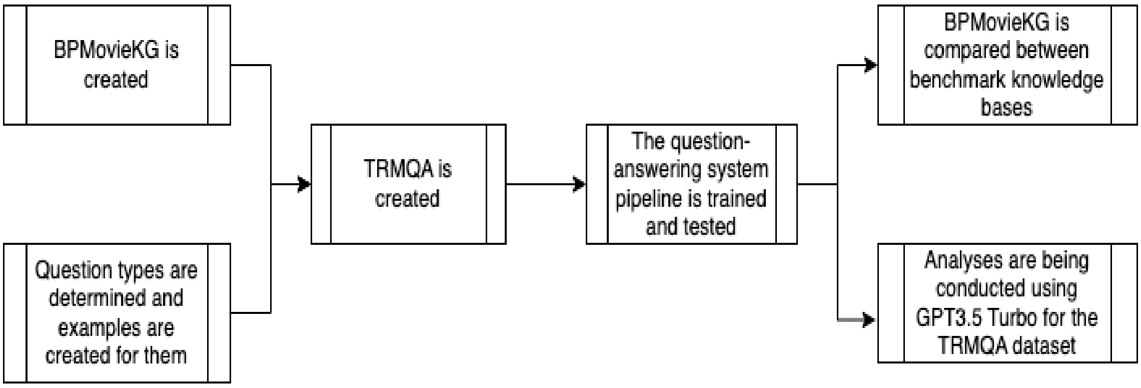

BPMovieKG is created
Question types are determined and examples are created for them
TRMQA is created
The question-answering system pipeline is trained and tested
BPMovieKG is compared between benchmark knowledge bases
Analyses are being conducted using GPT3.5 Turbo for the TRMQA dataset

Figure 1.1. The studies of the thesis is presented sequentially.

performance of different graph embedding methods and BPMovieKG against knowledge bases in literature. Secondly it evaluates question embedding techniques based on question answering system results. Finally, TRMQA dataset is evaluated using GPT3.5 Turbo[6] on both quantitative and qualitative analyses.

The studies conducted within this thesis, described above, as shown in the Figure 1.1. according to the order of execution.

In general, this study in question answering literature is mostly related to[7] since both approaches use similar methods. Our study differs from this study since it presents a different architecture to understand question and using of graph embedding methods on question-answering systems. In TRQA literature, our work is related with[8] both approaches use knowledge bases and NLP techniques but from a different perspective. Our study differs from this study as it incorporates both question and node representation learning, therefore our approach exhibits the capability to answer multi-hop questions by enabling retrieval of information from the knowledge graph up to three hops away.

The contributions of this study are as follows:

- We introduce the first Turkish QA system that utilizes knowledge graphs for multi-hop reasoning. To the best of our knowledge, there is no existing work on knowledge graphs and/or knowledge graph embeddings for Turkish QA systems.

- We are performing a comparison between the OpenAI GPT-3.5 Turbo, used the LlamaIndex [1] and a TRQA system for the first time in the literature. Additionally, we are comparing the BPMovieKG with benchmark knowledge graphs commonly used in the literature, based on the results obtained from various graph embedding

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

methods.

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

The remaining parts of this thesis are organized as follows. Chapter 2 describes the related work on link prediction methods, question answering systems, text embedding methods in literature. Chapter 3 begins by an explanation of knowledge graphs, followed by an explanation of how resources were collected and which resources will be utilized in the creation of the BPMovieKG. This section indicate the statistics of the BPMovieKG then describes how we create TRMQA dataset from BPMovieKG. In Chapter 4 we explain our method to capture answer from the knowledge graph. In chapter 5 discuss evaluations of both graph embedding methods and question-answering systems. Lastly, chapter 6 concludes the thesis.

CHAPTER 2

LITERATURE REVIEW

2.1. Question Answering Systems

Many studies are currently continuing to develop techniques for effectively answering questions posed in different languages using natural language processing. Various approaches were used for the task of question answering, which used rule-based, statistical, and neural methods, this study specifically focuses on the recent advancements in neural question answering. Our research primarily revolves around two specific domains of inquiry, specifically TRQA and Knowledge Graph Based Question Answering (KGQA).

2.1.1. Turkish QA Systems

In this section, we examine Turkish QA Systems using different approaches in the literature.[9] presents a closed-domain question answering system that operates through two distinct phases, namely question analysis and information retrieval. The objective of the question analysis module is to extract the focus, to be used in the information retrieval system. For this purpose, study uses a Hidden Markov Model for classification to extract the focus from the input question.

The information retrieval system, on the other hand, adopts a focused approach by combining the extracted focus with question and conducting searches on search engines like Indri and Apache Lucene to retrieve relevant documents.

[10] constructs a similar pipeline instead of 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. Then, patterns are prepared to identify named entities and keywords. Subsequently, the extracted keywords and named entities are stored in the database for each document.

Following 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.

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

[13] proposes an approach to solve the machine reading for question answering task. 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 construction of a Turkish Question Answering system using a transformer-based architecture, also introduces the first machine-reading comprehension dataset specifically designed for the Turkish language.

[14] uses 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, for the first time in the literature, the generation of Turkish questions from Turkish texts has been performed.

[8] address the challenge of question answering by leveraging the semantic web as a valuable knowledge resource. In this study a variety of NLP techniques are used to understand the question, create relationships between extracted named entities and convert the question into a SPARQL query. After question is converted into a SPARQL query, the semantic web is queried using this query, and the result of the query is provided as an answer to the question.

2.1.2. Knowledge Graph based QA Systems

[15] proposes a new approach for answering simple questions using a memory network architecture. They first introduce SimpleQuestions dataset based on Freebase knowledge base. Then proposes an architecture to store both knowledge base facts and questions in same vector space.

[16] proposes a method to understand question using template decomposition for question answering system. This approach uses knowledge graph and a text corpus to generate templates. After that, they align relations of generated templates and triples in knowledge base. After that these templates are used for answering simple and complex questions. Related entity and relation are extracted from question and used to find answer in generated templates.

[17] identifies the head entity and predicate in a question using a bidirectional LSTM(BiLSTM)[18] model, and then proposes a joint distance metric to find the most relevant fact in the knowledge graph based on the learned representations.

Recent methods combine knowledge graphs with a text corpus.[19,20] 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.

2.2. Embedding Methods

This study uses various methods to learn representations of natural language questions and nodes in the knowledge graph. The related literature is discussed below to obtain these representations.

2.2.1. Text Embeddings

In the literature, various methods have been trained and/or used for obtaining a vector representation of natural language input, typically given in the form of words or sentences. Below, these methods used or compared in this study are explained.

2.2.1.1. LSTM for Sentence Embeddings

Long Short-Term Memory[21] is a type of recurrent neural network (RNN) architecture that is widely used for sequence modeling tasks, including natural language processing. It can be used to construct sentence embeddings that capture the contextual information of the input sentence.

The LSTM model processes a sequence of words or tokens step by step, such as a sentence. It maintains a hidden state that captures the information learned from previous tokens and updates it as new tokens are processed.

Consider a sentence consisting of N words or tokens, represented as $x_1, x_2, \ldots, x_N$. The aim is to generate a sentence embedding s using LSTM.

The LSTM model consists of several key components:

1. **Word Embedding Layer**: Each word x_i is mapped to a continuous-valued word embedding $\mathbf{e}_i$ that represents its semantic meaning.

2. **LSTM Layer**: The LSTM layer uses word embeddings $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_N$ as input. At each time step t, the LSTM updates its hidden state $\mathbf{h}_t$ and cell state $\mathbf{c}_t$ based on the current input word embedding $\mathbf{e}_t$ and the previous hidden state $\mathbf{h}_{t-1}$.

3. **Final Sentence Embedding**: After all words are processed, the final hidden state $\mathbf{h}_N$ of the LSTM layer captures the contextual information of the entire sentence. This hidden state can be used as the sentence embedding s.

The output of the LSTM based hidden states obtained in this way has been used in various methodologies of recommended sentence embedding techniques, both in supervised and unsupervised techniques.[22,23]

2.2.1.2. Bidirectional Encoder Representations from Transformers - BERT

Bidirectional Encoder Representations from Transformers (BERT) is a language model proposed in.[24] BERT has had a significant impact on NLP tasks, such as sentiment analysis, named entity recognition, and question answering. It is based on transformer architecture.[25] It is pre-trained on masked language modeling (MLM) and next sentence prediction (NSP) objectives.

BERT achieves its power by pre-training a deep bidirectional representation of text using a corpus. This pre-training phase allows BERT to learn rich contextualized word representations, capturing both the left and right context of each word. Then BERT fine-tunes these pre-trained representations on downstream tasks.

Transformer-based Architecture: BERT is built upon the transformer architecture which consists of a stack of encoder layers. The encoder layers process the input text, combining self-attention mechanisms and position-wise feed-forward neural networks. The transformer's self-attention mechanism enables BERT to effectively capture contextual relationships between words.

Masked Language Modeling: In the pre-training phase, BERT uses a masked language modeling objective. In this objective, input sentences are randomly masked by replacing some words with a special [MASK] token. The objective is to predict the original word given the masked context.

Next Sentence Prediction: BERT uses a next sentence prediction objective during pre-training. It takes sentence pairs and trains the model to predict whether the second sentence follows the first sentence in the original corpus. This object helps BERT to understand the relationships between sentences.

Pre-training: BERT is pre-trained on large-scale unlabeled corpora, such as Wikipedia, using MLM and NSP objectives. The model is trained to minimize the combined loss of both objectives.

Fine-tuning: After pre-training, BERT is fine-tuned in downstream tasks. The pre-trained BERT model serves as a feature extractor, and additional task-specific layers are added on top. The entire model is then fine-tuned using labeled data from the specific task.

Various methods have been proposed to obtain sentence embeddings using transformer-based language models such as BERT.

- Pooling-based Approaches: One common method is to apply pooling operations, such as max pooling or average pooling, over the word embeddings in a sentence. This aggregates the information from all the words into a fixed-length vector representation for the sentence.
- Sentence Transformers: Another approach involves training specific models, known as sentence transformers, using transformer architectures. These models are designed to directly generate sentence embeddings by considering the context of the entire sentence.

2.2.1.3. Sentence-BERT - SBERT

SBERT (Sentence-BERT)[26] is an extension of the BERT model specifically designed for generating sentence embeddings. While BERT is primarily trained for word-level tasks, SBERT focuses on capturing the semantic meaning of entire sentences.

In SBERT different embedding techniques with transformer based language models are proposed. Most models in SBERT are trained for English. But there are also multilingual models.

Multilingual models are trained with multilingual knowledge distillation.[27] In training phase, two distinct models chosen as teacher and student. Teacher model produces sentence embeddings in one specific language. The student model is expected to imitate the teacher model. To ensure the student model can handle multiple languages, It is trained on parallel sentences with corresponding translations.

SBERT models generate contextualized word embeddings, which were then compressed into a n-dimensional vector representation through the use of different pooling techniques.

2.2.2. Graph Embeddings

In this section, various link prediction models used in this study are explained.

Knowledge graph represented as a set of triples $G = \{(h, r, t)\}$, where h represents the head entity, r represents the relation and t represents the tail entity. The goal of link prediction is to predict missing or future triples in the knowledge graph.

In training phase of link prediction models, a binary cross-entropy loss function is used, which is defined as:

$$L = -\sum_{(h,r,t)\in G} \log \sigma(f(h,r,t)) - \sum_{(h',r,t')\notin G} \log \sigma(-f(h',r,t'))$$

With this function, given a triple (h, r, t) from the knowledge graph, the model aims to maximize the score $f(h, r, t)$ to indicate a high likelihood of the triple being true. On the other hand, for a negative triple (h', r, t') that does not exist in the knowledge graph, the model aims to minimize the score $f(h', r, t')$ to indicate a low likelihood of the triple being true.

The loss function is optimized using gradient-based method which is Adam. During the training process, the embedding vectors e_i, r_j, and e_k are updated to minimize the loss function and improve the model's ability to predict missing or future triples in the knowledge graph.

2.2.2.1. DistMult

DistMult is a link prediction method proposed by.[28] It is a simple yet effective model that captures interactions between entities and relations in a knowledge graph.

DistMult assumes that each entity and relation in the knowledge graph can be represented as a low-dimensional embedding vector. The embedding vectors capture the semantic information of the entities and relations in a continuous vector space. Let $e_i \in \mathbb{R}^d$ and $e_k \in \mathbb{R}^d$ denote the embedding vectors of the head and tail entities, respectively, and let $r_j \in \mathbb{R}^d$ denote the embedding vector of the relation.

The scoring function of DistMult is defined as follows:

$$f(h, r, t) = \langle e_i, \text{diag}(r_j), e_k \rangle$$

where $\text{diag}(r_j)$ is a diagonal matrix with the elements of r_j along its diagonal.

DistMult measures head and tail entity compatibility using an entry-wise product. Since entry-wise product used in this study is symmetric, this method is unsuitable for asymmetric and anti-symmetric relations.

2.2.2.2. ComplEx

ComplEx is a link prediction method proposed by.[29] It extends the DistMult approach by representing entities and relations as complex-valued embeddings. The authors introduce ComplEx as a model that captures semantic information and rotational patterns in knowledge graphs.

ComplEx assumes that each entity and relation in the knowledge graph can be represented as complex-valued embeddings. Let $e_i \in \mathbb{C}^d$ and $e_k \in \mathbb{C}^d$ denote the complex embeddings of the head and tail entities, respectively, and let $r_j \in \mathbb{C}^d$ denote the complex embedding of the relation.

The scoring function of ComplEx is defined as:

$$f(h, r, t) = \text{Re}\left(\langle e_i, \text{diag}(r_j), \bar{e}_k\rangle\right)$$

where $\text{Re}(\cdot)$ returns the real part of a complex number, $\langle\cdot\rangle$ denotes the inner product, $\text{diag}(r_j)$ is a diagonal matrix with the elements of e_r along its diagonal, and $\bar{e}_k$ represents the conjugate of e_k.

With this method, model will be able to learn both symmetric and anti-symmetric relations using the complex space.

2.2.2.3. SimplE

SimplE is a link prediction method proposed by.[30] SimplE utilizes symmetric matrices to capture one-to-one and many-to-one relationships in a knowledge graph.

SimplE represents entities and relations as embedding vectors, and uses a scoring function that operates on symmetric matrices. Specifically, let $e_i \in \mathbb{R}^d$ and $e_k \in \mathbb{R}^d$ denote the embedding vectors of the head and tail entities, respectively, and let $r_j \in \mathbb{R}^{d\times d}$ denote the symmetric matrix associated with the relation.

The scoring function of SimplE is defined as:

$$f(h, r, t) = e_h^\top W_r e_t + e_t^\top W_r^\top e_h$$

With SimplE, authors introduce a new scoring based on the element-wise product of two embedding vectors instead of dot product to improve ComplEx and DistMult.

CHAPTER 3

CONSTRUCTION OF BPMovieKG & TRMQA DATASETS

3.1. Construction of BPMovieKG

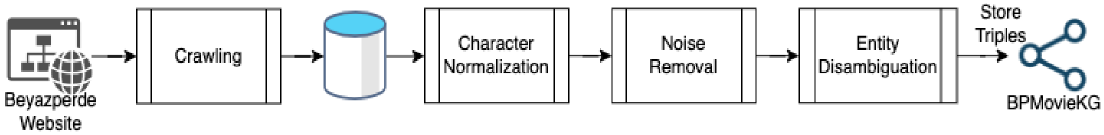


Figure 3.1. The construction process of BPMovieKG is illustrated step-by-step.

For the scope of this study, a knowledge graph on the movie domain as BPMovieKG is constructed. The relevant information is obtained by crawling the famous Turkish movie website beyazperde[1]. The construction of BPMovieKG has been illustrated step by step in 3.1. Below, these steps are explained.

3.1.1. Knowledge Graph

We can define the knowledge graph as a type of structured knowledge representation which consists of a combination of entities, relations that are represented as a set of triples. The triples in the knowledge graph corresponds to a basic unit of information, that consists of relation, subject and object. One triple example can be (The

[1] http://beyazperde.com

Matrix,yayinlanma_yili,1999) which represents a factual statement about the iconic science fiction film 'The Matrix' and its year of release, which was "1999".

Table 3.1. 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 distinct nodes in G
R_G	The number of distinct relations in G
T_G	The number of distinct triples in G

The notations and their descriptions about the knowledge graphs are listed in Table 3.1. The related notations used in knowledge graph, G for the knowledge graph itself, V for the set of entities or nodes in G, and E for the set of relations or edges in G. The notations for each individual entity nodes e_i and relation edges r_j, as well as notation for the triple (e_i, r_j, e_k) that connects two entity nodes with a relation are also included.

3.1.2. Crawling Process

The bs4[2] and selenium[3] are used in the crawling process. Not only movie descriptions but also the metadata about the movies, such as the release year, genre, language, budget, runtime, rating and also actor, director names and their professions, birth dates, and nationalities are collected.

[2] https://pypi.org/project/beautifulsoup4/

[3] https://selenium-python.readthedocs.io/

3.1.3. Preprocessing of Crawled Resources

3.1.3.1. Character Normalization

The crawled data contain different typos, conflicts, and exceptions. For instance, though the nodes "Yapımcı" and "Yapimci" refer to the same entity, their forms are different, the first one with the English letter "i" rather than the Turkish "ı". Yet another example is that, the movie name could have "filmi" at the end of the movie type, such as "Savaş" and "Savaş filmi". The entities are grouped by their types and edited manually, in order to be resolved correctly.

3.1.3.2. Noise Removal

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

3.1.3.3. Entity Disambiguation

In the data obtained by the web crawling, the conflicts are observed in director and actor node types. This issue is fixed by concatenating the node names that has unique ids. The similar name conflicts are also a case in the movies. This issue in the movies are resolved by, comparing the movie nodes based on the in-degree centrality and removing the movie that is less popular from the graph. The formula used to calculate in-degree centrality is given as,

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

After completing crawling and the preprocessing steps, we collect all remaining triples in a single graph and named it BPMovieKG. Figure 3.2. shows the subgraph that belongs to the movie "Thor Karanlık Dünya". This subgraph presents the "Thor Karanlık Dünya" and various relationships of that movie in the form of triples.

3.1.4. Statistics of BPMovieKG

The constructed BPMovieKG contains a total of 317,992 triples T_G, which relate to 36,489 unique nodes N_G and 16 unique relations R_G. Each relation is created in a bi-directional manner to ensure compatibility with graph embedding methods that commonly rely on undirected graphs to learn embeddings. Additionally, the TRMQA dataset consists of diverse question types that require access to backward information through relations in order to effectively infer an answer.

Table 3.2. includes the node types with their counts. Here, actors and directors are merged into the node type Person.

Node Type	Node Count
Person	25187
Movie	10958
Nationality	100
Year	93
Language	59
Profession	54
Genre	28
Rating	10

Table 3.2. Distribution of Node Types in BPMovieKG

Table 3.3. presents the frequency of the relations in BPMovieKG, indicating how many times each relation appears in the F. The limited number of movies for which rating values have been provided by website critics has resulted in a scarcity of data in the beyazperde_yildizi. This relation is missing in approximately %75 of the movies.

Relation Type	Relation Count
oyuncularindan_biridir	43583
oyuncularindan_biridir_reverse	43583
mesleklerinden_biridir	37623
mesleklerinden_biridir_reverse	37623
turlerinden_biridir	20031
turlerinden_biridir_reverse	20031
uyruklarindan_biridir	19232
uyruklarindan_biridir_reverse	19232
dillerinden_biridir	13397
dillerinden_biridir_reverse	13397
yonetmenlerinden_biridir	11673
yonetmenlerinden_biridir_reverse	11673
yayinlanma_yili	10947
yayinlanma_yili_reverse	10947
beyazperde_yildizi	2510
beyazperde_yildizi_reverse	2510

Table 3.3. Distribution of Relation Types in BPMovieKG

3.2. Construction of TRMQA

In this study, we construct the TRMQA dataset that contains 1-2, and 3-hop questions. It has 8 types of 1-hop, 19 types of 2-hop, and 14 types of 3-hop questions. Please refer to our github repository[4] for the related datasets and source codes.

In order to prepare the TRMQA dataset, we first design question types for each hop.

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

For each question type, a variety of questions were manually written to ensure the diversity of different types of questions. Each question type we have is actually represented as a path on the knowledge graph. We examined how many times each question type occurs on our knowledge graph. For each path that 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 were provided as the answer(s) to the question, and determine the question by selecting from a pool of manually written questions using random sampling.

In the creation of TRMQA, we are inspired by the MetaQA benchmark.[31,32] The purpose of publishing MetaQA benchmark is to expand the existing WikiMovies dataset in the literature and to release a multi-hop question dataset in the movie domain to evaluate models with reasoning capabilities.

Taking inspiration from this study, we create question types in the movie domain mentioned in the study for our own dataset and added different question types. In contrast to MetaQA, wherein the knowledge graph utilized includes distinct entities bearing identical names, resulting in inconsistencies between questions and their respective answers, we resolve such inconsistencies by generating our own knowledge graph wherein each entity is designated by a unique name. For instance, in MetaQA, there is a question as follows: ’Who’s 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 to this question 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, while the other is associated with Stephen Potter as writer.

Dataset	1-hop	2-hop	3-hop
Train	250296	287081	214688
Test	15389	21636	10946
Dev	15388	21629	10938

Table 3.4. Statistics for TRMQA dataset

Table 3.4. presents the statistics for TRMQA dataset in the train, test, and development partitions. We stratified question types equally among the train, test, and development sets, using split ratios of $0.9 : 0.05 : 0.05$, $0.88 : 0.065 : 0.065$, and $0.91 : 0.045 : 0.045$ respectively. Figures 3.3.,3.4., and 3.5. depict examples of 1-hop, 2-hop, and 3-hop questions, respectively.

To answer a question like "Hangi kişi Karayip Korsanları Salazar'ın intikamı filminde yönetmendir?", it's sufficient to go through one edge on the graph. The associated reasoning path is shown in Figure 3.3.

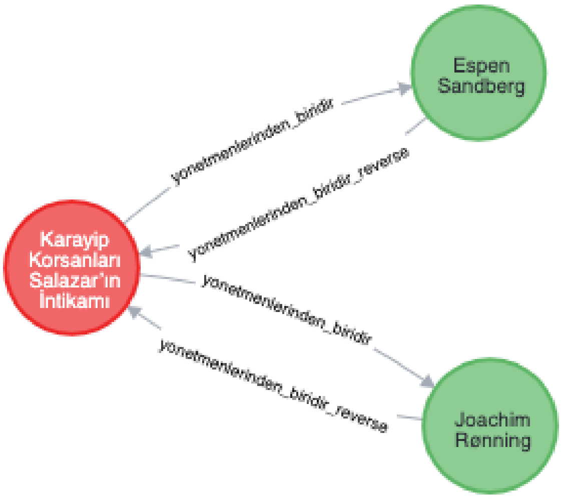

Espen Sandberg
yönetmenlerinden_biridir
yönetmenlerinden_biridir_reverse
Karayip Korsanları Salazar'ın İntikamı
yönetmenlerinden_biridir
yönetmenlerinden_biridir_reverse
Joachim Rønning

Figure 3.3. 1-hop reasoning example for "Movie to Director" question type

On the other hand, the answer to the question "Onur Saylak'ın oynadığı filmler hangi temalardadır?" requires two-hop connection on the movie graph. First, we need to infer movies by the given actor then find the genres of the inferred movies. Figure 3.4. illustrates the reasoning path for this example.

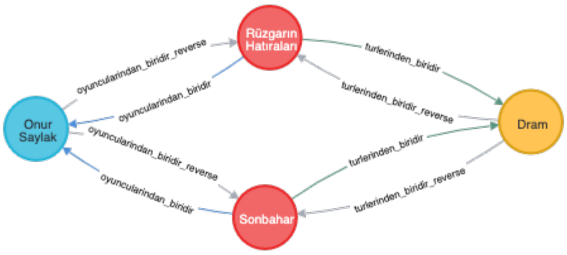


Figure 3.4. 2-hop reasoning example for "Actor to Movie to Genre" question type

Finally, a three-step reasoning is necessary to answer the question "Eric Reed yönetmeninin yönettiği filmlerin 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 should ask for the nationalities of these directors. Figure 3.5. depicts the associated reasoning path.

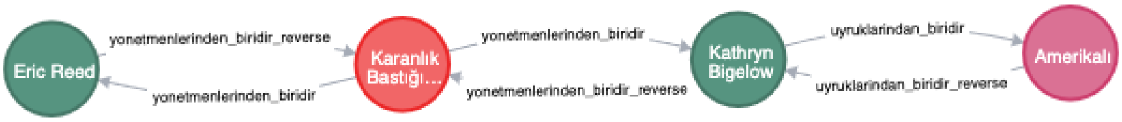


Figure 3.5. 3-hop reasoning example for "Director to Movie to Director to Nationality" question type

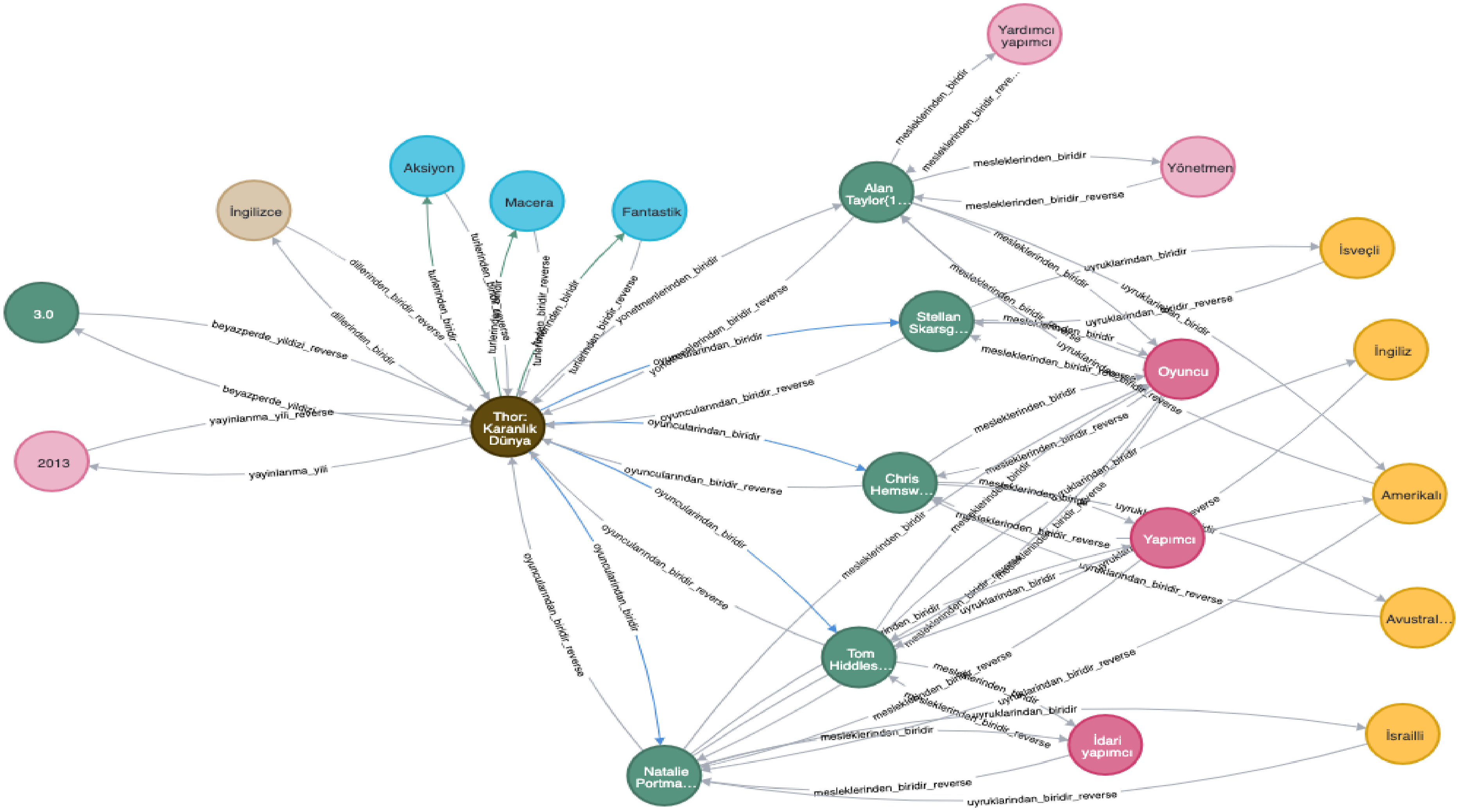


Figure 3.2. Subgraph of BPMovieKG

CHAPTER 4

METHOD

4.1. Method Architecture

In this section, we presented our approach which consists of three distinct modules, namely, Graph Embedding, Question Embedding, and Answer Selection. These modules are designed to build different steps of the question answering process, from capturing semantic relationships in the KG, to encoding natural language questions, and finally, selecting the best answers from a set of candidates. The structure of the modules involved in our method is illustrated in 4.1.

The baseline study utilized a ComplEx graph embedding approach with a BiLSTM and three fully connected layers to determine the correct answer.[7] In our study, we extracted representations of questions using a fine-tuned SBERT model, which we then applied mean pooling along with three fully connected layers. In addition to the ComplEx model used in the baseline study, our study also use DistMult and SimplE models.

To illustrate with an example, let's consider the question, "The Matrix filmi ne zaman yayınlanmıştır?". The system inputs "The Matrix" into the graph embedding module, and the question is given into the question embedding module. The output

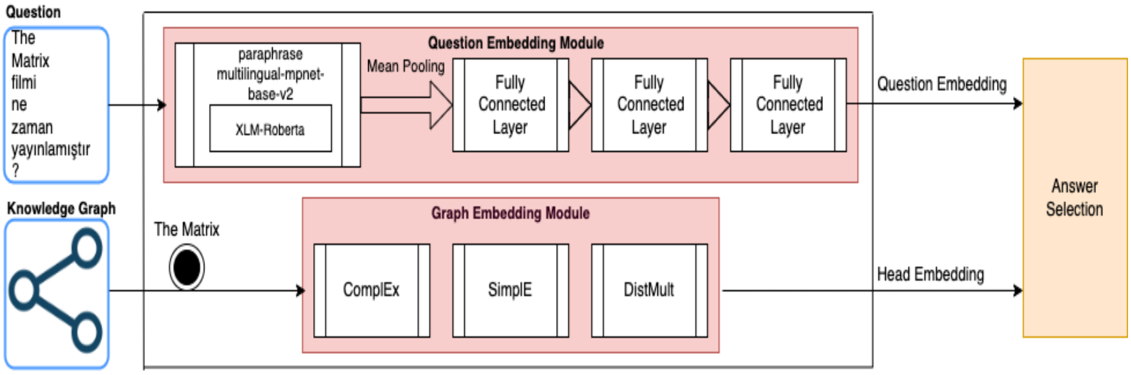


Figure 4.1. The structural design of our approach is illustrated

from the graph embedding module is treated as head while the output from the question embedding module is considered as the relation. These embeddings are then passed to the Answer Selection module. In the Answer Selection module, a score has been calculated for each node present in BPMovieKG based on the scoring function. These scores are then passed through a sigmoid function, and the node with the highest probability is determined as the tail, which in this case should be "1999".

4.2. Graph Embedding Module

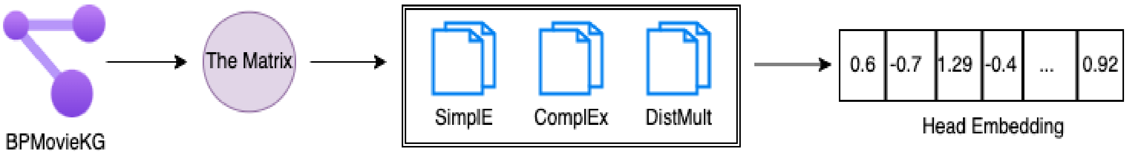


Figure 4.2. Graph Embedding Module Flow

This module takes a node as input and, based on the used graph embedding method, provides the representation of that node to the Answer Selection module. In this module DistMult, ComplEx and SimplE methods are used to construct node and relation embeddings from BPMovieKG. The flow of this module is shown in 4.2.

The Graph Embedding module was trained independently. Weights in this module were frozen during the training on Question Answering System. Before the training, triples were divided into train, test, and development sets, based on a $0.9 : 0.05 : 0.05$ split ratio and it was ensured that the relationships were distributed stratify during the process. 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 has created a loss score that could be optimized using the available graph embedding methods.

4.3. Question Embedding Module

The purpose of the Question Embedding module is to obtain a representation of the question in vector space. To accomplish these, we used SBERT as sentence encoder. We used a pre-trained multilingual model called paraphrase-multilingual-mpnet-base-v2 (PMMBV2).

The PMMBV2 model was trained using multilingual knowledge distillation, as stated in the SBERT section found in the literature review. In the training phase paraphrase-mpnet-base-v2[33] was used as teacher model to generate embeddings for English sentences and XLM-RoBERTa[34] was used as student model to generate embeddings for English and other languages such as Turkish.

The PMMBV2 model produced contextualized word embeddings, which were subsequently compressed into a 768-dimensional vector representation using mean pooling. The vector was processed through three fully connected linear layers with ReLU activation function. After activation function, the obtained embedding was given as input for Answer Selection module.

4.4. Answer Selection Module

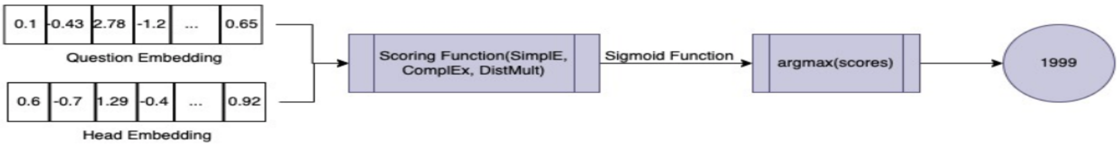


Figure 4.3. Answer Selection Module Flow

This module finds the answer by approaching to the Question Answering System as a link prediction problem. Answer Selection module flow is shown in 4.3.

The answer selection module takes the question and entity embeddings as input. The embedding representing the question is treated as a relation embedding, while the embedding for the entity serves as the head node. The module aims to accurately predict the tail node accurately by combining head and relation embeddings. The head and relation embeddings are given as input to the graph embedding method. As a result, a score is generated for each node within BPMovieKG. The scores pass through a sigmoid function, and the node with the highest probability is determined as the answer.

CHAPTER 5

EXPERIMENTS AND DISCUSSION

5.1. Experimental Setup

In this section, the experimental results are presented in three parts. The first part presents the results of graph embedding methods obtained from the knowledge bases presented in[35] and BPMovieKG. Additionally, it highlights the differences between the knowledge bases used in the study according to graph embedding metrics. In the second part, we compare the results of the question-answering system based on the results of 1, 2, and 3 hop questions. In the last part, the 1-hop questions from TRMQA are asked to the OpenAI GPT-3.5 Turbo module and the results are discussed.

5.1.1. Hyperparameter Setup

Table 5.1. Graph Embedding Model Hyperparameters

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

The hyperparameters used in the three different Graph Embedding methods are set the same as those used in the baseline study, and are shown in Table.5.1.

Table 5.2. Question Answering Model Hyperparameters

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

The hyperparameters used in the question answering system are shown in Table 5.2. The BiLSTM size is set to 256 in all the methods used, and all hyperparameters are set the same since are those used in the baseline study. Additionally, during the training of this system, early stopping is applied. The progress of the training process is monitored by evaluating the Hits@1 metric on the validation set every 5 epochs. If the result does not exceed the best result obtained within the previous 5 evaluations after finding the highest result, the training process is terminated, and the best model is selected.

5.1.2. Metrics

In this study, Hits@1, Hits@3, Hits@10, and MRR(Mean Reciprocal Rank) metrics are being used to compare graph embedding methods, while Hits@1 is being used to compare question answering system results.

Hits@K Hits@K is a measure of effectiveness of model, which calculates ratio of predictions that are ranked at or below a specific threshold which is K.

$$\text{Hits@k} = \frac{1}{|N|}\sum_{i=1}^{N}\frac{|\hat{S}\cap S|}{|S|}$$

where $|\hat{S}\cap S|$ represents the number of correctly predicted nodes among the top-k ranked nodes, where $|N|$ denotes the size of the dataset given for inference. S denotes the set of ground truth nodes, $\hat{S}$ denotes the set of predicted nodes.

Mean Reciprocal Rank MRR is calculated as the average of the reciprocals of the ranks of the ground-truth triples.

Let R be the set of ground-truth triples, and let R_i represent the ground-truth triple at position i. Assume that the ranking model assigns ranks to the triples in a ranked list.

The reciprocal rank for a given ground-truth triple R_i is defined as:

$$\text{Reciprocal Rank}(R_i) = \frac{1}{\text{Rank}(R_i)}$$

where $\text{Rank}(R_i)$ represents the rank of the ground-truth triple R_i in the ranked list.

(MRR) is then computed as:

$$\text{MRR} = \frac{1}{|R|}\sum_{i=1}^{|R|}\text{Reciprocal Rank}(R_i)$$

where $|R|$ denotes the total number of ground-truth triples.

The MRR metric provides an average measure of the quality of the ranking model, with higher values indicating better performance.

5.1.3. Knowledge Bases

WN18 is a knowledge base constructed from WordNet. Each node represents a synset and synsets are connected between relations.

FB15k is a knowledge base constructed from Freebase. Each triple represents relation between different node types such as persons, movies, sports, and more.

In Table 5.3. we present the statistics of our knowledge graph, as well as the WN18 and FB15k datasets.

Table 5.3. Knowledge Base Statistics

Knowledge Base	Entities	Relations	Training	Validation	Test
WN18	40,943	18	141,442	5000	5000
FB15k	14,951	1345	483,142	50000	59071
BPMovieKG	36,489	16	288,992	14500	14500

5.2. Graph Embedding Methods Results

Firstly, by running all three methods using the hyperparameters listed in 5.1., results were obtained for all three datasets. These results are presented in 5.4.

In our study, considering BPMovieKG as proposed, the SimplE method has provided the best results in terms of all evaluation metrics. Consistent with previous studies in the literature, the DistMult method gave the worst result among these three datasets.

Furthermore, as we can observe, the ComplEx method outperforms the SimplE method in the FB15k dataset, while in the WN18 dataset, only in the Hits@10 metric, the SimplE method outperforms the ComplEx method, and both methods yield similar results in other metrics. As the number of relations within the datasets increases, the ComplEx method tends to yield better results compared to the SimplE method in these experiments.

As can be seen in Table 5.4., the worst-performing dataset for the methods is FB15k. In Table 5.3., although FB15k has a significantly larger training set size, it also has 80 times more relations compared to other datasets. This expands the space of predicted relations, which may have led to poorer performance of the methods in this dataset during link prediction.

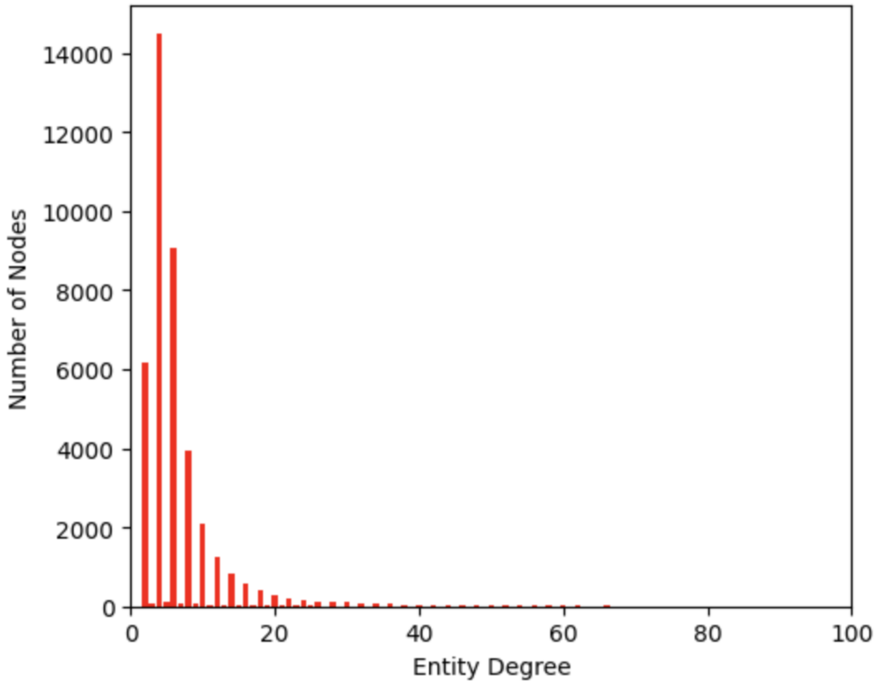


Figure 5.1. Distribution of entity degrees in WN18.

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 5.4. Performance Comparison of ComplEx, DistMult, and SimplE on the WN18, FB15k and BPMovieKG Datasets

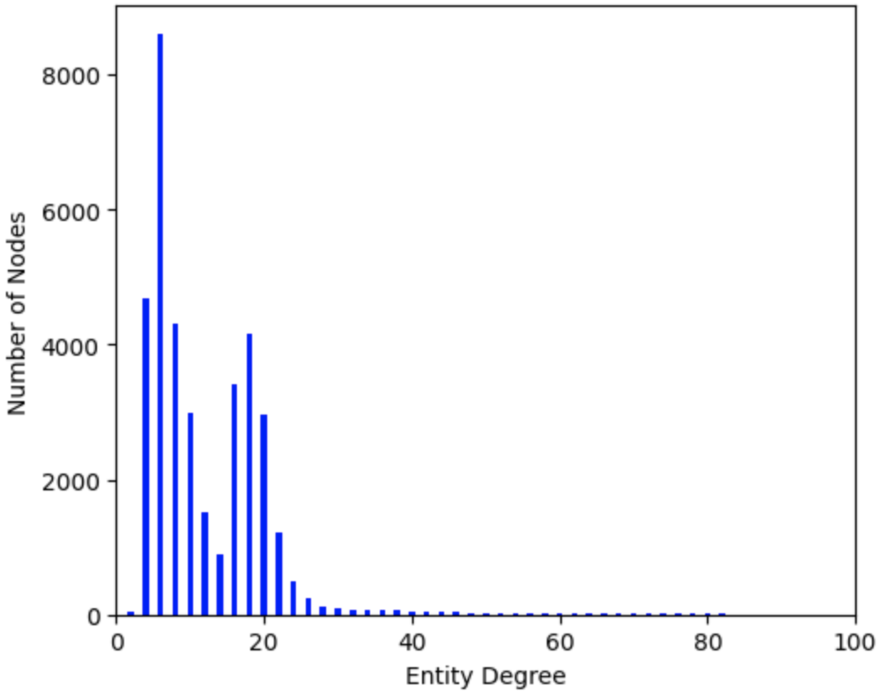


Figure 5.2. Distribution of entity degrees in BPMovieKG.

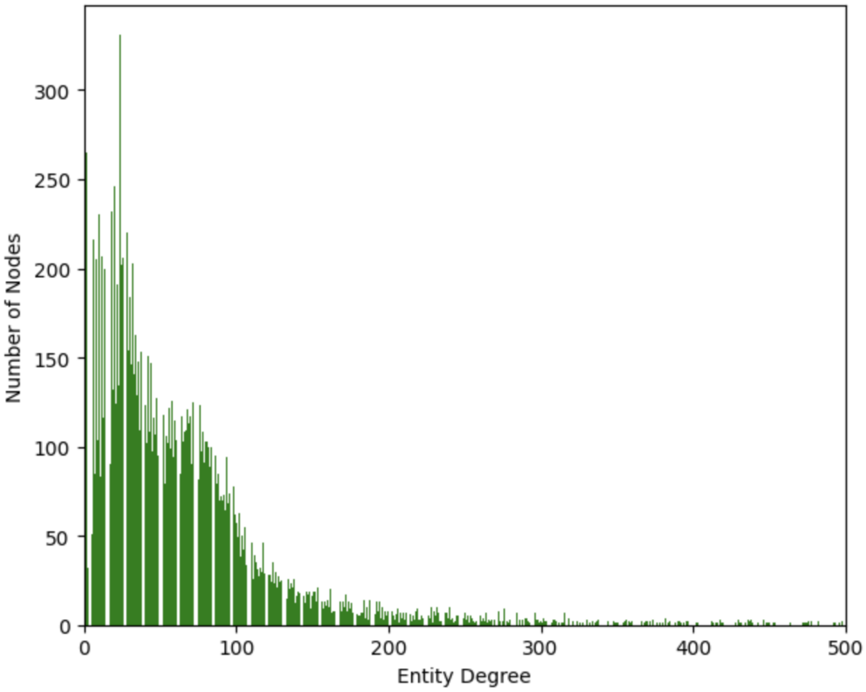


Figure 5.3. Distribution of entity degrees in FB15k.

The distribution of entity degrees for each dataset is shown in 5.1., 5.2. and 5.3. As observed from the distributions, the entity degrees in the FB15k dataset, which is the least generalizable dataset among those used in this study, 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 compared to the other two datasets. Based on the datasets and methods used in this study, it is demonstrated that as the distribution range of entity degrees widens and the entity degree values increase, the used methods showed lower performance. The dataset created in this study showed a distribution of entity degree values that fall between the distributions of the two datasets.

5.3. Question Answering System Results

The results obtained on the TRMQA dataset containing 1-2-3 hops for the proposed method and the baseline architecture are presented in 5.5. Each method was run five times, and the results were averaged and presented.

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

Table 5.5. Results of Testing and Validation (% hits@1) on datasets involving 1-2-3 hops.

As seen from the results, all methods were able to correctly answer all 1-hop questions in both datasets. Recent studies[36,37] on the MetaQA dataset have shown that they perform better on 2 and 3-hop questions than on 1-hop questions. This is due to the ambiguity caused by the entities with the same name in MetaQA. However, in our study, this issue did not arise in the BPMovieKG dataset, as each entity name is unique in this dataset.

The baseline method used for 2-hop questions gave the best results on both the test and validation datasets, while SimplE+SBERT gave the best results for 3-hop questions.

Since the context of 3-hop questions is wider and the questions are longer and more detailed, models that can capture longer sequences tend to perform better. Additionally, while the ComplEx method performs better for 2-hop questions, SimplE yields better results for 3-hop questions.

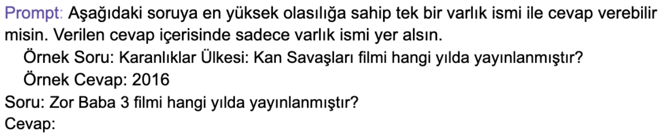
As observed in the results of embedding methods, DistMult was the least generalizable method for BPMovieKG. This was also reflected in the question answering system, where DistMult consistently yielded the lowest scores in both 2-hop and 3-hop evaluations.

5.4. Evaluation of 1-Hop Questions in TRMQA Dataset Using GPT

In this section, we present 1-hop questions from TRMQA that were evaluated using OpenAI's GPT-3.5 Turbo model. The model is fine-tuned on GPT3.5 which is a subclass of GPT-3.

By using the GPT3.5 Turbo model, a question answering system was developed to enable the evaluation of the Hits@K metric. To achieve this, first, we created a text document containing the names of all the nodes in the BPMovieKG. After that, we partitioned it into text segments, each containing 4096 tokens. For each text chunk, we obtained embeddings through LlamaIndex using GPT3.5 Turbo. We concatenated all the embeddings to obtain a single embedding.

To ask questions to GPT3.5 Turbo, we created 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 it when asking the question. An example of the prompt is shown in 5.4.



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

In the end, the embedding obtained from the prompt and the embedding obtained from the text chunks were concatenated and then given to GPT3.5 Turbo to generate an answer.

So, we received only a single entity name as the answer to the questions we asked, and this entity name matched one of the nodes within BPMovieKG. Thus, we were able to evaluate the given answers using the Hits@1 metric.

Questions containing the relation "beyazperde_yildizi" have not been included in the evaluation. This is because "beyazperde_yildizi" information is specific to that website and not a general knowledge.

5.4.1. Quantitative Analysis

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

Table 5.6. Evaluation results of the 1-hop Questions based on Hits@1.

In this section, we begin by conducting a quantitative evaluation of our dataset.

We evaluate the answers provided by GPT3.5 Turbo and compare them to our method using the Hits@1 metric.

In open question answering systems, 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 it provides may be incorrect for the entity we are actually asking about. As a result of this issue, we first evaluated our entire dataset and labeled it as "All". In a separate analysis, we filtered out some entities from our dataset and labeled it as "Filter". These evaluations are presented in 5.6.

To filter the dataset, we utilized the Cinemagoer[1] library to determine the frequency of movie or person entities within the movie domain for each question. If the entity appeared only once, we used it in the evaluation process.

As seen from 5.6., the results obtained from the Filtered dataset for all relation types were better than the results obtained from the All dataset in both test and validation. From this, we understand that when using the All dataset without filtering, the answers provided by GPT3.5 Turbo for certain persons or movies may also belong to different persons or different movies with the same name, which could lead to ambiguity. Therefore, the filtered dataset yields more accurate results because it only include entities that do not give rise to ambiguity.

In question types where the answer is a person which are 'oyuncularindan_birisidir' and 'yonetmenlerinden_birisidir' question types, GPT3.5 Turbo provide worse answers compared to other question types. Due to %69 of the nodes in the knowledge graph representing persons, the model may struggle in correctly identifying the right person since the search space is larger.

As seen from the results, the GPT3.5 Turbo has performed better in answering questions related to personal information, such as questions type 'mesleklerinden_birisidir' and 'uyruklarindan_birisidir' compared to questions about movie information. This could be because during the training phase of GPT3.5 Turbo, the trained corpora contained more information about persons and related information compared to movies and related information.

Additionally, our system is able to answer all 1-hop questions accurately, while

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

fine-tuned GPT3.5 Turbo model used in this study can answer fewer questions correctly.

5.4.2. Qualitative Analysis

In the Qualitative Analysis section, we will discuss some of the common wrong answers identified when comparing the results obtained from GPT3.5 Turbo with the ground truths.

Question-1: Hangi yıl Trendeki Kız filmi yayınlanmıştır?
Ground-Truth: 2016
GPT3.5 Turbo Answer: 2021
Discussion: There are multiple films titled 'Trendeki Kız' One of these films was released in 2021, while another was released in 2016. GPT3.5 Turbo provided an answer for the most recent released film, stating it as 2021. However, the film present in BPMovieKG was released in 2016. This leads to ambiguity.

Question-2: Zehirli Element filmindeki aktörlerden biri kimdir?
Ground-Truth: Paddy Considine
GPT3.5 Turbo Answer: Iron Man
Discussion: In second example, one of the actors from the film 'Zehirli Element' was asked for. While an actor was expected as the answer, the GPT3.5 Turbo model provided a film name as the response. In some evaluated questions, the model struggles to understand the type of answer the questions require.

Question-3: Gilbert'in Hayalleri filminde hangi tema işlenmiştir?
Ground-Truth: Dramatik komedi
GPT3.5 Turbo Answer: Dram
Discussion: In this example, one of the genres of the film 'Gilbert'in Hayalleri' is being asked. Although the GPT3.5 Turbo model was able to identify that the genre mentioned in the film is 'Dram' it failed to find the more specific genre of 'Dramatik Komedi' as the

answer.

Question-4: Hangi yıl Yarın Asla Ölmez filmi vizyona girmiştir?

Ground-Truth: 1997

GPT3.5 Turbo Answer: 1987

Discussion: This example asked the release year of ’Yarın Asla Ölmez’ one of the James Bond films. However, GPT3.5 Turbo answered the release year of another Bond film which is ’Gün Işığında Suikast’. Such incorrect responses may indicate that the model struggles to distinguish between sequel films.

CHAPTER 6

CONCLUSION

In this thesis, we have developed two datasets and presented a question answering system that built on these datasets. We use various graph embedding techniques to evaluate knowledge bases found in literature and compared them with BPMovieKG. Furthermore, we compared our proposed approach with a baseline study to understand the impact of graph embedding methods and SBERT on the question answering system. Additionally, we performed qualitative and quantitative analyses on TRMQA using GPT3.5 Turbo. Our findings are;

The WN18, FB15k, and BPMovieKG knowledge bases were compared using graph embedding methods. While the SimplE method generally yielded the best results, the ComplEx method performed better in knowledge bases with a high number of relations. Additionally, graph embedding methods exhibited better performance in knowledge bases with lower entity degree values and within lower ranges compared to other knowledge bases.

The graph embedding methods used in the question answering system directly effected the results. The DistMult method, which obtained the worst results among the graph embedding methods, also received worse results in the question answering system compared to other methods. The 2-hop dataset consist of 19 question types, while the 3-hop dataset consist of 14 question types. As each question type is treated as a relation in the question answering system, the SimplE method has better generalized the 3-hop dataset, while the ComplEx method has performed better on the 2-hop dataset with a higher number of relations. While BiLSTM provides better representation for 2-hop questions, the SBERT model used for 3-hop questions provides better representation.

When a question answering system was built using GPT3.5 Turbo on the 1-hop dataset, Experiments showed that the GPT3.5 Turbo model is more prone to making errors rather than a more specific model. This may be due to the lack of sufficient information about the film domain in the corpora used to fine-tune the GPT3.5 Turbo model. Another reason may be that the GPT3.5 Turbo model cannot distinguish named entities.

As a future work, the performed analyses in this study can be further developed to create reliable question answering systems by querying knowledge graphs using with large language models. These systems can be designed to prevent hallucinations and allow reasoning paths beyond 3-hop.

REFERENCES

(1) Er, N. P.; Cicekli, I. In *Sixth International Joint Conference on Natural Language Processing, IJCNLP 2013, Nagoya, Japan, October 14-18, 2013*, Asian Federation of Natural Language Processing / ACL: 2013, pp 854–858.

(2) Amasyali, M. F.; Biricik, G.; Solmaz, S. E.; Özdemir, E. In 2013.

(3) Li, D.; Yi, M.; He, Y. *CoRR* **2022**, *abs/2201.04843*.

(4) Wang, B.; Shen, T.; Long, G.; Zhou, T.; Chang, Y. *CoRR* **2020**, *abs/2004.14781*.

(5) Bi, Z.; Cheng, S.; Zhang, N.; Liang, X.; Xiong, F.; Chen, H. *CoRR* **2022**, *abs/2205.10852*, DOI: `10.48550/arXiv.2205.10852`.

(6) Brown, T. B. et al. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, ed. by Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; Lin, H., 2020.

(7) Saxena, A.; Tripathi, A.; Talukdar, P. P. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, ed. by Jurafsky, D.; Chai, J.; Schluter, N.; Tetreault, J. R., Association for Computational Linguistics: 2020, pp 4498–4507.

(8) Tasar, C. O.; Komesli, M.; Ünalir, M. O. *CoRR* **2023**, *abs/2301.04752*, DOI: `10.48550/arXiv.2301.04752`.

(9) Derici, C.; Aydin, Y.; Yenialaca, Ç.; Aydin, N. Y. *Nat. Lang. Eng.* **2018**, *24*, 725–762.

(10) Çelebi, E.; Günel, B.; Şen, B. In *2011 International Symposium on Innovations in Intelligent Systems and Applications*, 2011, pp 389–393.

(11) Yigit, G.; Amasyali, M. F. In *2019 Innovations in Intelligent Systems and Applications Conference (ASYU)*, 2019, pp 1–5.

(12) Weston, J.; Bordes, A.; Chopra, S.; Mikolov, T. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, ed. by Bengio, Y.; LeCun, Y., 2016.

(13) Gemirter, C.; Goularas, D. *Journal of Intelligent Systems Theory and Applications* **2021**, *4*, 65–75.

(14) Akyön, F. Ç.; Çavusoglu, A. D. E.; Cengiz, C.; Altinuc, S. O.; Temizel, A. *Turkish J. Electr. Eng. Comput. Sci.* **2022**, *30*, 1931–1940.

(15) Bordes, A.; Usunier, N.; Chopra, S.; Weston, J. *CoRR* **2015**, *abs/1506.02075*.

(16) Zheng, W.; Yu, J. X.; Zou, L.; Cheng, H. *Proc. VLDB Endow.* **2018**, *11*, 1373–1386.

(17) Huang, X.; Zhang, J.; Li, D.; Li, P. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019*, ed. by Culpepper, J. S.; Moffat, A.; Bennett, P. N.; Lerman, K., ACM: 2019, pp 105–113.

(18) Schuster, M.; Paliwal, K. K. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681.

(19) Sun, H.; Bedrax-Weiss, T.; Cohen, W. W. 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*, ed. by Inui, K.; Jiang, J.; Ng, V.; Wan, X., Association for Computational Linguistics: 2019, pp 2380–2390.

(20) Sun, H.; Dhingra, B.; Zaheer, M.; Mazaitis, K.; Salakhutdinov, R.; Cohen, W. W. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, ed. by Riloff, E.; Chiang, D.; Hockenmaier, J.; Tsujii, J., Association for Computational Linguistics: 2018, pp 4231–4242.

(21) Hochreiter, S.; Schmidhuber, J. *Neural Comput.* **1997**, *9*, 1735–1780.

(22) Kiros, R.; Zhu, Y.; Salakhutdinov, R.; Zemel, R. S.; Urtasun, R.; Torralba, A.; Fidler, S. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, ed. by Cortes, C.; Lawrence, N. D.; Lee, D. D.; Sugiyama, M.; Garnett, R., 2015, pp 3294–3302.

(23) Conneau, A.; Kiela, D.; Schwenk, H.; Barrault, L.; Bordes, A. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, ed. by Palmer, M.; Hwa, R.; Riedel, S., Association for Computational Linguistics: 2017, pp 670–680.

(24) Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. *CoRR* **2018**, *abs/1810.04805*.

(25) Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, ed. by Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; Garnett, R., 2017, pp 5998–6008.

(26) Reimers, N.; Gurevych, I. 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*, ed. by Inui, K.; Jiang, J.; Ng, V.; Wan, X., Association for Computational Linguistics: 2019, pp 3980–3990.

(27) Reimers, N.; Gurevych, I. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, ed. by Webber, B.; Cohn, T.; He, Y.; Liu, Y., Association for Computational Linguistics: 2020, pp 4512–4525.

(28) Yang, B.; Yih, W.; He, X.; Gao, J.; Deng, L. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, ed. by Bengio, Y.; LeCun, Y., 2015.

(29) Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; Bouchard, G. *CoRR* **2016**, *abs/1606.06357*.

(30) Kazemi, S. M.; Poole, D. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, ed. by Bengio, S.; Wallach, H. M.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; Garnett, R., 2018, pp 4289–4300.

(31) Zhang, Y.; Dai, H.; Kozareva, Z.; Smola, A. J.; Song, L. 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*, ed. by McIlraith, S. A.; Weinberger, K. Q., AAAI Press: 2018, pp 6069–6076.

(32) Miller, A. H.; Fisch, A.; Dodge, J.; Karimi, A.; Bordes, A.; Weston, J. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, ed. by Su, J.; Carreras, X.; Duh, K., The Association for Computational Linguistics: 2016, pp 1400–1409.

(33) Song, K.; Tan, X.; Qin, T.; Lu, J.; Liu, T. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, ed. by Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; Lin, H., 2020.

(34) Conneau, A.; Khandelwal, K.; Goyal, N.; Chaudhary, V.; Wenzek, G.; Guzmán, F.; Grave, E.; Ott, M.; Zettlemoyer, L.; Stoyanov, V. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, ed. by Jurafsky, D.; Chai, J.; Schluter, N.; Tetreault, J. R., Association for Computational Linguistics: 2020, pp 8440–8451.

(35) Bordes, A.; Usunier, N.; García-Durán, A.; Weston, J.; Yakhnenko, O. In *NIPS*, 2013.

(36) He, G.; Lan, Y.; Jiang, J.; Zhao, W. X.; Wen, J. In *WSDM '21, The Fourteenth ACM International Conference on Web Search and Data Mining, Virtual Event, Israel, March 8-12, 2021*, ed. by Lewin-Eytan, L.; Carmel, D.; Yom-Tov, E.; Agichtein, E.; Gabrilovich, E., ACM: 2021, pp 553–561.

(37) Shi, J.; Cao, S.; Hou, L.; Li, J.; Zhang, H. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, ed. by Moens, M.; Huang, X.; Specia, L.; Yih, S. W., Association for Computational Linguistics: 2021, pp 4149–4158.

APPENDIX A

EXAMPLES OF THE 1-2 AND 3-HOP QUESTION TYPES AND EXAMPLES

Question Type	Count	Example
Movie to Release Year	10947	Hangi sene İhtiyarlara Yer Yok filmi yayınlandı?
Movie to Language	13240	Hangi dil Çifte Hayatlar filminin seslendirilmesinde kullanılmıştır?
Actor to Profession	36750	Jennifer Ulrich kişisi meslek olarak ne yapmaktadır?
Movie to Actor	40238	Hicran Gecesi filminde rol alan aktörlerden biri kimdir?
Movie to Genre	19326	Straight Outta Compton filminin kategorisi nedir?
Actor to Nationality	19184	Hangi uyruk Zoë Wanamaker kişisinin uyruğudur?
Movie to Director	11620	Hangi kişi Bıçağın İki Yüzü filminde yönetmendir?
Movie to Rating	2510	Kevin Hakkında Konuşmalıyız filminin aldığı yıldız değeri kaçtır?

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

Question Type	Count	Example
Movie to Release Year to Movie	10947	Yarının Sınırında filminin yayınlanma yılında hangi filmler vizyona girmiştir?
Movie to Language to Movie	4377	13. Savaşçı filmi içerisinde konuşulan lisanlardan biri aynı zamanda hangi filmlerde kullanılmaktadır?
Movie to Genre to Movie	4382	Hangi filmlerin temaları The Company Men filminin temaları ile kesişmektedir?
Movie to Director to Movie	7324	Constantine filminin yönetmeninin yönettiği diğer filmler nelerdir?
Movie to Actor to Movie	10012	Fury filminde yer alan oyuncular aynı zamanda hangi filmde rol almıştır?
Movie to Actor to Nationality	10688	Parabellum filmindeki oyuncular hangi uyruktan gelmektedirler?
Movie to Actor to Profession	10957	Teksas Katliamı: Başlangıç filmindeki aktörler hangi meslek ile uğraşmaktadır?
Movie to Director to Profession	10947	Ocean's 12 filminin yönetmenlerinin meslekleri nelerdir?
Movie to Director to Nationality	9470	Yeşil Sokak Holiganları filminin yönetmenleri hangi milliyettendir?
Actor to Movie to Director	19632	Hangi yönetmenler Sari Lennick'in oynadığı filmleri yönetmiştir?
Actor to Movie to Genre	19788	Mel Gibson'in oynadığı filmler hangi temalardadır?
Actor to Movie to Release Year	19758	Ethan Hawke'ın yer aldığı filmlerin yayınlanma yılı nedir?
Actor to Movie to Language	19737	Daniel Brühl'in oynadığı filmlerde kullanılan diller nelerdir?
Actor to Movie to Actor	19792	Hangi aktörler ile Nick Cave aynı filmde yer almıştır?
Director to Movie to Actor	6095	Bruce Lee hangi oyuncuların oynadığı filmleri yönetmiştir?
Director to Movie to Director	977	Hakan Gürtop'in yönettiği filmlerde başka hangi yönetmenler bulunmaktadır?
Director to Movie to Release Year	6089	Hangi senelerde Katharine O'Brien'in yönettiği filmler yayınlanmıştır?
Director to Movie to Genre	6094	Stuart Beattie hangi kategorilere ait filmleri yönetmiştir?
Director to Movie to Language	6082	Hangi diller Peter Weir'in çektiği filmlerde kullanılmıştır?

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

Question Type	Count	Example
Actor to Movie to Actor to Profession	19788	Hangi iş üzerine Amanda Peet oyuncusu yer aldığı filmlerde bulunan aktörler uğraşmaktadır?
Actor to Movie to Actor to Nationality	18558	Hangi milletler Arielle Holmes oyuncusunun yer aldığı filmlerdeki oyuncuların milletleridir?
Director to Movie to Director to Profession	977	Hangi meslek Greg Strause yönetmeni ile aynı filmlerde yönetmen olan kişilerce yapılmaktadır?
Director to Movie to Director to Nationality	731	Zeynel Doğan yönetmeninin yönettiği filmlerin yönetmenleri hangi uyruktan gelmektedir?
Movie to Actor to Movie to Release Year	10010	A Million Little Pieces filmi içerisinde yer alan oyuncuların diğer oynadığı filmlerin yayınlanma yılı ne zamandır?
Movie to Actor to Movie to Language	10012	Mr. Bean Tatilde filminde rol alan kişilerin oynadığı filmlerde hangi dil konuşulmaktadır?
Movie to Actor to Movie to Genre	10012	Ip Man filminin oyuncularının oynadığı filmlerin içerisinde hangi kategoriler bulunmaktadır?
Movie to Actor to Movie to Director	10012	Hangi yönetmenler Yılanların Öcü filmi oyuncularının oynadığı filmleri yönetmektedir?
Movie to Actor to Movie to Rating	7795	Hangi değerlendirme puanı Rocky 4 filmi oyuncularının oynadığı filmler için verilmiştir?
Movie to Director to Movie to Release Year	7322	Hangi yıllar içerisinde Oslo, 31 Ağustos filminin yönetmeninin yönettiği filmler yayınlanmaktadır?
Movie to Director to Movie to Language	7322	Hangi dil Zor Ölüm 2 filmini yöneten kişinin yönettiği filmlerde kullanılmıştır?
Movie to Director to Movie to Genre	7324	Siyah Giyen Adamlar 3 filminin yönetmeninin diğer yönettiği filmlerde hangi konular yer almaktadır?
Movie to Director to Movie to Actor	7324	Hangi oyuncu Pearl Harbor filmi içerisinde yönetmen olarak rol alan kişinin filmlerinde oynamıştır?
Movie to Director to Movie to Rating	3858	Pasifik Savaşı filmi içerisinde yer alan yönetmenlerin diğer yönettiği filmlerin değerlendirmesi 5 üzerinden kaçtır?

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