CrunchQA - A Synthetic Dataset for Question Answering over Crunchbase Knowledge Graph

Lifan Yu
New York University
Shanghai, China
ly1164@nyu.edu

Nadya Abdel Madjid
New York University
Abu Dhabi, UAE
na3112@nyu.edu

Djellel Difallah
New York University
Abu Dhabi, UAE
dd115@nyu.edu

***Abstract*—The digital transformation in the finance and enterprise sector has been driven by the advances made in big data and artificial intelligence technologies. For instance, data integration enables businesses to make better decisions by consolidating and mining heterogeneous data repositories. In particular, knowledge graphs (KGs) are used to facilitate the integration of disparate data sources and can be utilized to answer complex natural language queries. This work proposes a new dataset for question-answering on knowledge graphs (KGQA) to reflect the challenges we identified in real-world applications which are not covered by existing benchmarks, namely, multi-hop constraints, numeric and literal embeddings, ranking, reification, and hyper-relations. To build the dataset, we create a new Knowledge Graph from the Crunchbase database using a lightweight schema to support high-quality entity embeddings in large graphs. Next, we create a Question Answering dataset based on natural language question generation using predefined multiple-hop templates and paraphrasing. Finally, we conduct extensive experiments with state-of-the-art KGQA models and compare their performance on CrunchQA. The results show that the existing models do not perform well, for example, on multi-hop constrained queries. Hence, CrunchQA can be used as a challenging benchmark dataset for future KGQA reasoning models. The dataset and scripts are available on the project repository.**

***Index Terms*—Knowledge Graph, Embedding, Question Answering, Fintech, Enterprise KB**

I. Introduction

Knowledge graphs are a simple data organization model that uses graph vertices and edges to represent real-world entities and their interconnections. This representation can facilitate data integration from various sources by leveraging common and machine-readable ontologies. Motivated by such flexibility and the opportunities they offer, KGs have been advocated and embraced within several industries [1]. Beyond data integration, KGs are used as a source of factual information (e.g., in Google Knowledge Panel, Wikipedia Infoboxes), to infer missing (latent) facts [2], and to answer complex multi-hop queries. The users can interact with the KG using structured language (such as SPARQL) or natural language. The latter is a task commonly referred to as Question Answering over Knowledge Graphs (KGQA) [3].

In this paper, we start by reviewing existing KGQA datasets and their limitations, highlighting missing features which are critical to real-world applications (Sec. II). To fill this gap, we take the enterprise and finance domain as a use-case and create a new dataset that reflects the challenges we identified. This choice is motivated by the sundry of applications of finance based KGs [4], [5], for example, fraud detection [6]–[8], stock prediction [9], [10], information extraction and linking [11], and automated KG construction [12].

Similar to our effort, Zehra et al. [13] constructed a KG from structured/unstructured data and implemented a system for resolution of intelligent financial queries on top of the KG. The system extracts target entities from a query, converts it into a graph querying statement and extracts the results from the graph. In this work, we create a KGQA dataset based on a knowledge graph we drive from the Crunchbase database[1], which contains information about technology companies. The questions are automatically generated from a set of carefully designed templates and paraphrases that capture several complexity levels. In summary, we make the following key contributions:

- We build a Knowledge Graph from the Crunchbase database using a lightweight schema. (Sec. III-A)
- We build a Question Answering dataset generator using multi-hop templates and question paraphrasing. Unlike other datasets, the templates have an ordinal facet which is relevant for fintech and enterprise related questions. (Sec. III-B)
- We propose a simple clustering-based method to add numerical literals support to existing embedding-based methods. (Sec. IV)
- We conduct extensive experiments to evaluate state-of-the-art KGQA models on CrunchQA, their shortcomings, and highlight future area of research. (Sec. IV)

II. Background

A knowledge graph $\mathcal{G}$ is a directed labeled graph where the nodes represent entities $\mathcal{V}$ and the edges connecting these nodes represent relations $\mathcal{R}$ existing between the entities. Mathematically, we define the knowledge graph $\mathcal{G}$ as a set of triplets $(h, r, t) \in \mathcal{E} \subseteq \mathcal{V} \times \mathcal{R} \times \mathcal{V}$ with each triplet representing a directed *relation* r between a *head entity* h and a *tail entity* t.

KGQA. The task of question answering over a knowledge graph $\mathcal{G}$ involves taking question q (in textual form) and a head-entity h (or the main subject of the question) and producing the correct set of answer entities $\mathcal{A}^q \subset \mathcal{V}$. Moreover, an effective KGQA system must be capable of multi-hop reasoning and handling questions in the presence of missing links in the

[1] https://data.crunchbase.com/

knowledge graph. This scenario can be simulated by truncating edges from the KG.

Datasets. To build a KGQA model, a QA dataset $\mathcal{D}$ is provided with a set of tuples (q, h, a), where $a \in \mathcal{A}^q$. Each tuple may have an *inferential chain*, that is, the subset of relations connecting the head h entity and the answer entity a, in the context of the question q. In the following, we summarize the properties of the three common datasets used to evaluate KGQA methods:

1) METAQA [14] is a large-scale multi-hop QA dataset containing over 400k movie questions. It includes 1-hop, 2-hop and 3-hop questions along with the answers, question templates, and the underlying knowledge graph. The questions do not contain constraints and have *years* as literals. The underlying graph contains 43K entities and only 10 relation types.
2) WEBQUESTIONSSP (WQSP) [15] is a smaller QA dataset consisting of 4,737 natural language questions that are answerable through 1 and 2-hop reasoning over a subset of Freebase KG, containing roughly 2M entities and 550 relation types. Part of questions have constraints setting the gender, city or date and few questions require year as answers.
3) COMPLEXWEBQUESTIONS (CWQ) [16] is built on top of WebQuestionsSP dataset by virtue of adding constraints to the existing questions. In total the dataset contains four types of constraints: composition (45%), conjunction (45%), comparative (5%), and superlative (5%) and require up to 3 and 4-hops of reasoning over Freebase KG to reach the answer.

Table I gives an overview of the performance of recent methods on these datasets. We observe that methods based on graph embeddings and path reasoning solve MetaQA in multi-hop questions. A manual inspection reveals that the remaining gap in MetaQA's 1-hop questions is mainly due to errors in the dataset. The same methods have improved the results in WQSP and CWQ; however, there is still plenty of room for improvement.

We develop CrunchQA based on the MetaQA approach (i.e., automatic question generation from templates) while adding the challenges that WQSP and CWQ offer, namely, having a large KG and constraints. Moreover, none of these datasets contain realistic numeric, range, and other ordinal-based questions. For example, *"How tall is the Eiffel tower?"*, *List the 5 latest EdTech companies IPOs*, *"Who were the presidents of the United States during the 1960s?"*. Additionally, all of these datasets have a unique form per question pattern, which poses the issue of overfitting. We tackle this issue by creating multiple template paraphrases unseen during training.

TABLE I
HITS@1 RESULTS ON KGQA DATASETS. THE METHODS ARE ORDERED BY ASCENDING PUBLICATION YEAR. (RESULTS FROM [17])

Model	MetaQA			WQSP	CWQ
	1-hop	2-hop	3-hop		
KVMemNN [18]	95.8	25.1	10.1	46.7	21.1
VRN [14]	**97.5**	89.9	62.5	–	–
GraftNet [19]	97.0	94.8	77.7	66.4	32.8
PullNet [20]	97.0	99.9	91.4	68.1	47.2
SRN [21]	97.0	95.1	75.2	–	–
ReifKB [22]	96.2	81.1	72.3	52.7	–
EmbedKGQA [23]	**97.5**	98.8	94.8	66.6	–
TransferNet [17]	**97.5**	**100**	**100**	**71.4**	**48.6**
SQALER-GNN	**97.5**	**100**	**100**	**71.4**	**48.6**

III. CRUNCHQA DATASET

In this section, we will describe the building blocks used by our method to construct a Knowledge Graph from the Crunchbase database dump[2], as well as the generation of QA templates and the QA dataset.

A. Knowledge Graph Construction

The Crunchbase data dump comprises 17 relational tables[3] with primary and foreign keys to link tables together. To build the KG, we use a simple approach. First, we create new entities for each main entity type, namely, ORGANIZATION, PERSON, FUND, EVENT, and FUNDING ROUND.

Next, we use reification nodes JOB, ACQUISITION and IPO to map the relationship between the base entity types. For example, instances of JOB are entities that link instances of ORGANIZATION, PERSON and add additional information about this particular position, for example, start date, end date, title, etc.

Finally, we map additional triples extracted from the entities' respective tables including numerical values and dates. At the same time, we exclude fields with textual literals and various metadata irrelevant to our task. The processed knowledge graph includes 3.2 million entities, 31 relations, and 17.6 million triples.

Figure 1 shows the overall structure of our Knowledge graph, where rectangular nodes denote the main classes of entities and the circular nodes visualize the reification we perform. A detailed description of all the entities and relation types is available on the dataset's website. In addition, since the Crunchbase dataset is subject to licensing, we provide a script to process a dump and reconstruct the KG at a given timestamp.

B. Question Answering Templates

To obtain an initial set of templates, we construct a supplementary bidirectional graph $\mathcal{G}_s$ with main class entities and reification nodes as vertices and relations as edges. The graph $\mathcal{G}_s$ reflects the scheme of the Knowledge graph $\mathcal{G}$. At the next stage, we traverse the graph $\mathcal{G}_s$ from every vertex of main class entities and record all possible paths and vertices we can reach with 1 and 2 edges. In the context of templates, the result of this stage is the list of potential inferential chains I, where the source vertex point to the head entity and destination vertex to the answer entity.

At the next step through manual screening, chains which seem redundant and not reasonable are filtered out. The remained inferential chains are screened for a possibility to attach constraints. The types of constraints used in this paper are entity, temporal, maximum and numeric. These specific

[2]The database schema description is available at: https://data.crunchbase.com/docs

[3]The database dump was obtained on December 2021.

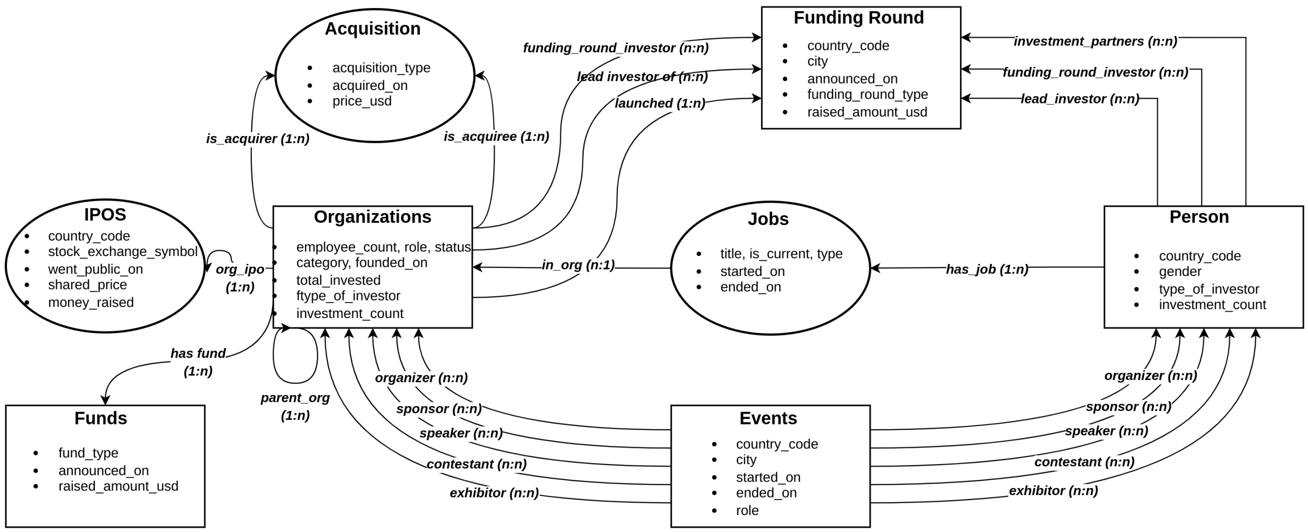

Fig. 1. Diagram of the constructed Knowledge Graph showing main entities, reification nodes and relations with cardinality. Elliptical nodes represent reification nodes.

TABLE II
QA TEMPLATES: OVERALL THE DATASET CONTAINS 243 TEMPLATES. NOTE, ADVANCED NO CONSTRAINT INDICATE MULTI-ENTITY/RELATION QUESTIONS; SINGLE-ENTITY ADVANCED QUESTIONS BELONG TO SEVERAL CONSTRAINTS, SO THE PERCENTAGES DO NOT SUM UP TO 100.

Type	Constraint category	Example	Percentage
1hop	no constraint	**Seed Round - Celtra** was launched in which country	11.5%
	entity constraint	list **conferences** that took place in **Abu Dhabi**	10.2%
	temporal constraint	which funding round did **Sweetch** launch in **July 2021**	21.8%
	max constraint	what types of event did **Meta most frequently** participate in?	0.4%
2hop	no constraint	from which industries are the investors of **Series A - Healthera**	25.9%
	entity constraint	**Meta** invested in which **U.S.** companies	2.9%
advanced	no constraint	who was the lead investor of **Seed Round - QuotaPath**	4.5%
	entity constraint	list companies in the **Home Decor** sector that released their initial public offering on **sse**	13.5%
	temporal constraint	which Ad Targeting companies were founded **after 2019**	8.2%
	max constraint	which Rental Property company has **the highest** ipo share price	2%
	numeric constraint	list people who formerly worked in **Chaotic Moon Studios** and founded **more than 3** companies	2.5%

types and the format of constraints are based on our inspiration to cover the most informative insights a user can infer from the KG. After this stage, each template t is represented by an inferential chain I^t and the set of constraints C^t.

The templates are classified into 3 categories: 1-hop (inferential chain of the length 1 and at most 1 constraint), 2-hop (inferential chain of the length 2 and at most 1 constraint) and the rest, as more challenging templates are classified as advanced. At the last stage, we significantly populate the advanced templates with the most popular business inquires we find on forums and Crunchbase FAQ. For advanced templates, to ensure diversity in lexicalization of questions, each template has several options (paraphrases) for a question query. When generating questions we randomly select which query form to use.

Following, we provide the description and format of each constraint type. ***Entity constraint*** requires a specific entity to be equal to a certain value, e.g., for queries asking about female founders in Abu Dhaib it can be specified as gender = 'female', city = 'Abu Dhabi' and job_title='founder'. ***Temporal constraint*** requires the date to be within a specified time range. A sufficient set of questions that we surfed require a time range, and not necessary an implicit timestamp. Some of the questions are aimed to see the dynamic attached to a specific period (e.g., pandemic), which can be reflected through a temporal constraint. For example, from the beginning of COVID-19 can specified as "after 2019" in the template. ***Maximum constraint*** is introduced to reflect key words as "top", "at most", "the highest" and etc. Maximum is always computed within a group, e.g., if we want to know "which Software companies have the highest ipo share price", the constraint first specifies grouping by the category Software and then selects the maximum among share prices. Another setting the maximum constraint supports is first counting over edges and then selecting maximum, e.g., "companies acquired by Meta mostly come from which industry". In this example, the number of companies that Meta acquired in each industry is counted and the industry with the highest count is selected. ***Numeric constraint*** reflect the key words "more than", "less than", "at least". This constraint implies counting over edges, e.g., for a question "list companies with acquired more than 50 companies", the constraint first specifies grouping by organization acquisitions where it is an acquirer, then counts acquirees and selects a company based on a condition for a number of acquirees > 50. ***Multi-entity/relation*** type is introduced to cover questions which can refer to multiple entities or relations, e.g., if we ask about investors, both companies and people can make investments, or if a question is about participating in an event without specifying a specific role, we should encounter all types of relations, i.e., sponsor, speaker, organizer, contestant and exhibitor.

C. Question Answering Dataset

Giving a set of templates T, we generate N questions per template, where N is a parameter. Algorithm 1 depicts the overall procedure of question generation from templates. Each constraint chain (an entity-relation-entity chain. e.g., constraint "companies from the U.S." has constraint chain "org-country_code-country_code, "country_code" = "U.S.") set is merged with the set formed by the main chain through inner join. Before sampling from the final set of records, we group by question to accumulate answers. After grouping, the questions with more than 20 answers are dropped. Finally, we sample N questions and improve the readability of the questions. In algorithm 1, *make_readable()* is a human-computation function to verify the lexicalization of the final questions. While this operation can be executed using crowdsourcing, we were able to screen most of the anomalies manually.

We further split the questions into three categories based on the data type of their answers: TEMPORAL questions return date information, NUMERIC questions return numbers. The rest of the questions are VANILLA questions, i.e., they return regular entities. Table III shows the statistics of the QA dataset.

Algorithm 1 Generation of QA dataset

Require: $T, \mathcal{G}, N$
  $Q = []$ #list of questions
  for t in T **do**
    main_set = *extract*(t.main_inferential_chain)
    for c in C^t **do**
      sub_set = *extract*(c.inferential_chain)
      inner_join(main_set, sub_set)
      group_by_question(main_set)
    end for
    samples = *select_samples*(main_set, N)
    questions = *make_readable*(samples)
    append(Q, questions)
  end for
  return Q

IV. EVALUATION

We focus our evaluation on EmbedKGQA [23], an approach that combines graph embeddings and reasoning using graph traversal. We select this approach because it proposes a general framework that captures the main ideas of the state-of-the-art techniques while allowing for experimenting with different backend KG embedding models. The EmbedKGQA framework has two training stages: (i) train the knowledge graph embedding on the link prediction task; (ii) train the model for QA by leveraging the pretrained embeddings. The model aims to project the question into the space of KG's embedding. To learn the question representation, the model uses a pretrained transformer (SBERT [24]) followed by two linear layers.

TABLE III
QA DATASET STATISTICS

	Train	**Valid**	**Test**
Vanilla	13485 (65.6%)	1847 (62.9%)	3792 (64.6%)
Temporal	6394 (31.1%)	969 (33%)	1878 (32%)
Numeric	679 (3.3%)	120 (4.1%)	203 (3.4%)
Total	20558 (70%)	2936 (10%)	5873 (20%)

1-hop	**2-hop**	**Advanced**	**Total questions**
10515 (35.8%)	6732 (22.9%)	12120 (41.3%)	29367

TABLE IV
THE EVALUATION OF EMBEDKGQA WITH SEVERAL KG EMBEDDINGS MODELS. WE REPORT HITS@1 ON CRUNCHQA.

KG Model	Vanilla	Temporal	Numeric
TransE	17%	-	-
ComplEx	12%	-	-
DistMult	12.7%	-	-
ComplExLiteral	13.86%	-	-
DistMultGatedLiteral	10.68%	-	-
TransE (clustering)	**18.2%**	8.09%	**37.43%**
ComplEx (clustering)	13.2%	8.89%	33.49%
DistMult (clustering)	11.1%	**8.99%**	8.86%

To learn KG embeddings, we use the PyKEEN [25] library. We conduct experiments with three KG embedding models: TransE [2], ComplEx [26], DistMult [27] and two literals models, i.e., ComplExLiteral [28] and DistMultGatedLiteral [28]. Literals models incorporate literals information to compute entity embeddings. However, the entities embedding matrix does not contain embeddings for literals in both settings. Since the model can generate as the answer only one of the entities, the model cannot answer temporal and numeric questions in these settings.

To answer the temporal and numeric question, we propose as a simple baseline a clustering-based variant of the models. In this approach, dates and numeric values are clustered and treated as entities; thus the model can compute their embeddings. To achieve this, dates are clustered into *years*, omitting days and month information. We perform K-means clustering among the categories for numeric as 'share_price' or 'investment_count'.

Table IV shows the qualitative results for the described setups where we report Hits@1. Among vanilla settings, TransE model achieves the best performance as in the case of clustered data for vanilla and numeric questions. For ComplEx model, link prediction Hits@1 after incorporating literals information increased almost two times, which results in rise of 1.86% for QA.

REFERENCES

[1] J. Shinavier, K. Branson, W. Zhang, S. Dastgheib, Y. Gao, B. Arsintescu, F. Özcan, and E. Meij, "Panel: Knowledge graph industry applications," in *Companion Proceedings of The 2019 World Wide Web Conference*, 2019, pp. 676–676.

[2] A. Bordes, N. Usunier, A. Garcia-Durán, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data." Red Hook, NY, USA: Curran Associates Inc., 2013.

[3] L. Hirschman and R. J. Gaizauskas, "Natural language question answering: the view from here," *Nat. Lang. Eng.*, vol. 7, no. 4, pp. 275–300, 2001.

[4] J. P. McCrae, P. Mohanty, S. Narayanan, B. Pereira, P. Buitelaar, S. Karmakar, and R. Sarkar, "Conversation concepts: Understanding topics and building taxonomies for financial services," *Information*, vol. 12, no. 4, 2021. [Online]. Available: https://www.mdpi.com/2078-2489/12/4/160

[5] Y. Wang, J. Zhao, F. Li, and M. Yu, "Construction of knowledge graph for internal control of financial enterprises," in *2020 IEEE 20th International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, 2020, pp. 418–425.

[6] Q. Zhan and H. Yin, "A loan application fraud detection method based on knowledge graph and neural network," ser. ICIAI '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 111–115. [Online]. Available: https://doi.org/10.1145/3194206.3194208

[7] R. Miao, X. Zhang, H. Yan, and C. Chen, "A dynamic financial knowledge graph based on reinforcement learning and transfer learning," in *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 5370–5378.

[8] J. Wang, Y. Guo, X. Wen, Z. Wang, Z. Li, and M. Tang, "Improving graph-based label propagation algorithm with group partition for fraud detection," *Applied Intelligence*, vol. 50, no. 10, p. 3291–3300, oct 2020. [Online]. Available: https://doi.org/10.1007/s10489-020-01724-1

[9] Y. Liu, Q. Zeng, J. Ordieres Meré, H. Yang, and B. M. Tabak, "Anticipating stock market of the renowned companies: A knowledge graph approach," *Complex.*, vol. 2019, jan 2019. [Online]. Available: https://doi.org/10.1155/2019/9202457

[10] S. Deng, N. Zhang, W. Zhang, J. Chen, J. Z. Pan, and H. Chen, "Knowledge-driven stock trend prediction and explanation via temporal convolutional network," ser. WWW '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 678–685. [Online]. Available: https://doi.org/10.1145/3308560.3317701

[11] Z. Wang, X. Zhang, and Y. Hu, "A semantic path based approach to match subgraphs from large financial knowledge graph," in *Proceedings of the 2019 International Conference on Mathematics, Big Data Analysis and Simulation and Modelling (MBDASM 2019)*. Atlantis Press, 2019/10, pp. 86–92. [Online]. Available: https://doi.org/10.2991/mbdasm-19.2019.20

[12] S. Elhammadi, L. V.S. Lakshmanan, R. Ng, M. Simpson, B. Huai, Z. Wang, and L. Wang, "A high precision pipeline for financial knowledge graph construction," in *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 967–977. [Online]. Available: https://aclanthology.org/2020.coling-main.84

[13] S. Zehra, S. F. M. Mohsin, S. Wasi, S. I. Jami, M. S. Siddiqui, and M. K.-U.-R. R. Syed, "Financial knowledge graph based financial report query system," *IEEE Access*, vol. 9, pp. 69 766–69 782, 2021.

[14] Y. Zhang, H. Dai, Z. Kozareva, A. J. Smola, and L. Song, "Variational reasoning for question answering with knowledge graph," in *AAAI*. AAAI Press, 2018, pp. 6069–6076.

[15] W.-t. Yih, M.-W. Chang, X. He, and J. Gao, "Semantic parsing via staged query graph generation: Question answering with knowledge base," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, Jul. 2015, pp. 1321–1331. [Online]. Available: https://aclanthology.org/P15-1128

[16] A. Talmor and J. Berant, "The web as a knowledge-base for answering complex questions," 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, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 641–651. [Online]. Available: https://aclanthology.org/N18-1059

[17] L. H. J. L. H. Z. Jiaxin Shi, Shulin Cao, "Transfernet: An effective and transparent framework for multi-hop question answering over relation graph," in *EMNLP*, 2021.

[18] A. Miller, A. Fisch, J. Dodge, A.-H. Karimi, A. Bordes, and J. Weston, "Key-value memory networks for directly reading documents," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 1400–1409. [Online]. Available: https://aclanthology.org/D16-1147

[19] H. Sun, B. Dhingra, M. Zaheer, K. Mazaitis, R. Salakhutdinov, and W. Cohen, "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: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 4231–4242. [Online]. Available: https://aclanthology.org/D18-1455

[20] H. Sun, T. Bedrax-Weiss, and W. Cohen, "PullNet: Open domain question answering with iterative retrieval on knowledge bases and text," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 2380–2390. [Online]. Available: https://aclanthology.org/D19-1242

[21] Y. Qiu, Y. Wang, X. Jin, and K. Zhang, *Stepwise Reasoning for Multi-Relation Question Answering over Knowledge Graph with Weak Supervision*. New York, NY, USA: Association for Computing Machinery, 2020, p. 474–482. [Online]. Available: https://doi.org/10.1145/3336191.3371812

[22] W. W. Cohen, H. Sun, R. A. Hofer, and M. Siegler, "Scalable neural methods for reasoning with a symbolic knowledge base," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=BJlguT4YPr

[23] A. Saxena, A. Tripathi, and P. Talukdar, "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: Association for Computational Linguistics, Jul. 2020, pp. 4498–4507. [Online]. Available: https://aclanthology.org/2020.acl-main.412

[24] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3982–3992. [Online]. Available: https://aclanthology.org/D19-1410

[25] M. Ali, M. Berrendorf, C. T. Hoyt, L. Vermue, S. Sharifzadeh, V. Tresp, and J. Lehmann, "PyKEEN 1.0: A Python Library for Training and Evaluating Knowledge Graph Embeddings," *Journal of Machine Learning Research*, vol. 22, no. 82, pp. 1–6, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-825.html

[26] T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," ser. ICML'16. JMLR.org, 2016, p. 2071–2080.

[27] Z. Huang, B. Li, and J. Yin, "Knowledge graph embedding via multiplicative interaction," ser. ICIAI '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 138–142. [Online]. Available: https://doi.org/10.1145/3194206.3194227

[28] A. Kristiadi, M. A. Khan, D. Lukovnikov, J. Lehmann, and A. Fischer, "Incorporating literals into knowledge graph embeddings," 2018. [Online]. Available: https://arxiv.org/abs/1802.00934