electronics

MDPI

Article

Knowledge Graph Multi-Hop Question Answering Based on Dependent Syntactic Semantic Augmented Graph Networks

Songtao Cai [1], Qicheng Ma [1], Yupeng Hou [1] and Guangping Zeng [1,2,*]

[1] School of Computer & Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China; b20180322@xs.ustb.edu.cn (S.C.); d202110388@xs.ustb.edu.cn (Q.M.); d202210392@xs.ustb.edu.cn (Y.H.)

[2] Beijing Key Laboratory of Knowledge Engineering for Materials Science, University of Science and Technology Beijing, Beijing 100083, China

[*] Correspondence: zgp@ustb.edu.cn

Abstract: In the rapidly evolving domain of question answering systems, the ability to integrate machine comprehension with relational reasoning stands paramount. This paper introduces a novel architecture, the Dependent Syntactic Semantic Augmented Graph Network (DSSAGN), designed to address the intricate challenges of multi-hop question answering. By ingeniously leveraging the synergy between syntactic structures and semantic relationships within knowledge graphs, DSSAGN offers a breakthrough in interpretability, scalability, and accuracy. Unlike previous models that either fall short in handling complex relational paths or lack transparency in reasoning, our framework excels by embedding a sophisticated mechanism that meticulously models multi-hop relations and dynamically prioritizes the syntactic–semantic context.

Keywords: question answering; knowledge graph-based multi-hop QA; knowledge graph embedding; deep learning

check for updates

Citation: Cai, S.; Ma, Q.; Hou, Y.; Zeng, G. Knowledge Graph Multi-Hop Question Answering Based on Dependent Syntactic Semantic Augmented Graph Networks. *Electronics* **2024**, *13*, 1436. https://doi.org/10.3390/electronics13081436

Academic Editor: Fabio Grandi

Received: 18 March 2024
Revised: 31 March 2024
Accepted: 7 April 2024
Published: 11 April 2024

1. Introduction

In the dynamic landscape of service mining and analytics, knowledge-based question answering (KBQA) systems, as outlined by Xiao et al. [1], have garnered significant interest across both academic and commercial sectors. These systems, designed to deduce precise target entities in response to queries posed in natural language, leverage extensive knowledge bases (KBs) as delineated by Bin et al. [2]. The core of KBQA's efficacy lies in its adeptness at interpreting complex semantic nuances inherent in natural language and pinpointing accurate responses within vast, structured knowledge repositories. A notable subset within this domain is knowledge graph question answering (KGQA), highlighted in studies by Hao et al. [3] and Michael et al. [4], which specifically employs knowledge graphs (KGs) for its knowledge repository. This approach, further explored by Bin et al. [2] and Yunshi et al. [5], capitalizes on the distinct organizational structure of KGs and their proficient querying mechanisms, offering users expedited access to deep, actionable insights encapsulated within KGs, thereby enhancing the overall user experience.

In the evolving landscape of multi-hop question answering (QA), the quest for sophisticated models capable of navigating complex information development across multiple documents has led to the development of specialized datasets like OpenBookQA [6], NarrativeQA [7], MultiRC [8], WikiHop [9], CommonsenseQA [10], and HotpotQA [11]. These platforms are designed to challenge the abilities of QA systems to perform intricate reasoning, with tasks ranging from identifying the correct answer from a set of options in WikiHop to locating specific answer snippets within paragraphs in HotpotQA.

Initial strategies focused on leveraging recurrent neural networks (RNNs) equipped with attention mechanisms to distill and extract pertinent information from texts, as seen in the early query-focused extractor (QFE) method by Nishida et al. [11] and the DecompRC approach by Min et al. [12], which simplifies complex multi-hop questions into more manageable sub-questions. Subsequent innovations have explored the selection of relevant document paragraphs and the utilization of chain reasoning, as demonstrated by the Dynamically Fused Graph Network (DFGN) by Qiu et al. [13] and hierarchical memory network models proposed by Jiang and Bansal [14], reflecting a growing sophistication in tackling multi-hop QA tasks. Recent advancements have embraced attention-based mechanisms and adaptive reinforcement learning, notably improving performance on complex QA tasks, as evidenced by the work of Zhao et al. [15] and the adaptive reinforcement learning (ARL) framework by Zhang et al. [16]. These developments underscore a shift towards more detailed and adaptive QA systems.

Simultaneously, the integration of graph neural networks (GNNs) has emerged as a pivotal innovation, enhancing the representation of complex entity relationships essential for multi-hop QA. The introduction of methods like multi-task prompting for graph neural networks highlights the potential of leveraging pre-trained models across various graph tasks, effectively bridging the gap between general graph knowledge and specific application needs [17]. The exploration of hypergraph representation for sociological analysis emphasizes the richness of social interactions and environments, providing a novel approach to understanding complex sociological phenomena through data mining techniques [18]. The factor-mixed Hawkes process (FMHP) for event-based incremental recommendations introduces a nuanced understanding of event generation, considering intrinsic, external, and historical intensities, thereby enhancing recommendation systems [19]. Furthermore, graph-masked autoencoders (GMAEs) represent a significant step forward in learning graph representations, adopting a self-supervised, transformer-based model that addresses the challenges of training deep transformers from scratch [20]. The novel recommendation model based on graph diffusion and the Ebbinghaus curve offers insights into users' evolving online interests, incorporating a graph diffusion method and neural network inspired by the Ebbinghaus curve to capture long-term and short-term tastes [21].

In the field of multi-hop QA, GNN-based models, such as those developed by Kipf and Welling [22] and Veličković et al. [23], have shown significant promise as well. Coref-GRU by Dhingra et al. [24], MHQA-GRN by Song et al. [25], and the breadth first reasoning graph (BFR-Graph) by Huang and Yang [26] exemplify the integration of entity recognition and graph construction to facilitate deeper information analysis. Moreover, the from easy to hard (FE2H) model by Li et al. [27] and the bidirectional recurrent graph neural network (BRGNN) by Zhang et al. [28] illustrate the ongoing evolution of graph-based reasoning in multi-hop QA, highlighting efforts to minimize errors and leverage relational patterns for improved QA performance.

However, as user queries become more complex, especially in multi-hop QA tasks that require synthesizing information from multiple knowledge points, the limitations of existing QA frameworks have become more pronounced. This includes challenges such as data sparsity, the difficulty of mapping complex queries to specific knowledge graph nodes and edges, and the need to dynamically update the knowledge base to incorporate the latest information. These issues highlight the need for adaptive, robust, and context-aware models that can navigate intricate knowledge graph structures with greater accuracy and comprehension.

In confronting these challenges, our paper introduces the Dependent Syntactic Semantic Augmented Graph Network (DSSAGN), an architecture designed to revolutionize QA systems' approach to multi-hop relational reasoning. To address the identified limitations, DSSAGN integrates external knowledge graphs to enrich the model with relevant entities, thereby providing a richer context for answering complex queries. Furthermore, by incorporating graph convolutional networks (GCNs) alongside dependency syntax analysis, DSSAGN significantly boosts the semantic understanding and interpretation of queries. These approaches not only facilitate a deeper comprehension of the relationships and entities within the knowledge graph, but also mirror the cognitive processes involved in human reasoning and comprehension.

Specifically, we present innovations in the realm of knowledge graph-based question answering, encapsulating our novel architecture, the DSSAGN. Our contributions are manifold, incorporating dependent syntactic analysis for a refined understanding of questions, graph convolutional networks (GCNs) for capturing intricate entity relationships, a KG Embedding Generator for advanced entity and relation embeddings, and an Answer Scoring Module for precise answer identification and ranking. These elements collectively propel DSSAGN beyond the current state-of-the-art, showcasing superior performance in multi-hop reasoning tasks through a synergistic approach that marries syntactic clarity with semantic depth and graph-based insights. The detailed source code is available at https://github.com/USTBSCCE1028/DSSAGN (accessed on 6 April 2024).

2. Materials and Methods

2.1. Materials

2.1.1. Dataset and Settings

In the experimental setup, two benchmark datasets are utilized to evaluate the performance of knowledge graph question answering tasks, which include CommonsenseQA [10] and OpenbookQA [6]. Success in these tests demands a deep grasp of world knowledge that surpasses mere textual comprehension. CommonsenseQA challenges users with a requirement for diverse kinds of commonsense reasoning. It generates its questions using elements from ConceptNet, aiming to explore the underlying complex relationships among these elements within ConceptNet's framework. OpenBookQA adopts the structure of an open-book test to evaluate an individual's grasp of a subject. Accompanying the queries are 1326 points of scientific knowledge tailored for elementary school education. Around 6000 crafted questions aim to test the comprehension of these knowledge points and their adaptability in unfamiliar situations. The data set in the experiment was divided as shown in Table 1.

Table 1. Partitioning of the data set in the experiment.

Dataset	Training Set	Test Set	Validation Set
CommonsenseQA	9741	1140	1221
OpenbookQA	4957	500	500

In addition, our evaluation uses ConceptNet, a knowledge graph across multiple domains proposed by Speer et al. [29], as an external knowledge base to assess the model's ability to utilize structured knowledge sources.

The initial configuration for training neural network models employs several key parameters designed to optimize performance and efficiency. Central to these settings is the use of the Adam optimizer, which is popular in deep learning applications due to its adaptive learning rate feature. The learning rate is set to 5×10^{-4}, a value that balances the risk of rapid convergence and minimization of the overshooting loss function. The batch size is set to 128, which is large enough to ensure that each batch captures representative samples of the dataset's diversity and thus stabilizes the gradient update. The training process was designed to run continuously for 30 epochs. This predetermined length of training time allows the model enough time to learn from the data and adjust the weights to effectively minimize the loss function. The discard rate is set to 0.3, a technique that prevents the network from over-relying on any one neuron, thus reducing model redundancy. Finally, BatchNorm (batch normalization) is enabled. It reduces internal covariance bias by normalizing the inputs of each layer, thereby increasing the learning rate, reducing the model's sensitivity to initialization, and ultimately speeding up the training process.

2.1.2. Experimental Environment

In this study, the DSSAGN model was built using the deep learning framework PyTorch, a Python3.7-based scientific computing library that provides highly flexible deep learning tools that support dynamic computational graphs and static computational graphs. The experimental environment and parameter settings are shown in Table 2 below.

Table 2. Experimental platform and environmental parameters.

Computer Information	Operating System	Windows 10 64-bit
	CPU	Intel(R) Core (TM) i5-8265U CPU @ 1.60 GHz (8 CPUs) ~1.8 GHz
	GPU	RTX 3060
	RAM	16 GB
Toolkit	Python 3.7	Numpy 1.21.5
		Scikit_Learn 1.0.2
		Pandas 0.25.1
		Torch 1.12.0
		Matplotlib 3.5.2

2.2. Methods

This section presents the details of the DSSAGN framework designed for knowledge graph question answering (KGQA) tasks, as illustrated in Figure 1. The framework incorporates various components and strategies, including a feature extractor, dependent syntactic analysis, graph neural network, graph embedding generator, and an answer scoring module.

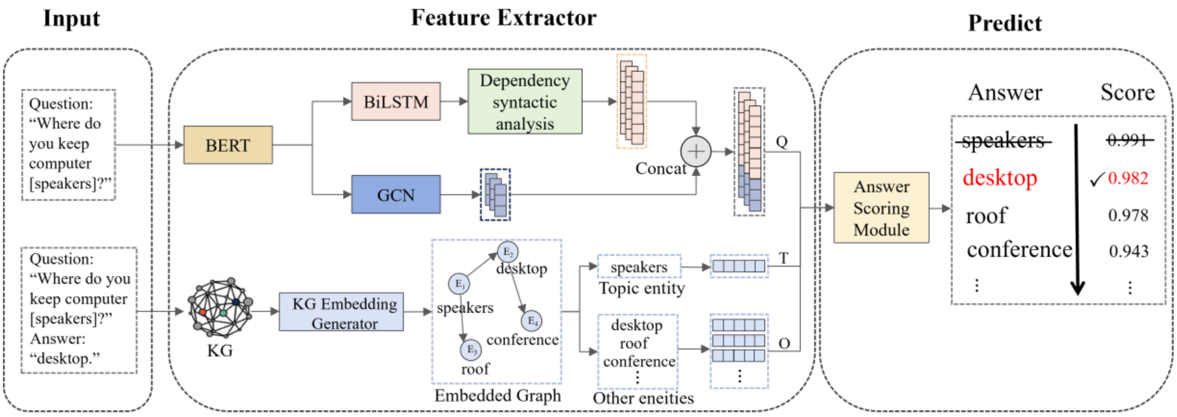

Figure 1. Overview of the proposed framework.

The initialization vectors of the question text are extracted using the BRET pre-training model [30], which is a popular word embedding learning method that incorporates local contextual information to learn word vectors. The Q&A text is fed into the KG to extract the entities associated with the text. BiLSTM works by reading an input sequence forward and backward twice to obtain context-sensitive hidden states at each time step. Specifically, for the input sequence $\{x_1, x_2, x_3, \ldots, x_n\}$ (where n is the number of words in the sequence), BiLSTM will generate a series of hidden states, $H = \{h_1, h_2, h_3, \ldots, h_n\}$. The GCN network [22] leverages the structure of the graph to capture the relationships between nodes by aggregating and passing information across the nodes within the graph. Within the structure of $G = (V, E)$, where each vertex, v, is endowed with a feature, x, the graph convolutional network (GCN) is employed to construct a representational vector, h, for every vertex. The updating mechanism of each vertex's representation leverages both the features of the vertex itself and the representations of its neighboring vertices. By iterating this mechanism, the representations of the vertices are progressively enriched to encapsulate the local and global attributes of $G = (V, E)$.

2.2.1. KG Embedding Generator

In our study, inspired by the outstanding results demonstrated in previous work, especially those of EmbedKGQA [31] and Rce-KGQA, proposed by Jin et al. [32], we explore the use of knowledge graph (KG) embedding techniques to solve some problems that are difficult to cope with through traditional methods, such as the inference of implicit relations and the handling of subgraph localization. We found that by embedding the global relational knowledge and structural information of KG into a continuous low-dimensional space, we can not only simplify the processing flow of KG, but also expect to improve the overall accuracy of the question answering process.

In this work, we adopt the method of mapping entities and relations in KG into a low-dimensional continuous vector space to obtain a sparse representation of these elements. In particular, we employ the Complex Embeddings model [33], which maps entities and relations into a complex vector space, in conjunction with considering semantic associations between elements. Compared to previous KG embedding techniques such as TransE [34] and its variants, semantic matching models such as ComplEx typically provide superior performance.

In the initialization phase of the embedding, the vector representations of all KG elements are randomly selected from a uniform distribution. The embedding dimensions of entities and relations are usually set to be no less than 100, and 200 dimensions are chosen in this study to be consistent with related studies. The training process involves extracting positive example fact triples of real-world relations from the KG while introducing false fact triples in the negative example generation step through negative sampling, i.e., randomly replacing tail entities or relations. This method helps the model learn to distinguish between true and false relationships.

Further analysis by Trouillon et al. points out that increasing the number of negative samples usually improves model performance. An appropriate ratio of positive to negative samples of about 1:50 balances both inference accuracy and training overhead. This empirical setting was followed in this study. For each relational triple $T, O \in \varepsilon$ and $Q \in \mathrm{R}$, our embedding method assigns it to a representation $v_T, v_Q, v_O \in Cd$ in the d-dimension complex vector space, and the scoring function is computed according to the following definition:

$$\begin{aligned} \varnothing(T, Q, O) &= Re(< v_T, v_Q, v_O >) \\ &= Re\left(\sum_{k=1}^{d} v_T^{(k)} v_Q^{(k)} v_O^{(k)}\right) \end{aligned} \tag{1}$$

$$\varnothing(T, Q, O) > 0 \, \forall \{T, Q, O\} \in A \tag{2}$$

$$\varnothing\left(T, \overline{Q}, \overline{O}\right) < 0 \, \forall \left\{T, \overline{Q}, \overline{O}\right\} \notin A \tag{3}$$

where $Re(\cdot)$ denotes the real portion of a complex number, while $\overline{v_O}$ indicates the complex conjugate of a specified target entity, v_O. Moreover, Q' and O' are used to denote sets of alternative triples; these are generated by selecting incorrect relations or tail entities at random, respectively. $\{T, Q, O\}$ refers to a relational triple composed of entities T, O and relation Q. Set A aggregates all the triples that reflect real-world knowledge.

Our goal is to optimize specific formulas to reduce the scores of false triples with negative values, while increasing the scores of true triples with positive values. This optimization process can be smoothly executed in each training round by stochastic gradient descent (DAD) or the Adam optimizer. In this way, the initial structural and relational information in the knowledge graph (KG) is preserved in the learned vectors, which effectively supports the execution of various downstream applications.

2.2.2. Dependent Syntactic Analysis Module

The feature vectors output from BiLSTM are fed into the dependency syntactic analysis module to obtain new feature representations. Dependency syntactic analysis is used to parse natural language sentences with the aim of revealing the dependencies between words. Specifically, a dependency tree is constructed, where each node represents a vocabulary word and each edge represents a syntactic dependency between words, forming a structured sentence representation. Its core objective can be expressed by the following formula:

$$L = \sum_{(i,j)\in E} \log P\left(r_{ij} \middle| w_i, w_j, \theta\right) \tag{4}$$

where L symbolizes the log-likelihood of a sentence's dependency tree, with E representing the collection of edges where each edge (i, j) signifies a dependency from word i to word j. Here, r_{ij} is defined as the relationship type between word i and word j, while w_i and w_j correspond to the lexical representations of word i and word j, respectively. θ denotes the parameters of the model.

The analysis process starts with the identification of the main verb of the sentence, progressively identifies the words directly associated with it (e.g., subject, object), and so on, until every word in the sentence is fully parsed. This process relies on a set of predefined grammatical rules and patterns that are based on a large amount of corpus data and in-depth linguistic theory.

2.2.3. Graph Convolutional Network

After obtaining the vector representations of BERT, we further input these vectors as node features into the graph neural network. The graph neural network can efficiently process graph structure data by iteratively updating the node representations to capture the dependencies between nodes and the global structure of the entire graph. In this process, the features of each node not only contain the semantic information of the original text, but also incorporate the information of other nodes connected to it, resulting in a new node representation that synthesizes the text content and graph structure information.

We employ a typical GNN variant, the graph convolutional network (GCN), for node feature updating. For every node within the graph, the updated feature representation is derived through a weighted aggregation of its features combined with those of its adjacent nodes. This process is mathematically depicted as:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{5}$$

where $H^{(l)}$ is the node identity matrix of layer l, $\tilde{A} = A + I_N$ is the adjacency matrix A of the graph plus the unit matrix I_N, $\tilde{D}$ is the diagonal node degree matrix, $W^{(l)}$ is the weight matrix of the layer, and σ is the activation function.

2.2.4. Answer Scoring Module

In this study, we employ a scoring mechanism that aims to rank each potential answer, t, based on a combined score of all possible knowledge graph (KG) entity pairs (i.e., topic entities and relationships). The formula is as follows:

$$Rank(O) = \begin{cases} \max(\varnothing(T, Q, O)),\ \forall O \in A \\ \min(\varnothing(T, Q, O)),\ \forall O \notin A \end{cases} \tag{6}$$

This process utilizes the complex scoring function proposed by Trouillon et al. [33], which seeks to optimize the model such that the likelihood of a positive sample t being included in set A is increased, while simultaneously reducing the score for any negative sample t' (not included in A). Here, A denotes the set comprising all real-world knowledge triples. It is crucial to note that to preserve the stability of the pre-trained knowledge graph entity embeddings, these embeddings are not altered during the model training phase.

Our approach doesn't just seek a balance between accuracy and recall; it implements a strategy that initially filters answers by setting a threshold of n (options are 5, 10, and 15). This is aimed at identifying candidate answers with high recall but moderate scores. In the inference phase, we assign a likelihood score to each candidate answer to reflect its credibility as the correct answer and screen out those top-ranked n but centered scoring candidates for further relational chain inference analysis.

The design of this method takes into account the adaptability of the actual relational properties and ensures that the model can efficiently process and analyze complex knowledge while progressively optimizing the model.

3. Results

This section presents the outcomes of comparative experiments, highlighting the efficacy of the DSSAGN framework. Additionally, an ablation study is conducted to assess the performance of each newly introduced architecture detailed within this document.

To enhance clarity and comprehension for our readers, we introduce a comprehensive overview of the evaluation metrics employed in our study. The metrics are as follows:

Accuracy (Acc.): This metric calculates the percentage of questions for which the model identified the correct answer out of the total questions posed. Given the task's emphasis on precise answer retrieval, accuracy serves as a primary indicator of the model's performance in correctly leveraging the knowledge graph for question answering.

Interpretability Index (II): A novel metric introduced in this study, measuring the ease with which the reasoning process of the model can be understood by human observers. The II is crucial for evaluating how well the model's decision-making process can be traced and comprehended, reflecting our research's objective to not only improve answer accuracy but also enhance transparency in reasoning.

3.1. Comparative Experiments

For a rigorous assessment of performance, various baseline models were chosen for benchmarking in comparative experiments against our proposed model. The selected baseline models are outlined as follows:

The R-GCN framework [35] is proposed for link prediction and entity classification tasks. It introduces parameter sharing and enforces sparsity constraint techniques to apply R-GCN to multi-graphs with a large number of relations.

The GconAttn framework [36] utilizes a combination of techniques for external knowledge to improve the performance of leaderless intelligence problems in scientific problem domains.

KagNet [37] is a textual reasoning framework designed to handle common sense-based queries that enables an interpretable reasoning process by integrating an external structured commonsense knowledge graph.

MHGRN [38] provides a new knowledge-aware approach that performs multi-hop multi-relational reasoning on subgraphs extracted from external knowledge graphs.

Rce-KGQA [26] has been introduced to discern inferential connections between topical entities and their answers within a knowledge graph (KG), leveraging both the explicit relation chains presented in user-posed questions and the implicit relation chains embedded within the structured KGs.

The results of the comparison experiments between the proposed model and other models are shown in Table 3.

In summary, the DA model framework demonstrated superior performance over the standard benchmark model. Specifically, within the CommonsenseQA dataset, the DA model enhanced the accuracy of answering knowledge graph-based questions by approximately 2% more than competing models, with validation set accuracy increasing by around 1.7% and test set accuracy by roughly 2.1%. When applied to the OpenbookQA dataset, the DA model framework showed an improvement in Q&A accuracy related to knowledge graphs by about 1% over alternative models. These outcomes underscore the DSSAGN framework's effectiveness in elevating performance across knowledge graph Q&A challenges. Most importantly, the BERT model is able to capture rich contextual information, and the design and combination with the dependent syntax approach and the GCN allow the model to more thoroughly understand the semantic information in the problem text. In this way, our framework not only utilizes the powerful representational capabilities of BERT, but also obtains a deeper understanding of the semantics through dependent syntax and GCN, making full use of the representational capabilities of the entire model. The KG Embedding Generator utilizes an external KG to embed related entities into a continuous low-dimensional space method, facilitating the inference of implicit relationships that may not be directly observed in the data. Additionally, the embedding process simplifies the identification and processing of relevant subgraphs in the KG. This is important for focusing on specific parts of the knowledge graph that are relevant to the query, thus improving the efficiency and accuracy of the response. In addition, the answer scoring module keeps the knowledge graph entity embeddings unchanged during the model training process, which helps to maintain the stability of the model and the validity of the pre-trained embeddings and improves the accuracy of question answering by more precisely identifying the most likely correct answer with comprehensive scoring and ranking of all potential answers.

Table 3. Performance of comparative experiments.

Model	**CommonsenseQA**		**OpenbookQA**	
	IHdev-Acc. (%)	**IHtest-Acc. (%)**	**Dev-Acc. (%)**	**Test-Acc. (%)**
R-GCN * [35]	56.72 (±0.42)	53.90 (±0.62)	63.51 (±1.81)	61.83 (±1.60)
GconAttn * [36]	56.37 (±0.72)	53.64 (±0.78)	62.62 (±1.07)	61.21 (±2.14)
KagNet * [37]	55.77 (±0.50)	56.39 (±0.53)	64.77 (±1.17)	61.83 (±2.05)
MHGRN * [38]	60.12 (±0.33)	56.93 (±0.72)	67.40 (±1.33)	66.15 (±1.45)
Rce-KGQA * [32]	61.52 (±0.42)	59.18 (±0.63)	67.72 (±1.13)	66.45 (±1.29)
DSSAGN framework *	63.22 (±0.20)	62.35 (±0.45)	68.52 (±0.93)	67.38 (±1.05)

All asterisks (*) denote results run on the local machine. For the CommonsenseQA dataset, we used the results of the evaluations performed on the internal development set (IHdev) and the test set (IHtest) based on the data partitioning defined by Lin et al. [37]. This includes the average accuracy of the four runs and their standard deviations.

Additionally, we conducted a comparative experiment on training time and testing time, comparing our model with the current advanced models. The purpose of this experiment was to evaluate not only the accuracy and effectiveness of each model in processing and answering questions based on knowledge graphs, but also the efficiency in terms of computational resources and time required for training. This is shown in Table 4.

Table 4. Efficiency of comparative experiments.

Model	**CommonsenseQA**		**OpenbookQA**	
	Training Time (min)	**Testing Time (min)**	**Training Time (min)**	**Testing Time (min)**
R-GCN * [35]	25.90	6.23	24.85	4.66
GconAttn * [36]	24.66	5.68	23.60	4.08
KagNet * [37]	22.05	4.88	19.55	3.26
MHGRN * [38]	21.25	4.45	19.03	3.05
Rce-KGQA * [32]	19.30	3.85	17.26	2.43
DSSAGN framework *	21.21	4.33	17.21	2.12

All asterisks (*) denote results run on the local machine.

In terms of efficiency, the DSSAGN framework performs slightly differently in terms of training and testing time on the two datasets. On the OpenbookQA dataset, which has the shortest testing time, DSSAGN is the most efficient, while on the CommonsenseQA dataset, DSSAGN has the second highest efficiency and has a similar training time as the fastest model, Rce-KGQA. This small difference highlights the robustness and adaptability of the DSSAGN framework across different datasets and tasks. It emphasizes the framework's ability to maintain a balance between computational efficiency and accuracy, not only ensuring fast processing times but also maintaining a high level of accuracy. This balance is crucial for real-world applications where both accuracy and computational efficiency in answering knowledge graph-based questions are critical.

3.2. Ablation Study

This section outlines a set of ablation studies designed to ascertain the discrete impact of each component on the overall efficacy of the proposed model. The objective of these experiments is to assess the utility of different modules when configured as follows:

Model 1: Excludes the dependent syntax module, relying solely on the features generated by BiLSTM combined with the outputs from GCN. This setup allows for a direct comparison with variants incorporating the dependency syntax approach.

Model 2: The answer scoring module is replaced with a fully connected layer as a predictor. In this configuration, we can compare it with the version that employs the answer scoring module strategy.

Model 3: Omits the GCN module, utilizing only the outputs from the dependent syntax module in tandem with the representations produced by EERT. This allows for evaluation against configurations that include the GCN component.

Based on the data presented in Table 5 for both the CommonsenseQA and OpenbookQA datasets, several key observations can be made:

Table 5. Ablation experiments for the proposed model.

Model	**CommonsenseQA**		**OpenbookQA**	
	IHdev-Acc. (%)	**IHtest-Acc. (%)**	**Dev-Acc. (%)**	**Test-Acc. (%)**
Model 1	61.35 (±0.38)	60.52 (±0.73)	66.40 (±1.64)	65.65 (±1.88)
Model 2	61.58 (±0.42)	61.88 (±0.67)	66.90 (±1.35)	65.91 (±1.47)
Model 3	62.67 (±0.25)	61.93 (±0.36)	67.89 (±1.03)	66.31 (±1.21)
DSSAGN framework	63.22 (±0.20)	62.35 (±0.45)	68.52 (±0.93)	67.38 (±1.05)

Firstly, eliminating the dependent syntax module significantly impacts the accuracy of the KGQA task, leading to a reduction in the overall accuracy for the CommonsenseQA dataset by approximately 2% and a similar decrease of about 2% for the OpenbookQA dataset. Furthermore, substituting the answer scoring module with a fully connected prediction layer results in a decline of roughly 1.5% in the overall accuracy for CommonsenseQA and about a 1.6% decrease for OpenbookQA. Lastly, the removal of the GCN module causes a decrease in overall accuracy by around 0.5% for CommonsenseQA and about 0.8% for OpenbookQA. Dependency syntax plays a more crucial role in the model's understanding of semantic information. The model interprets the intention of the sentence more accurately by revealing the dependency relationship between words, and the constructed dependency tree provides a structured representation of the sentence, which enables the model to better understand the deeper meanings and complex structures of the sentence. The answer scoring module also has a significant effect on the model's performance. It reduces the interference of low-quality answers by setting the screening threshold and preliminary filtering of candidate answers and keeps the knowledge graph entity embeddings unchanged during the model training process, which helps to maintain the stability of the model and the effectiveness of the pre-trained embeddings. In addition, the removal results of the GCN module show that graph convolutional networks play an important role in combining textual semantic information and graph structural information, further enhancing the accuracy and robustness of the Q&A system.

In summary, these experiments not only validate the importance of individual modules in enhancing the model's ability to handle complex Q&A tasks, but also emphasize the synergies of these components with each other, which together form an efficient and accurate Q&A framework.

4. Discussion

This study provides insights into the performance of our proposed DSSAGN framework and the contribution of its individual modules to the performance of the knowledge graph quizzing (KGQA) task through a series of comparative and ablation experiments. The experimental results clearly show that the DSSAGN framework outperforms the current baseline model. Furthermore, the removal of the dependent syntactic analysis module, the answer scoring module, and the graph convolutional network (GCN) module all lead to significant degradation of the model's performance, which attests to their importance in improving the model's understanding of complex Q&A tasks.

The enhanced interpretability of the DSSAGN framework is a pivotal advancement over existing approaches, primarily attributed to its unique integration of syntactic and semantic analysis within the knowledge graph domain. The core of this advancement lies in the dependency syntactic analysis module, which provides a systematic extraction and representation of syntactic dependencies between words. This explicit representation facilitates a deeper comprehension of the query structure, enabling the model to map textual queries to their semantic equivalents within the knowledge graph with high precision and transparency.

In particular, the dependency syntax analysis module enhances the model's understanding of deeper meanings and complex structures of sentences by revealing dependencies between words. The answer scoring module effectively improves answer accuracy and system robustness by comprehensively scoring and ranking candidate answers, while the GCN module further enhances the performance of the Q&A system by combining the semantic information of the text and the structural information of the knowledge graph.

Moreover, the synergy between the dependency syntactic analysis and the graph convolutional networks within DSSAGN further enriches the model's interpretability. The graph convolutional layers, informed by the structured syntactic representation, are adept at capturing the intricate relationships between entities within the knowledge graph. This dual-layered approach—combining syntactic clarity with semantic depth—provides a clear and coherent framework for understanding how the model navigates through the knowledge graph to arrive at an answer. DSSAGN enriches its model by integrating external knowledge graphs, a design that allows the model to adapt to and utilize larger or more comprehensive external knowledge graphs to enhance its performance. As the external knowledge graph expands, DSSAGN is able to access richer information and relationships to more accurately understand and answer complex queries. As a result, DSSAGN has the potential to demonstrate superior performance on both existing datasets and larger datasets that may emerge in the future. This approach based on the integration of external knowledge graphs ensures the scalability and flexibility of the model in dealing with diverse and complex problems and is a key factor in its ability to maintain high performance on different datasets.

In addition, DSSAGN demonstrates its uniqueness in dealing with knowledge graph problems compared to larger language models, such as LLaMA. DSSAGN achieves efficient computation through a smaller number of model parameters, which reduces the demand for computational resources to a certain extent and makes it more suitable for resource-limited environments. In addition, DSSAGN directly integrates with an external knowledge base, a design that allows the model to utilize the rich knowledge graph information to enhance the accuracy and depth of question answering, especially in scenarios that require multi-hop reasoning and complex relationship understanding. A clear advantage of DSSAGN over approaches such as LLaMA, which rely on massive amounts of data and complex network structures to improve model performance, is the time-efficiency of its training and inference processes. With a smaller number of parameters, DSSAGN is more flexible and faster in updating and optimizing.

In summary, through the in-depth analysis and validation of the key components in the DSSAGN framework, this study clearly demonstrates the importance and effectiveness of these components for enhancing the performance of knowledge graph Q&A, providing valuable insights and methodologies for further optimizing the Q&A system.

5. Conclusions

In summary, our proposed DSSAGN framework demonstrates significant effectiveness and innovation in handling knowledge graph question-answering tasks. Through a well-designed combination of modules, such as dependent syntactic analysis, answer scoring, and graph convolutional networks, the framework not only improves the depth and breadth of question comprehension, but also significantly enhances the accuracy of the quiz and the robustness of the system. The results of the ablation experiment further validate the importance of each module and the synergistic effect between them, providing valuable insights for future research and development in complex Q&A systems.

This study also demonstrates the importance of combining textual semantic information and knowledge graph structural information, proving the effectiveness of this combination in improving the performance of Q&A systems. Moreover, with the continuous expansion and enrichment of external knowledge graphs, DSSAGN is expected to further improve its performance and realize deeper semantic understanding and more accurate information retrieval. In addition, as technology advances and knowledge domains expand, larger and more comprehensive knowledge graphs are expected to emerge, providing DSSAGN with richer information sources. This will enable DSSAGN to not only provide more accurate answers, but also to understand and answer more complex multi-hop quizzing tasks. With the rapid development of knowledge graphs in various domains, we believe that the methodology and findings presented in this study will provide important references and insights for future research and applications of knowledge graph Q&A tasks.

Author Contributions: Conceptualization, S.C.; methodology, S.C. and Q.M.; software, S.C. and Y.H.; validation, S.C., Y.H. and Q.M.; formal analysis, S.C.; investigation, G.Z.; resources, S.C.; data curation, Y.H. and Q.M.; writing—original draft preparation, S.C., Q.M. and Y.H.; writing—review and editing, G.Z.; visualization, Q.M. and Y.H.; supervision, G.Z.; project administration, G.Z.; funding acquisition, G.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (General Program), grant number 62072031, and the Second Department, Second Institute of China Aerospace Science and Industry Corporation, grant number classified.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author.

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

References

1. Huang, X.; Zhang, J.; Li, D.; Li, P. Knowledge Graph Embedding Based Question Answering. In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, Melbourne, Australia, 30 January 2019; ACM: Melbourne, VIC, Australia, 2019; pp. 105–113.
2. Fu, B.; Qiu, Y.; Tang, C.; Li, Y.; Yu, H.; Sun, J. A Survey on Complex Question Answering over Knowledge Base: Recent Advances and Challenges. *arXiv* **2020**, arXiv:2007.13069.
3. Hao, Y.; Zhang, Y.; Liu, K.; He, S.; Liu, Z.; Wu, H.; Zhao, J. An End-to-End Model for Question Answering over Knowledge Base with Cross-Attention Combining Global Knowledge. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 221–231.
4. Petrochuk, M.; Zettlemoyer, L. SimpleQuestions Nearly Solved: A New Upperbound and Baseline Approach. *arXiv* **2018**, arXiv:1804.08798.
5. Lan, Y.; He, G.; Jiang, J.; Jiang, J.; Zhao, W.X.; Wen, J.-R. A Survey on Complex Knowledge Base Question Answering: Methods, Challenges and Solutions. *arXiv* **2021**, arXiv:2105.11644.

6. Mihaylov, T.; Clark, P.; Khot, T.; Sabharwal, A. Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering. *arXiv* **2018**, arXiv:1809.02789.
7. Kočiský, T.; Schwarz, J.; Blunsom, P.; Dyer, C.; Hermann, K.M.; Melis, G.; Grefenstette, E. The NarrativeQA Reading Comprehension Challenge. *Trans. Assoc. Comput. Linguist.* **2017**, *6*, 317–328. [CrossRef]
8. Khashabi, D.; Chaturvedi, S.; Roth, M.; Upadhyay, S.; Roth, D. Looking Beyond the Surface: A Challenge Set for Reading Comprehension over Multiple Sentences. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), New Orleans, LO, USA, 1–6 June 2018; pp. 252–262.
9. Welbl, J.; Stenetorp, P.; Riedel, S. Constructing Datasets for Multi-Hop Reading Comprehension Across Documents. *TACL* **2018**, *6*, 287–302. [CrossRef]
10. Talmor, A.; Herzig, J.; Lourie, N.; Berant, J. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. *arXiv* **2019**, arXiv:1811.00937.
11. Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W.W.; Salakhutdinov, R.; Manning, C.D. HotpotQA: A Dataset for Diverse, Explainable Multi-Hop Question Answering. *arXiv* **2018**, arXiv:1809.09600.
12. Min, S.; Zhong, V.; Zettlemoyer, L.; Hajishirzi, H. Multi-Hop Reading Comprehension through Question Decomposition and Rescoring. *arXiv* **2019**, arXiv:1906.02916.
13. Qiu, L.; Xiao, Y.; Qu, Y.; Zhou, H.; Li, L.; Zhang, W.; Yu, Y. Dynamically Fused Graph Network for Multi-Hop Reasoning. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 6140–6150.
14. Jiang, Y.; Bansal, M. Self-Assembling Modular Networks for Interpretable Multi-Hop Reasoning. *arXiv* **2019**, arXiv:1909.05803.
15. Zhao, C.; Xiong, C.; Rosset, C.; Song, X.; Bennett, P.; Tiwary, S. Transformer-xh: Multi-evidence reasoning with extra hop attention. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 30 April 2020.
16. Zhang, Q.; Weng, X.; Zhou, G.; Zhang, Y.; Huang, J.X. ARL: An Adaptive Reinforcement Learning Framework for Complex Question Answering over Knowledge Base. *Inf. Process. Manag.* **2022**, *59*, 102933. [CrossRef]
17. Sun, X.; Cheng, H.; Li, J.; Liu, B.; Guan, J. All in One: Multi-Task Prompting for Graph Neural Networks. In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, ACM, Long Beach, CA, USA, 6 August 2023; pp. 2120–2131.
18. Sun, X.; Cheng, H.; Liu, B.; Li, J.; Chen, H.; Xu, G.; Yin, H. Self-Supervised Hypergraph Representation Learning for Sociological Analysis. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 11860–11871. [CrossRef]
19. Cui, Z.; Sun, X.; Pan, L.; Liu, S.; Xu, G. Event-Based Incremental Recommendation via Factors Mixed Hawkes Process. *Inf. Sci.* **2023**, *639*, 119007. [CrossRef]
20. Zhang, S.; Chen, H.; Yang, H.; Sun, X.; Yu, P.S.; Xu, G. Graph Masked Autoencoders with Transformers. *arXiv* **2022**, arXiv:2202.08391.
21. Cui, Z.; Sun, X.; Chen, H.; Pan, L.; Cui, L.; Liu, S.; Xu, G. Dynamic Recommendation Based on Graph Diffusion and Ebbinghaus Curve. *IEEE Trans. Comput. Soc. Syst.* **2024**, *11*, 2755–2764. [CrossRef]
22. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv* **2017**, arXiv:1609.02907.
23. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. In Proceedings of the 6th ICLR 2018, Vancouver, BC, Canada, 30 April–3 May 2018.
24. Dhingra, B.; Jin, Q.; Yang, Z.; Cohen, W.W.; Salakhutdinov, R. Neural Models for Reasoning over Multiple Mentions Using Coreference. *arXiv* **2018**, arXiv:1804.05922.
25. Song, L.; Wang, Z.; Yu, M.; Zhang, Y.; Florian, R.; Gildea, D. Exploring Graph-Structured Passage Representation for Multi-Hop Reading Comprehension with Graph Neural Networks. *arXiv* **2018**, arXiv:1809.02040.
26. Huang, Y.; Yang, M. Breadth First Reasoning Graph for Multi-Hop Question Answering. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online, 6–11 June 2021; pp. 5810–5821.
27. Li, X.-Y.; Lei, W.-J.; Yang, Y.-B. From Easy to Hard: Two-Stage Selector and Reader for Multi-Hop Question Answering. In Proceedings of the ICASSP 2023—2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 4 June 2023; pp. 1–5.
28. Zhang, G.; Liu, J.; Zhou, G.; Xie, Z.; Yu, X.; Cui, X. Query Path Generation via Bidirectional Reasoning for Multihop Question Answering from Knowledge Bases. *IEEE Trans. Cogn. Dev. Syst.* **2023**, *15*, 1183–1195. [CrossRef]
29. Speer, R.; Chin, J.; Havasi, C. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31. [CrossRef]
30. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2019**, arXiv:1810.04805.
31. Saxena, A.; Tripathi, A.; Talukdar, P. Improving Multi-Hop Question Answering over Knowledge Graphs Using Knowledge Base Embeddings. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 4498–4507.
32. Jin, W.; Zhao, B.; Yu, H.; Tao, X.; Yin, R.; Liu, G. Improving Embedded Knowledge Graph Multi-Hop Question Answering by Introducing Relational Chain Reasoning. *Data Min. Knowl. Disc.* **2023**, *37*, 255–288. [CrossRef]
33. Trouillon, T.; Welbl, J.; Riedel, S. Complex Embeddings for Simple Link Prediction. *Proc. Mach. Learn. Res.* **2016**, *48*, 2071–2080.

34. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating Embeddings for Modeling Multi-Relational Data. In Proceedings of the Advances in Neural Information Processing Systems 26, Lake Tahoe, NV, USA, 5–10 December 2013.
35. Gangemi, A.; Navigli, R.; Vidal, M.-E.; Hitzler, P.; Troncy, R.; Hollink, L.; Tordai, A.; Alam, M. (Eds.) *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, 3–7 June 2018*; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2018; Volume 10843, ISBN 978-3-319-93416-7.
36. Wang, X.; Kapanipathi, P.; Musa, R.; Yu, M.; Talamadupula, K.; Abdelaziz, I.; Chang, M.; Fokoue, A.; Makni, B.; Mattei, N.; et al. Improving Natural Language Inference Using External Knowledge in the Science Questions Domain. *AAAI* **2019**, *33*, 7208–7215. [CrossRef]
37. Lin, B.Y.; Chen, X.; Chen, J.; Ren, X. KagNet: Knowledge-Aware Graph Networks for Commonsense Reasoning. *arXiv* **2019**, arXiv:1909.02151.
38. Feng, Y.; Chen, X.; Lin, B.Y.; Wang, P.; Yan, J.; Ren, X. Scalable Multi-Hop Relational Reasoning for Knowledge-Aware Question Answering. *arXiv* **2020**, arXiv:2005.00646.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.