information

Article

A Quick Prototype for Assessing OpenIE Knowledge Graph-Based Question-Answering Systems

Giuseppina Di Paolo †, Diego Rincon-Yanez *,† and Sabrina Senatore †

Department of Information and Electrical Engineering and Applied Mathematics, University of Salerno, 84084 Fisciano, Italy; giusidipaolo@gmail.com (G.D.P.); ssenatore@unisa.it (S.S.)
* Correspondence: drinconyanez@unisa.it
† These authors contributed equally to this work.

Abstract: Due to the rapid growth of knowledge graphs (KG) as representational learning methods in recent years, question-answering approaches have received increasing attention from academia and industry. Question-answering systems use knowledge graphs to organize, navigate, search and connect knowledge entities. Managing such systems requires a thorough understanding of the underlying graph-oriented structures and, at the same time, an appropriate query language, such as SPARQL, to access relevant data. Natural language interfaces are needed to enable non-technical users to query ever more complex data. The paper proposes a question-answering approach to support end users in querying graph-oriented knowledge bases. The system pipeline is composed of two main modules: one is dedicated to translating a natural language query submitted by the user into a triple of the form <*subject*, *predicate*, *object*>, while the second module implements knowledge graph embedding (KGE) models, exploiting the previous module triple and retrieving the answer to the question. Our framework delivers a fast OpenIE-based knowledge extraction system and a graph-based answer prediction model for question-answering tasks. The system was designed by leveraging existing tools to accomplish a simple prototype for fast experimentation, especially across different knowledge domains, with the added benefit of reducing development time and costs. The experimental results confirm the effectiveness of the proposed system, which provides promising performance, as assessed at the module level. In particular, in some cases, the system outperforms the literature. Finally, a use case example shows the KG generated by user questions in a graphical interface provided by an ad-hoc designed web application.

Keywords: question answering; knowledge graph; knowledge graph embeddings; knowledge base

check for updates

Citation: Di Paolo, G.; Rincon-Yanez, D.; Senatore, S. A Quick Prototype for Assessing OpenIE Knowledge Graph-Based Question-Answering Systems. *Information* **2023**, *14*, 186. https://doi.org/10.3390/info14030186

Academic Editor: Ryutaro Ichise

Received: 5 February 2023
Revised: 26 February 2023
Accepted: 7 March 2023
Published: 16 March 2023

1. Introduction

Over the past ten years, knowledge graphs (KG) have received a lot of interest, since they effectively organize and represent knowledge, allowing it to be used in many applications. Knowledge graph models have been widely used to arrange and describe data in various sectors [1], including medicine [2], health care [3], biology [4], and finance [5]. They are playing an increasingly critical role in many applications [1], such as drug discovery [2], user recommendation [6], dialog systems [7], and question answering [8,9].

Actual knowledge graphs usually include millions or billions of facts. Because of their vast volume and sophisticated data structures, non-technical users need help accessing the considerable and essential knowledge inside them. Question-answering systems gain attention from the scientific community [8]; they aim to automatically translate natural language inquiries from end users into structured queries, such as SPARQL, and return entities and predicates from the KG as replies. Knowledge graph-based question answering (KGQA) allows for overcoming the gap [9].

A question-answering (QA) system typically consists of a natural language question (the input), a knowledge base, the engine (agent), and an answer (the output). A knowledge base is typically structured as a knowledge graph [10], which offers a consistent way to express heterogeneous entities and concepts in the form of triples *(head entity, relation, tail entity)* (denoted as (h, r, t)). An increasing amount of attention is being directed to QA systems with the widespread of KGs in academia and industry. The introduction of large-scale open-domain KGs, such as Freebase [11], Wikidata [12], and DBpedia [13]; it can be said that KGs are the source that QA system mines to get answers.

The open information extraction (OpenIE) paradigm systems built by exploiting the OpenIE paradigm are noteworthy [14]; this paradigm allows for the extraction of open relations and arguments (in the form: *subject; relation; object*) from sentences, with no domain restriction and without requiring any previous training data. This technique enables a semantic representation via straightforward predictive statements [15] that is useful for extracting the relations from simple statements, such as questions (e.g., who directed the Star Wars movie?); so, they can be collected for a wide variety of downstream tasks, such as precise question answering. Thanks to the OpenIE paradigm, automatic annotation replaces the time-consuming and error-prone need for manual annotation [8]; the domain-independent design is possible with the rich representation of knowledge, a scalable solution for extracting facts and relationships from unstructured text, and represents a powerful tool for automated knowledge management and retrieval.

The paper introduces a KG-based QA system to allow users to formulate simple natural language questions. The system design comprises two modules: the first is REBEL [16], a seq2seq model built on BART [17] that performs end-to-end relation extraction for several relation types. Then, the second module represents a knowledge graph embedding model for fact prediction. The whole system accomplishes fast prototyping based on OpenIE principles: it leverages a transformer model, REBEL, which translates natural language sentences into triples to feed the second knowledge graph embedding module aimed at evaluating triples. A front-end web application allows users to interact with the system, returning answers in textual and graph-based representation when questions are submitted. The experimental results confirm the satisfactory overall performance of the framework, expressed in terms of F1-score, precision, recall, accuracy, MRR, and Hits@N on the MovieQA dataset (aka Wikimovies) [18]. Moreover, in some cases, the performance of the knowledge graph embedding model, TransE, exceeds the state of the art.

Although the system is based on the existing modules, it represents a fast prototype that allows for assessing the quality of answers, especially in the case of specific knowledge domains. Additionally, it helps hasten the creation of graph-based question-answering systems, which can be particularly helpful in fields such as healthcare or finance, where prompt and effective responses are required.

The remainder of this work is structured as follows: Section 2 provides an overview of knowledge graph-based approaches and related query-answering systems. Section 3 briefly introduces the preliminary concepts of KGs; then, Section 4 presents the proposed methodological approach motivating them from a traditional viewpoint. Then, Section 5 describes the implementation details and the proposed method's evaluation. Section 6 shows a use case of the proposed question-answering system through the primary interactions. Finally, the conclusions and the future work are highlighted in Section 7.

2. Related Work

Knowledge graphs harvest, organize, and efficiently manage knowledge from massive amounts of data, in order to increase the quality of information services and provide smarter services to consumers. All of these aspects rely on knowledge reasoning over KGs, making it one of the most important technological aspects [19]. Finding errors and drawing new conclusions from existing data are the objectives of knowledge reasoning over knowledge graphs. Through knowledge reasoning, new relationships between entities may be derived, which can subsequently be used to enhance knowledge graphs and enable sophisticated applications.

Especially, *knowledge graph embedding* (KGE) techniques have recently attracted a lot of interest, since they can learn the representations (i.e., embeddings) of entities and relations in low-dimensional vector spaces [1], and these embeddings may be utilized as features to enable link prediction, entity categorization, and question answering. In particular, research on the question-answering domain, finds large-scale knowledge graphs, a powerful tool for building robust and efficient question-answering systems, especially in the field of natural language processing. This way, to answer questions by traversing entities and relations in a knowledge graph, needs to convert the natural language to a logical query.

2.1. Knowledge Graph Construction

Natural language processing (NLP) tasks play an important role in KG construction. Recent research finds relations from sentences by exploiting a graph neural network for sentence representations and aligning them to the KG [20]. Tasks such as named entity recognition (NER) [21] and relation extraction (RE) [16,22] aims at extracting structural information at the sentence, bag, and document levels [23], also exploiting a description-based representation of entities [24], as well as ontologies [25] starting from unstructured texts. Identifying and extracting accurate and comprehensive information from unstructured data could be a challenging task for knowledge construction. The OpenIE paradigm allows for the automated extraction of structured knowledge, without the need for pre-defined schemas, ontologies, and evolving knowledge bases or manual annotation [8].

Additionally, prompt tuning for few-shot classification tasks has gained some attention in enhancing knowledge extraction [26]. However, most of the recent literature on KGs focuses on particular facets of KGs, consisting of their construction [27] and embedding [28].

Despite the fact that KGs contain a significant number of entities and relations, link prediction is a critical issue for knowledge graphs because they are typically incomplete. Indeed, large-scale KGs, such as NELL [29], Freebase [11], WordNet [30], and YAGO [31], have emerged as important sources of auxiliary information for many AI-related tasks, such as question answering, report extraction, and recommendation systems [8,28].

The learning of low-dimensional representations of entities and relations for missing link prediction has recently been the subject of extensive research on topics such as completeness, partiality, and newly-added information [32]. At the same time, since generated knowledge graphs can contain millions of entities and relationships, embedding them all in a low-dimensional space might be computationally demanding, making scalability difficult [33].

In terms of completeness, some academic trends have centered on link prediction, which seeks to detect missing facts in KGs.

Knowledge graphs can be sparse, and many entity pairs lack any known relationships. Due to this sparsity, it may be challenging for embedding models to properly represent and then predict the relationships between entities [34].

Existing approaches for link prediction are known as knowledge graph embedding models (KGE) [24,35]. To retain the basic structure of the KGs, KGE models learn the embedded representation of both the relations and the entities, preserving the inherent structure of the KGs. Predicting missing links with knowledge graph embedding (KGE) methods has been extensively investigated in recent years. The general methodology is to define a score function for the triple [36–38]. Geometric properties are used in several KG embedding techniques. Some improvements have been gained by utilizing either more complicated spaces (e.g., moving from Euclidean to complex or hyperbolic space) [36] or more advanced procedures (e.g., from translations to isometries or to learning graph neural networks) [38]. Other alternative strategies have progressed in both directions [39].

Nevertheless, knowledge graph embedding models are typically trained on a specific knowledge domain [1,2], and may not generalize well to other knowledge graphs or new entities and relationships that are not present in the training data [1]. Moreover, KGE models are not able to handle complex queries with multiple entities and relations. In [40], the method called LEGO creates a latent execution-guided reasoning framework that infers a question's latent structure and reasons in the latent space for multi-hop logical inference.

2.2. Knowledge Graph-Based Question Answering

Knowledge graph-based question answering (KGQA) allows for answering questions by exploring facts in a knowledge graph.

Traditional QA methods [41] parse the question and synthesize the query using hand-engineered templates. However, these approaches require extensive domain expertise in order to manually create the set of rules and characteristics that will limit the search area.

Some end-to-end question-answering (E2EQA) models overcome this annotation bottleneck since they exploit question-answer pairs that only need weak supervision. They learn to predict paths in a knowledge network using only the answer for the training. In [42], indeed, differentiable knowledge graphs are used as a technique to describe KGs as tensors and queries as differentiable mathematical operations, in order to train the model in a completely differentiable manner. In addition to the E2EQA models based on single-entity questions, a E2EQA system based on multiple-entity questions and intersection learning in a dynamic multi-hop environment is proposed in [43]. Intersection models learn to follow relationships and intersect the generated sets of entities to arrive at the right answer.

Recent KGQA approaches [44,45] use deep networks to accomplish neural semantic parsing, in order to avoid the rule requirement. These methods, however, require ground truth queries for supervision, which suggests some human work.

KGQA provides a method for artificial intelligence systems to leverage knowledge graphs as a key component to respond to human questions, with applications ranging from search engine design to conversational agent construction. Indeed, a research trend on KGQA retrieves information from the KG using RL agents [46] or graph nets [47]; these approaches rely even more on text corpora to improve their performances [48].

Anyway, QA systems based on knowledge graphs inherited KG-based issues, such as poor accuracy due to the incompleteness of the knowledge graph or limited coverage of knowledge domain, and finally, ambiguity in question understanding, which requires accurate NLP tasks to contextualize the domain where the question is placed properly. [49]. Improvements in the performance of QA systems may depend on the structured representation of natural language sentences. For instance, in [50], in the training supervision, the ground truth was extracted from the given tabular databases, whereas in other approaches, such as [51], which used unstructured text understanding, a reading comprehension and sequence-to-sequence translation [52] was achieved.

In KBQA systems with embedding techniques, noteworthy is EmQL [53], a query embedding approach that employs set operators, although these operators must be taught for each KB. TransferNet [54] is a model that trains KGQA in a differentiable manner; however, because facts are stored as a $N \times N$ matrix, where N represents the number of entities, its scalability concerns bigger knowledge graphs. The effectiveness of knowledge graph embedding models [36–38] in different real-world applications [55] prompted an investigation into its possible usage in resolving the KGQA issues. In the recent literature, there are many KGQA systems that achieve a state-of-the-art with enhanced KGE models [40,44], but the incompleteness in answering and domain coverage push researchers to continue to develop new methods to address these challenges and improve the performance of knowledge graph-based question-answering systems.

Evaluating knowledge-based query-answering systems [56] is an active area of research. Our approach seeks to intercept this line of research by proposing a fast prototype to accelerate the development of graph-based question-answering systems, which can be particularly useful in domains where there is a need for fast and efficient answers, such as health care or finance. Reusing existing tools to avoid reinventing the wheel, as the literature provides good open-source tools, allows for focusing solely on the actual goal of evaluating the performance using specific KGE models or in a specific domain. Our approach represents a simple attempt to build a fast prototype of a KGQA system aimed at testing its effectiveness in a specific domain.

3. Preliminaries on Knowledge Graphs

Knowledge graphs (KGs) are typically structured knowledge bases, used to represent the objective world's concepts, entities, and relationships. KGs organize, handle, and comprehend enormous amounts of data in a manner similar to human cognitive reasoning. A KG is used in a variety of downstream applications, including semantic search, intelligent recommendation, and question-answering systems, because of its rich semantic content and clear logical structure [10].

A knowledge graph (KG) is a directed heterogeneous multigraph composed of nodes and edges, where a node represents an entity or an abstract concept, while an edge is an attribute of an entity or a link between two entities. The information may be encoded using KGs in a way that is both understandable by humans and adaptable to machine analysis and inference. KGs represent many kinds of information as entities connected by relations. More formally, given a set of entities E and a set of relations L, a knowledge graph G consists of a set of triples K such that $K \subseteq E \times L \times E$. A triple is represented as (h, l, t), with h, $t \in E$ denoting the subject and object entities, respectively, and $l \in L$ being the relation between them.

The triples combine to create a directed graph, where the nodes represent entities and the edges represent predicates. In a knowledge graph, each triple denotes a single fact or piece of information. In a graph triple node-edge-node, there are two main edge roles: one as a property associated with an entity: "Blade Runner, release_year, 1982, and another as a connection between two entities: "Blade Runner, directed_by Ridley Scott" (Figure 1).

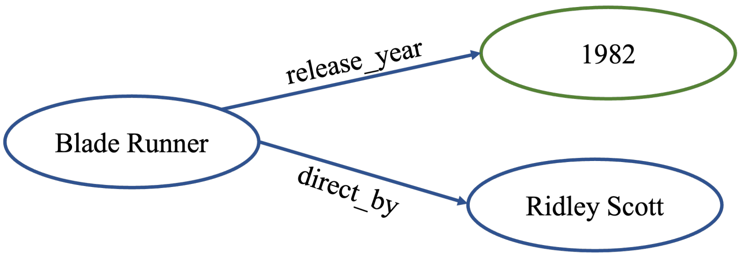

Figure 1. A triple example with a *property* edge "release_year" and a *relation* edge, "direct_by".

A knowledge graph is a large-scale semantic network. According to the kinds of knowledge contained, the knowledge graph may be characterized as a domain knowledge graph or an open knowledge graph. Domain knowledge focuses on information on a certain topic and usually comprises more professional and precise knowledge in a narrow interpretation. Open knowledge graphs are made available online so that the general public can access their information, and they have also been published within specific domains. Examples of open KGs are DBpedia [13], Freebase [11], WordNet [30], and YAGO [31].

4. The Approach

The proposed framework aims to assist the users in getting an answer to the question submitted in natural language. To achieve this, the question will be processed to extract the sentence's main components into triples (h, r, t): (1) a question marker, which could be one of the 5W (i.e., *who, what, where, which, and when*), (2) an action, and (3) an entity, which can be the subject or the object performing the described action.

The system design consists of two main modules: the **NLP module**, which performs the triple extraction, and the **KGE module**, which executes the triple evaluation. These modules are arranged sequentially (Figure 2). The *NLP module* uses a transformer model, called REBEL [16], to translate natural language sentences into triples, with an end-to-end approach that takes unstructured textual input and generates structured output that is compliant with a given vocabulary. REBEL is a BART-based transformer, fine-tuned on common-sense relation extraction datasets, such as the news-based: NYT-Multi [57] and CONLL04 [58], document level: DocRED [59], and pharmaceutical ADE [60]. The KGE module, instead, queries a pre-trained knowledge base, using knowledge graph embedding (KGE) models to compare the vector representations of the triples taken as input to its internal representation. Additional details about the introduced modules are given as follows.

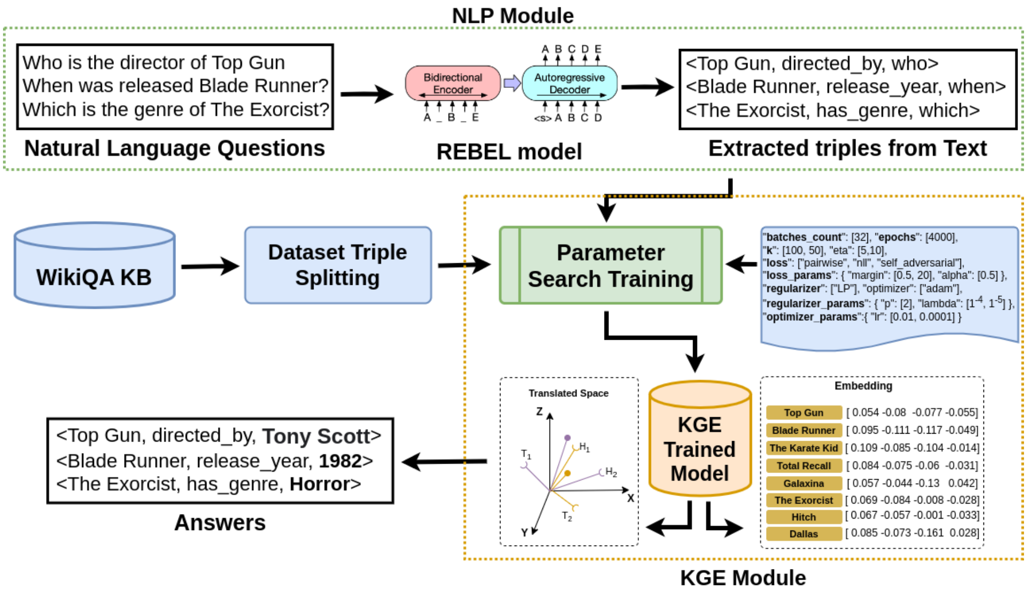

Figure 2. Overview of the proposed system: The first module, REBEL, takes a natural language question as input. The textual question is then processed by REBEL, which returns a <s,p,o> triple, whose object (o) represents one of the five possible question types. The extracted triple is given as input to the KGE module. The KGE model interprets relationships as translated operations on low-dimensional embeddings. Once the translated space is obtained, the model compares the distances between *subject* + *predicate* with *object* embedding features to verify if *subject* + *relation* = *object*. If the distance is lower than a threshold, the fact is considered reliable and true, and the triple has been completed successfully. The output given by TransE consists of the triple containing the correct answer.

4.1. NLP Module

REBEL is a seq2seq model built on BART [17] that performs end-to-end relation extraction for several relation types. The goal is to translate raw input sentences into a set of triples. This model tackles relation extraction and classification as a generation task, similar to a "translation". It is based on the teacher forcing method widely used in RNN [61] for translation tasks. It leverages text pairs in two languages by conditioning the decoded text on the input. At training time, the encoder receives the text in one language, and the decoder gets the text in the other language, outputting the prediction for the next token at each position. A similar mechanism is used in REBEL: a raw input sentence containing entities that are translated with implicit relations between them into a set of triplets that explicitly refer to those relations. The triplets must, therefore, be expressed as a series of tokens for the model to decode. The model uses a reversible linearization with special tokens that enable the model to output the relations in the text in the form of triplets while minimizing the number of tokens that need to be decoded.

In addition to REBEL, another framework, namely Seq2RDF [62], was considered. An encoder–decoder framework translates natural language sentences X to an RDF triple Y, whose entities and relationships comply with a certain KG vocabulary set. Moreover, it treats triples within a KG as an independent graph language. Compared to Seq2RDF, REBEL provides better performance, and combined with the KGE models, it reveals Seq2RDF's weaknesses in capturing sentence context (further details are given in the experiments) that REBEL can fix.

The module's input is a natural language question, for instance, $Q_1 =$ *Who directed Blade Runner?* Given the question Q_1, the module translates it into a triple $T_1 = (h, r, t)$ by leveraging the translation-oriented transformers to deliver end-to-end relation extraction from the input text.

REBEL takes raw text as input and outputs linearized triples. If our input sentence is Q_1 to x and the result of linearization of the relations in Q_1 is T_1 to y, then REBEL's task is to autoregressively generate y given x, as shown in [16]:

$$pBART(y|x) = \prod_{i=1}^{len(y)} pBART(y_i|y < i, x) \tag{1}$$

So, for example, given the question *Who directed Blade Runner?*, the module generates, as output, the triple $T_1 = \{BladeRunner, directed\_by, Who\}$.

4.2. KGE Module

Knowledge graph embeddings (KGEs) are supervised learning models that learn vector representations of labeled, directed multigraph nodes and edges. They learn low-dimensional representations of entities and relations to predict missing facts. Roughly speaking, given an incomplete knowledge base, one of the possible tasks is to predict unknown links. KGE models achieve this through a scoring function ϕ that assigns a score $s = \phi(h, r, t) \in \mathbb{R}$, which indicates whether a triple is true, with the goal of being able to score all missing triples correctly. The score function $\phi(h, r, t)$ measures the salience of a candidate triple (h, r, t). The goal of optimization is usually to assign a higher score to the true triples (h, r, t) than to corrupted false triples (h', r, t) or (h, r, t'). Let us remark that a triple one between the head and the tail can be corrupted (denoted by superscript).

The KGE is commonly used for link prediction; the task focuses on the missing part of a triple against a specific KB that was trained. In the low-dimensional embedding space, these KGE models are denoted by various score functions [63] that quantify the distance between two entities through the relation type, as shown in Figure 3. The KGE models are trained using these score functions, so those entities connected by relations are close to one another, and entities without connections are far away.

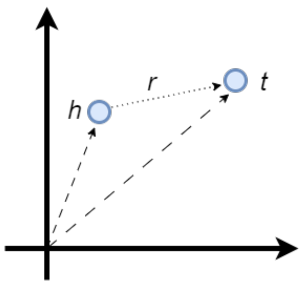

Figure 3. Embedding space representation of a single triple (h, r, t).

To exploit individual features of the KB and predict the answer from the triple generated by the OpenIE approach, these KGE models were taken into account:

- TransE [36]: It is an energy-based model for learning low-dimensional features of entities. It models relationships by interpreting them as translations acting those low-dimensional embeddings of the entities. The key feature of this model is how well it can automatically add new facts to multi-relational data without the need for additional knowledge.
- DistMult [37]: It forces all the embeddings into diagonal matrices, reducing the dimensional space and transforming the relation into a symmetric one. This makes

it unsuitable for general knowledge graphs, since it only uses a diagonal matrix to represent the relationships.

- ComplEx [38]: It handles symmetric and antisymmetric relations, using complex embeddings (real and imaginary parts) involving the conjugate-transpose of one of the two vectors. ComplEx embedding facilitates joint learning of subject and object entities, while preserving the asymmetry of the relation. It uses the Hermitian dot product of embedding subject and object entities. Complex vectors can successfully encapsulate antisymmetric connections, while retaining the efficiency benefits of the dot product, namely linearity in both space and time complexity.

These knowledge graph embeddings are some of the most popular models for various tasks, such as link prediction, entity classification, and relation extraction; these three models are chosen for performance comparison at the link prediction level, even though they are simple and effective and widely-used as baselines in comparative studies with newer models [63]. These embeddings use the same dimensional space to represent entities and relations and are basically translation models, since they represent the relationship between two entities as a translation vector in the embedding space. Moreover, all the models use a scoring function to assess the plausibility of a triple. Finally, they minimize an objective function that evaluates the discrepancy between the predicted triples and the actual triples in the knowledge base. These models are relatively simple compared to the newer models based on complex architectures, involving multiple layers of neural networks and advanced techniques, such as attention mechanisms, graph convolutional networks, and tensor factorization [63]. Finally, they provide a clear interpretation of the learned embeddings, compared with the newer ones that are harder to interpret, even though their prediction performance is more accurate [64]. Moreover, these models, namely TransE, DistMult, and ComplEx, still seem to be effective in our prototyping-oriented scenario.

Table 1 gives a synthetic view of the KGE main features, their respective scoring functions, and additional non-functional model properties.

Table 1. Selected models breakdown by type for the question-answering evaluation. h, l, t represent the head, relation, and tail of the triple, respectively, while the W means the parameter matrix and K the rank of the matrix.

Property	TransE	DistMult	ComplEx
Scoring Function	$-\|e_l + r_h - e_t\|_n$	$\{r_l, e_h, e_t\}$	$Re(\sum_{k=1}^{K} W_l, e_h, \overline{e_t})$
Type	Translational	Bilinear	Negative Log
Family	Geometric	Matrix Factorization	Matrix Factorization
Interpretability	High	Medium	Low
Performance	Low	Medium	High
Complexity	Low	Medium	High

In particular, the table introduces some features that are taken into account based on empirical evidence. The *interpretability* of the three models is different: while TransE is more intuitive with respect to DistMult, which is based on matrix factorization while ComplEx relies on complex numbers over high-dimensional spaces. They can extract more complex relations (symmetric and complex ones), with consequently better *performance* just as is the case of ComplEx. Finally, at the *complexity* level, the TransE model is quite simple, involving a small number of parameters, so this makes it computationally efficient and easy to train; contrarily, ComplEx and DistMult are more complex models that involve a larger number of parameters and can be computationally expensive to train.

The modeling method generally consists of identifying local or global connectivity patterns between entities and then predicting the observed relationships between a specific entity and all others using these patterns. The notion of locality for a single relationship may be purely structural, but it may also depend on the entities (e.g., those who liked Star Wars IV also liked Star Wars V, but may or may not like Titanic).

The difficulty with relational data is that the locality may involve relationships and entities of different types simultaneously, so modeling multi-relational data requires more general approaches that can choose appropriate models by considering all heterogeneous relationships simultaneously. For example, in TransE, relationships are represented as translations in the embedding space. Suppose *(h, r, t)* holds. In that case, the embedding of the tail entity *t* should be close to the embedding of the head entity *h*, plus some vector that depends on the relationship *l*, as shown in Figure 3.

In more formal terms, the model learns vector embeddings of the entities and relationships for each training set S of triples (h, r, t) composed of two entities $h, tinE$ (the set of entities) and a relationship $rinL$. The embeddings take values in $\mathbb{R}^N$, where N is the space dimensionality. TransE considers the translation of the vector representations, i.e., the head entity embedding should be close to the tail entity embedding, plus relation embedding, when the head entity is similar to the tail entity (when (h, r, t) holds). Otherwise, the head entity should be far away from the tail entity. According to the energy-based framework proposed in [65], the energy of a triple is equal to $d(h + l, t)$ for some dissimilarity measure d, which can be either the L1 or the L2-norm.

To learn such embeddings, the model minimizes a margin-based ranking criterion over the training set described in the following equation:

$$\sum_{(h,r,t)\in S} \sum_{(h',r,t')\in S'_{h,r,t}} [\gamma + d(h + r, t) - d(h' + r, t')]_+ \tag{2}$$

where:

- $[x]_+$: denotes the positive part of x;
- $\gamma > 0$: is a margin hyper-parameter;
- $S'_{h,r,t} = \{(h', r, t') \mid h' \in E\} \bigcup \{(h, r, t') \mid t' \in E\}$

The set of corrupted triplets $S'_{h,r,t}$ is composed of training triplets with either the head or tail replaced by a random entity (but not both simultaneously). The loss function (2) favors lower energy values for training triples than for corrupted triples and is, thus, a natural implementation of the intended criterion.

To summarize, once triplets such as $T1 = (Blade\ Runner, directed\_by, who)$ are generated by the *NLP module*, the idea is to leverage the information in an existing KB by using suitable models that will allow for representing the triple in an embedding space. The module's implementation and experimentation are described in Section 5, and the models described in Table 1 have been tested and queried.

5. Experimentation

To validate the proposed approach, a dataset for question answering was chosen and processed to perform a link prediction task. Initial testing was performed on REBEL, the pre-trained BART-based transformer that was employed to convert the question from natural language into a triple using the OpenIE paradigm. After that, the KGE models are trained on the chosen KB, in order to compare, query, and compute the experiment scoring results.

The metrics used for performance evaluation of the whole framework are recall, precision, F1-score, MRR, and hits@N (with N ∈ { 1, 3, 10 }).

5.1. Dataset

The MovieQA (WikiMovies) dataset [18] is a question-answering pair dataset built by Wikipedia that includes the raw source text and the corresponding KB framed in the movies domain. The dataset comprises three forms of knowledge representation: (i) raw Wikipedia *documents* describing movies; (ii) a classical graph-based *KB* consisting of entities and relations created from the Open Movie Database (OMDB) and MovieLens (disclosed in separate files); and (iii) information extracted (*IE*) by processing the Wikipedia pages to build a KB. The dataset matches the query with the three knowledge types mentioned.

The experiments were conducted on a dataset with a graph-based KB representation. This dataset is made up of $T = < h, r, t >$ triples that correspond to the structure (film, relation, object). The KB dataset holds information on 18 thousand movies using OMDb and MovieLens metadata, with entries for each movie and nine different relation types: director, writer, actor, release year, language, genre, tags, IMDb rating, and IMDb votes, with 10k related actors, 6k directors, and 43k entities in total, in which 75k+ are distinct entities, on ten structured relations, reaching over 186 thousand triples.

Only the triples where the entities also appear in the Wikipedia articles are kept to ensure that all QA pairs have an equal chance of receiving an answer from either the KB or Wikipedia document sources. However, the original triple KB has been modified to ensure proper processing by our framework, extracting the individual <subject, predicate, object>; as shown in Figure 4, each original triple is split into as many triples as there are tail entities. The triples reached over 376 thousand statements; the final dataset was published in GitHub [66].

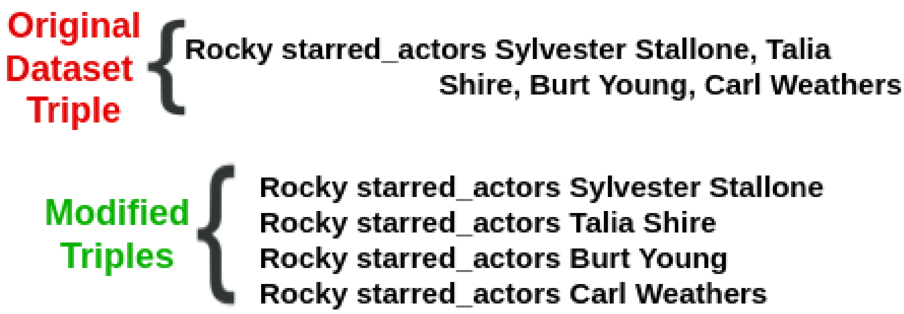

Figure 4. Example of a triple from the original MovieQA dataset with the modified new single triples.

As stated, REBEL allows for transforming a natural speech question into a triple ($< h, r, t >$). More specifically, REBEL is a transformer based on BART that performs end-to-end relation extraction for more than 200 different relation types. It uses the OpenIE paradigm to translate a raw input phrase containing entities with implicitly stated relations and produces explicit triples expressing the relationships between the entities.

5.2. Question Triple Translation with REBEL

REBEL was trained to predict up to 220 different relationship types. For this particular dataset, and given the nature of the relations, a post-triple-generation normalization task was taken into account. So, for instance, by considering the natural question *"Who directed American Gigolo?"*, according to the 5W question markers, the sentence is translated into a triple by substituting the object (head) or the subject (tail) with the question mark. In this case, the final triple generated by the REBEL decoder is $<AmericanGigolo, directed\_by, Who>$. REBEL was prompt-tuned, even though the input component has been revised to enable the processing of our dataset (placeholders defined in the returned text have been removed to guarantee the proper translation of input questions). REBEL was tested on 20% of the Wikimovies dataset (≈27.600 questions), downloaded directly from the public repository. Figure 5 shows the performance of REBEL, expressed in terms of recall, precision, and F1-score, for each component and the overall triples.

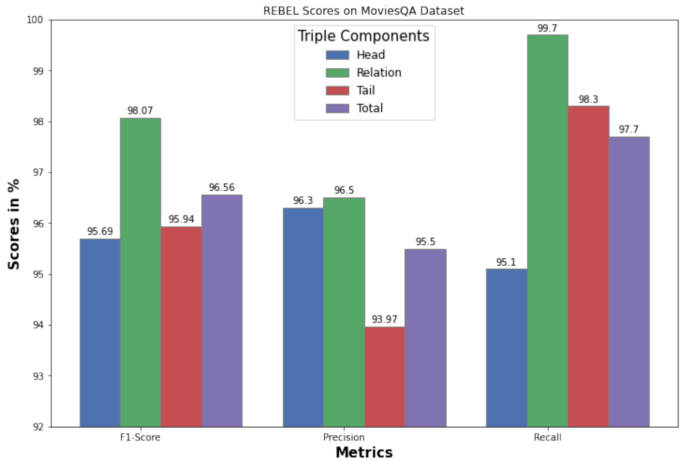

Figure 5. REBEL model validation score using 20% in the WikiMovies dataset, the scores show evaluation on each triple component: <subj, pred, obj> and an overall metric, in terms of Recall, Precision, F1-score.

As previously introduced, REBEL was selected after some experiments were carried out on Seq2RDF [62], an encoder-decoder framework (see Section 4). It consists of an encoder taking in a natural language sentence, a sequence input, and a decoder generating the target RDF triple. Encoders and decoders are designed as recurrent neural networks with long short-term memory (LSTM) cells. The attention mechanism is applied to force the model to learn to focus on specific parts of the input sequence when decoding, instead of relying only on the last hidden state of the encoder. Although Seq2RDF is structured and trained to extract URI related to entities of a text, in this approach, it was used to extract just the triplets concerning the <subject, predicate, object> of a given text or question. For this reason, the Seq2RDF architecture has been modified to predict just the subject, predicate, and object from a text and not search for the corresponding URIs.

Seq2RDF has been trained on 90% of the dataset and tested on 10% of it (≈10.380 questions). The experiments highlighted a clear issue with Seq2RDF: it was not effective enough to capture the whole context of the sentence. Indeed, the F1-score was 0.98 on predicates, 0,15 on subjects, and 0,08 on objects, only on the training set. The scores on the test set were still 0,97 on predicates, 0,11 on subjects, and 0,06 on objects. The network was unable to learn, not only to generalize. This is because the dataset's sentences almost always have their subjects and objects inverted, which makes it difficult for the network to correctly identify the subject and object.

For this reason, REBEL was chosen for sentence translation in the prototype, based on which the transformer was able to better generalize and correctly classify the subjects and objects in sentences and questions, even if these are interchangeable.

5.3. KGE Evaluation

The training on the selected dataset was performed on the three KGE models introduced in Section 4, namely *TransE*, *DistMult*, *ComplEx*; the metrics considered were the mean reciprocal rank (MRR) and HITS@N.

$$\mathbf{MRR} = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i} \in [0,1] \tag{3}$$

Precisely, MRR, defined in Equation (3), is a statistical measure for evaluating any process that produces a list of possible responses to a sample of queries (in this case, triples), ordered by the probability of correctness. More formally, the mean reciprocal rank is the average of the reciprocal ranks of results for a sample of triples Q; where $rank_{qi}$ refers to the rank position of the first relevant document for the i^{th} query q. It represents the ability of the model to produce a correct answer.

$$\textbf{HITS@N} = \frac{1}{|Q|}\sum_{i=1}^{|Q|} |q_i \in Q : rank_i < N| \in [0,1] \tag{4}$$

More formally, the *Hits@N* is defined in Equation (4) and represents the percentage rate of the original triples ranked at the top N in the prediction. A HITS@N with $N = 1$ is the conventional accuracy; the model prediction (the one with the highest probability) must be exactly the expected answer. It captures the fraction of triples that appear in the first N triples of the sorted rank list, or in other words, it calculates the percentage of examples for which the predicted label matches the specific target label. Different values of hits have been calculated, in particular: HITS@1, HITS@3, HITS@10. This measure represents the ability of the model to produce a correct answer in the top 1, 3, or 10 answers produced, respectively.

After performing a fine-tuning process on these models, the final result emphasized the fact that TransE overcomes the other two models; the performance of the training process is reported in Figure 6, the validation process is reported in Figure 7, and in Table 2, the best hyperparameter setting is given. Let us notice that careful fine-tuning of the hyper-parameters was accomplished, resulting in a Hits@1 of 85.7%, far higher than the one presented in [67], which obtains 25% on the same WikiMovie-300K dataset.

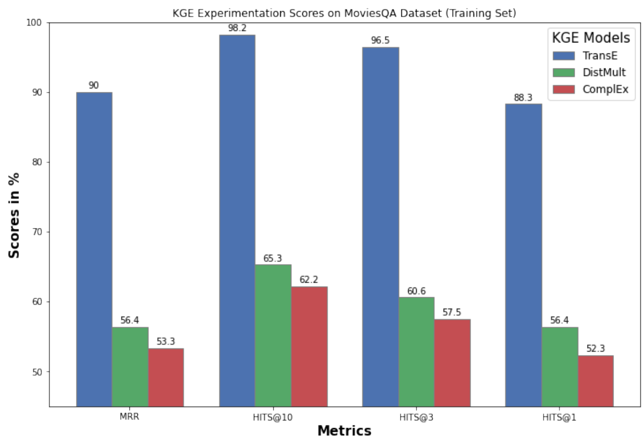

Figure 6. KGE Training Performance-Comparison between TransE, DistMult, and CompEx models on the modified triples in the WikiMovies dataset, measured in terms of MRR and HITS@N.

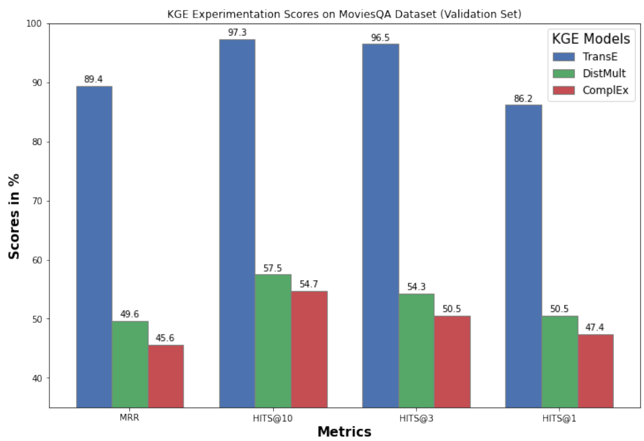

Figure 7. KGE Validation Performance-Validation values a comparison among TransE, DistMult, and CompEx models on the modified triples on the WikiMovies dataset, measured in terms of MRR and HITS@N.

Table 2. Hyper-parameter setting for TransE model.

Parameter	Value
batches count	32
seed	0
epochs	200
k	100
eta	100
regularizer	LP
optimizer	adam
Regularizer	p2
Regularizer	lambda 1^{-5}
Optimizer:lr	0.002
negative corruption entities	batch
loss	self adversarial
loss params: margin	10
loss params: alpha	0.001

Finally, the evaluation of the overall approach, composed of two modules, (1) the REBEL module and (2) the KGE module, is given in Table 3. The results show the outperformance of the *TransE* model over the *DistMult* and *ComplEx* models, despite the effectiveness of these two over the TransE model. This is because the TransE model requires less computation to extract from the KB better representative features. Furthermore, this result is consistent with the nature of the KB structure, which is composed of one-to-many and many-to-one relationships, rather than a highly connected graph for many-to-many.

Table 3. Results of the evaluation using REBEL combined with each embedding model.

Model	MRR	Hits@1	Hits@3	Hits@10
REBEL + TransE	**88.2**	**85.7**	**96.3**	**98.4**
REBEL + DistMult	41.7	40.6	47.4	41.7
REBEL + ComplEx	43.7	43.2	45.4	49.4

6. The System at Work: A Use Case

The question-answering system accomplished by the integration of the two used modules TransE and REBEL, supports as a back-end for a simple web application, designed to facilitate user interaction. Given the incompatibility of two used models, TransE and REBEL, based on the libraries Tensorflow 1.x and Pythorch 2.x, respectively, results in the need to use a container-based design to guarantee the communication between the modules.

The web application interface enables the formulation of a simple question in natural language, and the return of an answer graphical representation as triple is located in the knowledge graph. Moreover, additional code and data resources were published in a GitHub repository (GitHub Repository | available online: https://github.com/d1egoprog/FastKGQA (accessed on 23 February 2023)).

In Figure 8, the snapshots show the basic steps of the user interaction. In the first step, the user web application interface shows a portion of the knowledge graph created (Figure 8-Step 1) in modified triples by the dataset (as described in Section 5); these are stored using *NEO4J*, an industry-level graph database management system.

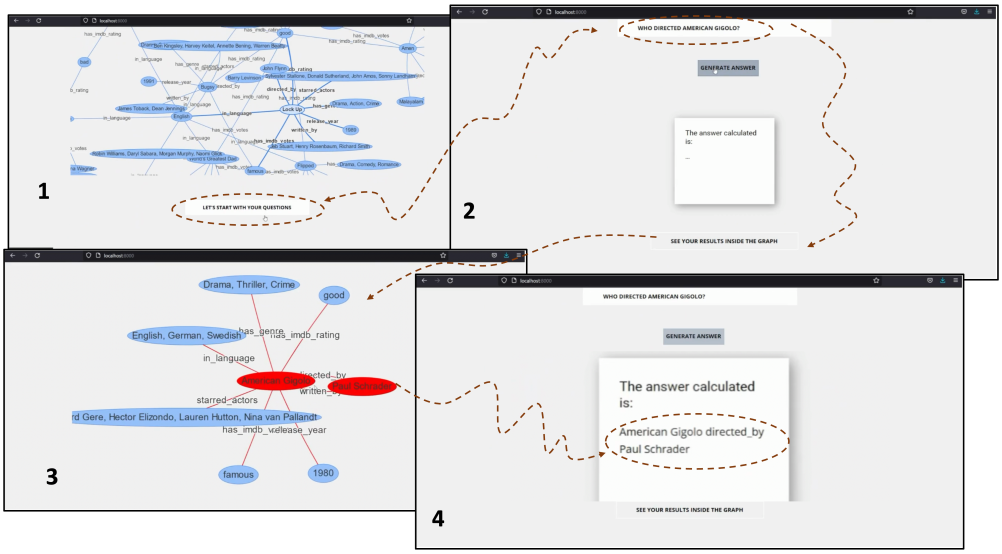

Figure 8. The system at work: initially, the user can submit a question (1) by clicking on *Let's start with your question* on the web application GUI; then, the question *Who directed "American Gigolò"?* is submitted (2), and the corresponding graph is shown (along with the graph nodes colored) (3); finally, the answer is returned in the interface, as shown in (4).

Initially, the *Let's start with your question* button at the bottom of the interface allows the user to submit a question (Figure 8-Step 1). The user, by clicking on that button, enables a new interface where the question can be entered (Step 2). In the example shown in Figure 8, the question submitted in natural language is "Who directed American Gigolo?". Then, the user can click the *Generate Answer* button to obtain the answer in a textbox provided in triple format (Step 4). On a more technical level, the user click triggers the REBEL module, which transforms the question into a triple that is input to the TransE module. TransE predicts the final answer by placing the embedding associated with the triple generated by REBEL in TransE's vector space. While the system elaborates the right answer, by clicking *See your results inside the graph* button, the triple generated by REBEL is also translated into cypher language and submitted to Neo4J, which graphically visualizes the relative portion of the graph containing the answer.

More specifically, the subgraph describing the nodes representing the question entities and related entity-nodes is shown; in particular, the nodes involved in the answer are colored in red (step 3): the node labeled "American Gigolo" is in the middle, surrounded by linked nodes, and one of them, labeled "Paul Schrader", is also colored and represents the answer.

Let us remark that the database used for visualizing the graph was originally stored in Neo4j. The dump of the database from Neo4j was embedded inside the Docker container, making the system portable, as well.

7. Conclusions

The paper proposes a system for simple question answering, based on two existing consolidated components: REBEL, a seq2seq model for transforming natural language questions into triple-based questions, and a KGE model that takes, as input, the REBEL generated triples, returns, and answers.

The system comprehensively provides good performance, particularly concerning the KGE model TransE: the combination of REBEL and TransE outperforms the Hit@1 performance of the state-of-the-art, considering the parameter settings shown in Table 2. The final use case shows the simplicity and effectiveness of the proposed system through the main interactions with the basic system interface, which shows the answer as both a triple and a (portion of) graph-based representation.

The proposed system is an off-the-shelf approach for fast prototyping design, leveraging OpenIE principles to build a pipeline composed of two KGQA-related modules from the literature. The REBEL module is, indeed, based on the OpenIE paradigm. Moreover, REBEL was prompt-tuned for the downstream task of triple extraction. It was tested on a new dataset to verify its effectiveness in returning high-quality triples from domain-specific questions. Additionally, the KGE-based module was trained and validated on TransE, DistMult, and ComplEx on the re-adapted triples in the MovieQA (WikiMovies) dataset (see Figure 4).

In a nutshell, the proposed system represents a fast prototype for:

- Simple question-answering that exploits existing tools from the literature.
- Leveraging on OpenIE principles to automatically extract structured information from natural language text, guaranteeing scalability, unsupervised learning, flexibility, accuracy, and integration with other natural language processing tools.
- Specializing the system to answer on a selected knowledge base, without retraining the question-triple translator model: in our case, REBEL was tested on a portion of Wikimovies without any pre-training.
- Assessing the quality of a fast composition design in question-answering effectiveness. Our prototypical system shows that the designed pipeline can overcome the state-of-the-art in some specific situations.

Question-answering systems may offer solid and accurate answers to various questions by combining the advantages of knowledge graphs with OpenIE, making them valuable tools for numerous applications, such as search engines, chatbots, and personal assistants. Achieving a fast prototyping of knowledge graph question-answering systems can help developers create effective systems by experimenting with new features and domains, thus enabling them to develop KGQA systems more quickly and at a lower cost, in terms of time and resources expended.

Author Contributions: Conceptualization, G.D.P., D.R.-Y. and S.S.; methodology, G.D.P., D.R.-Y. and S.S.; software, G.D.P. and D.R.-Y.; validation, G.D.P., D.R.-Y. and S.S.; writing, review and editing, G.D.P., D.R.-Y., and S.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: MoviesQA (Wikimovies) Dataset | Available online: https://metatext.io/datasets/movieqa (accessed on 23 February 2023). Modified Triples | Available online: https://github.com/d1egoprog/FastKGQA (accessed on 23 February 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Abu-Salih, B. Domain-specific knowledge graphs: A survey. *J. Netw. Comput. Appl.* **2021**, *185*, 103076. [CrossRef]
2. Bonner, S.; Barrett, I.P.; Ye, C.; Swiers, R.; Engkvist, O.; Hoyt, C.T.; Hamilton, W.L. Understanding the performance of knowledge graph embeddings in drug discovery. *Artif. Intell. Life Sci.* **2022**, *2*, 100036. [CrossRef]
3. Wang, M.; Zhang, J.; Liu, J.; Hu, W.; Wang, S.; Li, X.; Liu, W. PDD Graph: Bridging Electronic Medical Records and Biomedical Knowledge Graphs Via Entity Linking. In Proceedings of the Semantic Web–ISWC 2017: 16th International Semantic Web Conference, Vienna, Austria, 21–25 October 2017; Proceedings, Part II; Springer: Berlin/Heidelberg, Germany, 2017; pp. 219–227. [CrossRef]
4. Mohamed, S.K.; Nounu, A.; Nováček, V. Biological applications of knowledge graph embedding models. *Briefings Bioinform.* **2021**, *22*, 1679–1693. [CrossRef] [PubMed]
5. Day, M.Y. Artificial Intelligence for Knowledge Graphs of Cryptocurrency Anti-Money Laundering in Fintech. In Proceedings of the ASONAM '21: 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Virtual Event, The Netherlands, 8–11 November 2021; Association for Computing Machinery: New York, NY, USA, 2022; pp. 439–446. [CrossRef]
6. Wang, H.; Zhao, M.; Xie, X.; Li, W.; Guo, M. Knowledge Graph Convolutional Networks for Recommender Systems. In Proceedings of the WWW '19: The World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 3307–3313. [CrossRef]

7. Wu, S.; Wang, M.; Zhang, D.; Zhou, Y.; Li, Y.; Wu, Z. Knowledge-Aware Dialogue Generation via Hierarchical Infobox Accessing and Infobox-Dialogue Interaction Graph Network. In Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21, Montreal, Canada, 19–27 August 2021; Zhou, Z.H., Ed.; International Joint Conferences on Artificial Intelligence Organization: San Francisco, CA, USA , 2021; pp. 3964–3970. Main Track. [CrossRef]
8. Lukovnikov, D.; Fischer, A.; Lehmann, J.; Auer, S. Neural Network-Based Question Answering over Knowledge Graphs on Word and Character Level. In Proceedings of the WWW '17: 26th International Conference on World Wide Web, Perth, Australia, 3–7 May 2017; International World Wide Web Conferences Steering Committee: Republic and Canton of Geneva, Switzerland, 2017; pp. 1211–1220. [CrossRef]
9. Sabou, M.; Höffner, K.; Walter, S.; Marx, E.; Usbeck, R.; Lehmann, J.; Ngonga Ngomo, A.C. Survey on Challenges of Question Answering in the Semantic Web. *Semant. Web* **2017**, *8*, 895–920. [CrossRef]
10. Ji, S.; Pan, S.; Cambria, E.; Marttinen, P.; Yu, P.S. A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 494–514. [CrossRef] [PubMed]
11. Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; Taylor, J. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In Proceedings of the SIGMOD '08: 2008 ACM SIGMOD International Conference on Management of Data, Vancouver Canada, 9–12 June 2008; Association for Computing Machinery: New York, NY, USA, 2008; pp. 1247–1250. [CrossRef]
12. Vrandečić, D.; Krötzsch, M. Wikidata: A Free Collaborative Knowledgebase. *Commun. ACM* **2014**, *57*, 78–85. [CrossRef]
13. Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; Ives, Z. DBpedia: A Nucleus for a Web of Open Data. In *Lecture Notes in Computer Science*, *Proceedings of the 6th International Semantic Web Conference (ISWC), Busan, Republic of Korea, 11–15 November 2007*; Springer: Berlin/Heidelberg, Germany, 2007; Volume 4825, pp. 722–735. [CrossRef]
14. Kolluru, K.; Adlakha, V.; Aggarwal, S.; Mausam.; Chakrabarti, S. OpenIE6: Iterative Grid Labeling and Coordination Analysis for Open Information Extraction. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Barceló Bávaro Convention Centre, Punta Cana, Dominican Republic, 8–12 November 2020; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 3748–3761.
15. Martinez-Rodriguez, J.L.; Lopez-Arevalo, I.; Rios-Alvarado, A.B. OpenIE-based approach for Knowledge Graph construction from text. *Expert Syst. Appl.* **2018**, *113*, 339–355. [CrossRef]
16. Huguet Cabot, P.L.; Navigli, R. REBEL: Relation Extraction By End-to-end Language generation. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2021, Punta Cana, Dominican Republic, 16–20 November 2021; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 2370–2381. [CrossRef]
17. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 7871–7880. [CrossRef]
18. Miller, A.; Fisch, A.; Dodge, J.; Karimi, A.H.; Bordes, A.; Weston, J. Key-Value Memory Networks for Directly Reading Documents. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; Association for Computational Linguistics: Stroudsburg, PA, USA, 2016; pp. 1400–1409. [CrossRef]
19. Huang, X.; Zhang, J.; Li, D.; Li, P. Knowledge Graph Embedding Based Question Answering. In Proceedings of the WSDM '19: Twelfth ACM International Conference on Web Search and Data Mining, Melbourne, Australia, 11–15 February 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 105–113. [CrossRef]
20. Bastos, A.; Nadgeri, A.; Singh, K.; Mulang, I.O.; Shekarpour, S.; Hoffart, J.; Kaul, M. RECON: Relation Extraction Using Knowledge Graph Context in a Graph Neural Network. In Proceedings of the WWW '21: Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021; Association for Computing Machinery: New York, NY, USA, 2021; pp. 1673–1685. [CrossRef]
21. Gui, T.; Zou, Y.; Zhang, Q.; Peng, M.; Fu, J.; Wei, Z.; Huang, X. A Lexicon-Based Graph Neural Network for Chinese NER. 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), Hong Kong, China, 3–7 November 2019; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 1040–1050. [CrossRef]
22. Lin, Y.; Shen, S.; Liu, Z.; Luan, H.; Sun, M. Neural Relation Extraction with Selective Attention over Instances. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–12 August 2016.
23. Han, X.; Gao, T.; Yao, Y.; Ye, D.; Liu, Z.; Sun, M. OpenNRE: An Open and Extensible Toolkit for Neural Relation Extraction. 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): System Demonstrations, Hong Kong, China, 3–7 November 2019; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 169–174. [CrossRef]
24. Xie, R.; Liu, Z.; Jia, J.; Luan, H.; Sun, M. Representation Learning of Knowledge Graphs with Entity Descriptions. In Proceedings of the AAAI'16: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 2659–2665. [CrossRef]
25. Della Rocca, P.; Senatore, S.; Loia, V. A semantic-grained perspective of latent knowledge modeling. *Inf. Fusion* **2017**, *36*, 52–67. [CrossRef]

26. Chen, X.; Zhang, N.; Xie, X.; Deng, S.; Yao, Y.; Tan, C.; Huang, F.; Si, L.; Chen, H. KnowPrompt: Knowledge-Aware Prompt-Tuning with Synergistic Optimization for Relation Extraction. In Proceedings of the WWW '22: ACM Web Conference 2022, Online, 25–29 April 2022; Association for Computing Machinery: New York, NY, USA, 2022; pp. 2778–2788. [CrossRef]
27. Zhang, N.; Xu, X.; Tao, L.; Yu, H.; Ye, H.; Xie, X.; Chen, X.; Li, Z.; Li, L.; Liang, X.; et al. DeepKE: A Deep Learning Based Knowledge Extraction Toolkit for Knowledge Base Population. *arXiv* **2022**, arXiv:2201.03335.
28. Hogan, A.; Blomqvist, E.; Cochez, M.; D'amato, C.; Melo, G.D.; Gutierrez, C.; Kirrane, S.; Gayo, J.E.L.; Navigli, R.; Neumaier, S.; et al. Knowledge Graphs. *ACM Comput. Surv.* **2021**, *54*, 1–37. [CrossRef]
29. Carlson, A.; Betteridge, J.; Kisiel, B.; Settles, B.; Hruschka, E.R.; Mitchell, T.M. Toward an Architecture for Never-Ending Language Learning. In Proceedings of the AAAI'10: Twenty-Fourth AAAI Conference on Artificial Intelligence, Atlanta GA, USA, 11–15 July 2010; AAAI Press: Washington, DC, USA, 2010; pp. 1306–1313.
30. Fellbaum, C., Ed. *WordNet: An Electronic Lexical Database*; Language, Speech, and Communication; MIT Press: Cambridge, MA, USA, 1998.
31. Suchanek, F.M.; Kasneci, G.; Weikum, G. Yago: A Core of Semantic Knowledge. In Proceedings of the WWW '07: 16th International Conference on World Wide Web, Banff, AB, Canada, 8–12 May 2007; Association for Computing Machinery: New York, NY, USA, 2007; pp. 697–706. [CrossRef]
32. Jagvaral, B.; Lee, W.K.; Roh, J.S.; Kim, M.S.; Park, Y.T. Path-based reasoning approach for knowledge graph completion using CNN-BiLSTM with attention mechanism. *Expert Syst. Appl.* **2020**, *142*, 112960. [CrossRef]
33. Jain, N.; Tran, T.K.; Gad-Elrab, M.H.; Stepanova, D. Improving Knowledge Graph Embeddings with Ontological Reasoning. In Proceedings of the Semantic Web–ISWC 2021, Virtual, 24–28 October 2021; Hotho, A., Blomqvist, E., Dietze, S., Fokoue, A., Ding, Y., Barnaghi, P., Haller, A., Dragoni, M., Alani, H., Eds.; Springer International Publishing: Cham, Switzerland, 2021; pp. 410–426.
34. Chen, W.; Cao, Y.; Feng, F.; He, X.; Zhang, Y. Explainable Sparse Knowledge Graph Completion via High-order Graph Reasoning Network. *arXiv* **2022**, arXiv:2207.07503.
35. Chen, X.; Jia, S.; Xiang, Y. A review: Knowledge reasoning over knowledge graph. *Expert Syst. Appl.* **2020**, *141*, 112948. [CrossRef]
36. Bordes, A.; Usunier, N.; Garcia-Durán, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. In Proceedings of the NIPS'13: Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013; Curran Associates Inc.: Red Hook, NY, USA, 2013; pp. 2787–2795.
37. Yang, B.; Yih, W.t.; He, X.; Gao, J.; Deng, L. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015; p. 12.
38. Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, E.; Bouchard, G. Complex Embeddings for Simple Link Prediction. In Proceedings of Machine Learning Research, Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; Balcan, M.F., Weinberger, K.Q., Eds.; PMLR: New York, NY, USA, 2016; Volume 48, pp. 2071–2080.
39. Chami, I.; Wolf, A.; Juan, D.C.; Sala, F.; Ravi, S.; Ré, C. Low-Dimensional Hyperbolic Knowledge Graph Embeddings. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; Association for Computational Linguistics: Stroudsburg, PA, USA; pp. 6901–6914. [CrossRef]
40. Ren, H.; Dai, H.; Dai, B.; Chen, X.; Yasunaga, M.; Sun, H.; Schuurmans, D.; Leskovec, J.; Zhou, D. LEGO: Latent Execution-Guided Reasoning for Multi-Hop Question Answering on Knowledge Graphs. In Proceedings of Machine Learning Research, Proceedings of the 38th International Conference on Machine Learning, Online, 18–24 July 2021; Meila, M., Zhang, T., Eds.; PMLR: New York, NY, USA, 2021; Volume 139, pp. 8959–8970.
41. Bast, H.; Haussmann, E. More Accurate Question Answering on Freebase. In Proceedings of the CIKM 15: 24th ACM International on Conference on Information and Knowledge Management, Melbourne, Australia, 19–23 October 2015; Association for Computing Machinery: New York, NY, USA, 2015; pp. 1431–1440. [CrossRef]
42. Cohen, W.W.; Sun, H.; Hofer, R.A.; Siegler, M. Scalable Neural Methods for Reasoning With a Symbolic Knowledge Base. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
43. Sen, P.; Oliya, A.; Saffari, A. Expanding End-to-End Question Answering on Differentiable Knowledge Graphs with Intersection. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Punta Cana, Dominican Republic, 7–11 November 2021f; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 8805–8812. [CrossRef]
44. Lan, Y.; Jiang, J. Query Graph Generation for Answering Multi-hop Complex Questions from Knowledge Bases. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 969–974. [CrossRef]
45. Qiu, Y.; Zhang, K.; Wang, Y.; Jin, X.; Bai, L.; Guan, S.; Cheng, X. Hierarchical Query Graph Generation for Complex Question Answering over Knowledge Graph. In Proceedings of the CIKM '20: 29th ACM International Conference on Information and Knowledge Management, Virtual, 19–23 October 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 1285–1294. [CrossRef]
46. Qiu, Y.; Wang, Y.; Jin, X.; Zhang, K. Stepwise Reasoning for Multi-Relation Question Answering over Knowledge Graph with Weak Supervision. In Proceedings of the WSDM '20: 13th International Conference on Web Search and Data Mining, Houston, TX, USA, 3–7 February 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 474–482. [CrossRef]

47. Sun, H.; Dhingra, B.; Zaheer, M.; Mazaitis, K.; Salakhutdinov, R.; Cohen, W. 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, 31 October–4 November 2018; Association for Computational Linguistics: Brussels, Belgium, 2018; pp. 4231–4242. [CrossRef]
48. Lu, X.; Pramanik, S.; Saha Roy, R.; Abujabal, A.; Wang, Y.; Weikum, G. Answering Complex Questions by Joining Multi-Document Evidence with Quasi Knowledge Graphs. In Proceedings of the SIGIR'19: 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 105–114. [CrossRef]
49. Yani, M.; Krisnadhi, A.A. Challenges, Techniques, and Trends of Simple Knowledge Graph Question Answering: A Survey. *Information* **2021**, *12*, 271. [CrossRef]
50. Liang, C.; Norouzi, M.; Berant, J.; Le, Q.; Lao, N. Memory Augmented Policy Optimization for Program Synthesis and Semantic Parsing. In Proceedings of the NIPS'18: 32nd International Conference on Neural Information Processing Systems, Montréal, QC, Canada, 3–8 December 2018; Curran Associates Inc.: Red Hook, NY, USA, 2018; pp. 10015–10027.
51. Chen, X.; Liang, C.; Yu, A.W.; Zhou, D.; Song, D.; Le, Q.V. Neural Symbolic Reader: Scalable Integration of Distributed and Symbolic Representations for Reading Comprehension. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
52. Chen, X.; Liang, C.; Yu, A.W.; Song, D.; Zhou, D. Compositional Generalization via Neural-Symbolic Stack Machines. In Proceedings of the NIPS'20: 34th International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 6–12 December 2020; Curran Associates Inc.: Red Hook, NY, USA, 2020.
53. Sun, H.; Arnold, A.O.; Bedrax-Weiss, T.; Pereira, F.; Cohen, W.W. Faithful Embeddings for Knowledge Base Queries. In Proceedings of the NIPS'20: 34th International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 6–12 December 2020; Curran Associates Inc.: Red Hook, NY, USA, 2020.
54. Shi, J.; Cao, S.; Hou, L.; Li, J.Z.; Zhang, H. TransferNet: An Effective and Transparent Framework for Multi-hop Question Answering over Relation Graph. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Punta Cana, Dominican Republic, 7–11 November 2021.
55. Rincon-Yanez, D.; Senatore, S. FAIR Knowledge Graph construction from text, an approach applied to fictional novels. In Proceedings of the 1st International Workshop on Knowledge Graph Generation from Text and the 1st International Workshop on Modular Knowledge co-located with 19th Extended Semantic Web Conference (ESWC 2022), Hersonissos, Greece, 30 May 2022; pp. 94–108.
56. Diamantini, C.; Potena, D.; Storti, E. A Knowledge-Based Approach to Support Analytic Query Answering in Semantic Data Lakes. In Proceedings of the Advances in Databases and Information Systems, Turin, Italy, 5–8 September 2022; Chiusano, S.; Cerquitelli, T., Wrembel, R., Eds.; Springer International Publishing: Cham, Switzerland, 2022; pp. 179–192.
57. Xiangrong, Z.; Daojian, Z.; Shizhu, H.; Kang, L.; Jun, Z. Extracting Relational Facts by an End-to-End Neural Model with Copy Mechanism. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; pp. 506–514. [CrossRef]
58. Roth, D.; Yih, W.t. A Linear Programming Formulation for Global Inference in Natural Language Tasks. In Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004, Boston, MA, USA, 6–7 May 2004; Association for Computational Linguistics: Stroudsburg, PA, USA, 2004; pp. 1–8.
59. Yao, Y.; Ye, D.; Li, P.; Han, X.; Lin, Y.; Liu, Z.; Liu, Z.; Huang, L.; Zhou, J.; Sun, M. DocRED: A Large-Scale Document-Level Relation Extraction Dataset. In Proceedings of the ACL 2019, Florence, Italy, 28 July–2 August 2019.
60. Gurulingappa, H.; Rajput, A.M.; Roberts, A.; Fluck, J.; Hofmann-Apitius, M.; Toldo, L. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *J. Biomed. Inform.* **2012**, *45*, 885–892. [CrossRef] [PubMed]
61. Zhang, M. *Artificial Higher Order Neural Networks for Economics and Business*, 1st ed.; Information Science Reference: Hershey, PA, USA, 2009; Volume 1, p. 517.
62. Liu, Y.; Zhang, T.; Liang, Z.; Ji, H.; McGuinness, D.L. Seq2RDF: An End-to-end Application for Deriving Triples from Natural Language Text. In *CEUR Workshop Proceedings*, *Proceedings of the ISWC 2018 Posters & Demonstrations, Industry and Blue Sky Ideas Tracks Co-Located with 17th International Semantic Web Conference (ISWC 2018), Monterey, CA, USA, 8–12 October 2018*; van Erp, M., Atre, M., López, V., Srinivas, K., Fortuna, C., Eds.; CEUR-WS.org: Aachen, Germany 2018, Volume 2180.
63. Rossi, A.; Barbosa, D.; Firmani, D.; Matinata, A.; Merialdo, P. Knowledge graph embedding for link prediction: A comparative analysis. *ACM Trans. Knowl. Discov. Data* **2021**, *15*, 1–49. [CrossRef]
64. Kazemi, S.M.; Poole, D. SimplE Embedding for Link Prediction in Knowledge Graphs. In Proceedings of the NIPS'18: 32nd International Conference on Neural Information Processing Systems, Montréal, QC, Canada, 3–8 December 2018; Curran Associates Inc.: Red Hook, NY, USA, 2018; pp. 4289–4300.
65. Du, Y.; Mordatch, I. Implicit generation and modeling with energy-based models. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2019; Volume 32.

66. Di-Paolo, G.; Rincon-Yanez, D.; Senatore, S. FastKGQA: A Modified Knowledge Base of the MoviesQA Dataset for Prototyping. *Zenodo* **2023**. [CrossRef]
67. Nayyeri, M.; Xu, C.; Hoffmann, F.; Alam, M.M.; Lehmann, J.; Vahdati, S. Knowledge Graph Representation Learning using Ordinary Differential Equations. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Punta Cana, Dominican Republic, 7–11 November 2021; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 9529–9548. [CrossRef]