Knowledge Graph Augmentation for Increased Question Answering Accuracy

Jorge Martinez-Gil[1], Shaoyi Yin[2], Josef Küng[3], Franck Morvan[2]

[1]Software Competence Center Hagenberg GmbH
Softwarepark 32a, 4232 Hagenberg, Austria
jorge.martinez-gil@scch.at

[2]Paul Sabatier University, IRIT Laboratory
118 route de Narbonne, Toulouse, France
{shaoyi.yin,franck.morvan}@irit.fr

[3]Johannes Kepler University Linz
Altenbergerstraße 69, 4040 Linz, Austria
josef.kueng@jku.at

Abstract. This research work presents a new augmentation model for knowledge graphs (KGs) that increases the accuracy of knowledge graph question answering (KGQA) systems. In the current situation, large KGs can represent millions of facts. However, the many nuances of human language mean that the answer to a given question cannot be found, or it is not possible to find always correct results. Frequently, this problem occurs because how the question is formulated does not fit with the information represented in the KG. Therefore, KGQA systems need to be improved to address this problem. We present a suite of augmentation techniques so that a wide variety of KGs can be automatically augmented, thus increasing the chances of finding the correct answer to a question. The first results from an extensive empirical study seem to be promising.

Keywords: Expert Systems, Knowledge Engineering, Knowledge Graphs, Question Answering

1 Introduction

One of the most widely used representations of Knowledge Bases (KBs) is in the form of Knowledge Graphs (KGs). The nodes represent entities connected by relations in the form of a directed acyclic graph (DAG). Extensive research in the past decade has shown that these KGs can be extremely useful for many-core language tasks due to their simplistic structure and the ability to abstract facts and knowledge. Some disciplines that can benefit from KGs are question answering (QA) [12], recommender systems [10], etc.

In the case of QA, many systems depend on a suitable KG to find proper answers, so there is little any QA system can do if the KG does not contain the answer. Thus, larger KGs generally lend to better QA performance unless the question domain is orthogonal to the KG [17]. In this way, having the correct information appear in many different forms reduces the burden on the QA system to perform complex operations to understand the text. However, to date, this research direction has been little explored.

In the context of QA systems, chatbots, or voice assistants, one of the most usual ways of operating consists of using a KG (e.g., Wikidata) to be queried (e.g., using SPARQL) to find the correct answer to a given question [7]. The question can even be supplemented with synonyms to help users ask the same question differently [1]. The problem is that the question reformulation in different ways is always executed on the same structural model. If the answer structure the QA system is looking for is not in the KG, all attempts are futile.

For this reason, data augmentation strategies can be proposed. The problem is that data augmentation for explicitly graph-structured data is still in its early stages [26]. Our proposal represents one of the first solutions in this direction. The Wikidata KG is automatically augmented to significantly increase the chances of finding an answer for any query reformulations. In addition, our strategy applies to any KG developed under the Resource Description Framework (RDF) umbrella, where knowledge is represented through facts very close to natural language. Therefore, we aim to facilitate the development of better quality KGs in an automated way, at least from the QA viewpoint.

As a result, the main contributions of this research work can be summarized as follows:

- We present a new augmentation model for KGs intended to improve the performance of KGs for QA. Our strategy is based on several levels of augmentation that considerably reduce the possibility of false positives during the process.
- We empirically evaluate such a strategy to establish the corresponding improvements over baseline methods.

The rest of this paper is structured as follows: Section 2 presents the state-of-the-art in knowledge graphs augmentation. Section 3 presents the technical details of our contribution. Section 4 reports the empirical study to which we have subjected our strategy and compares it with baseline strategies. Furthermore, we end with conclusions and lessons that can be learned from this research work.

2 Related Works

Currently, the use of KGs is widespread since they are beneficial in computational disciplines that require the use and exploitation of background knowledge to develop their tasks. Since integrating heterogeneous sources has improved as computer and communication technology has advanced, the amount of background knowledge generated increases and changes continuously. New techniques based on neural computation are the most popular for implementing QA systems. However, in the literature, we see that neural approaches often lack the background knowledge to complete some of their tasks. In recent times, there has been some agreement in the community that this background knowledge is highly desirable. Furthermore, it has been proved that this background knowledge can be effectively represented in a structured form like fact triplets [21].

Therefore, it seems clear that background knowledge can help in new domains, especially when crucial external information is needed. However, little has been done to augment KGs directly [3]. Instead, the research on KGs mainly focuses on three open issues: KG representation, KG construction, and KG application, which integrates many computer-related disciplines such as knowledge representation, information retrieval, and natural language processing. For this reason, our work presents as a novelty, an exploration of the KG augmentation to increase the efficiency and reliability of QA systems.

The rest of this section is structured as follows. Subsection 2.1 sets out the current state of the art concerning QA. Subsection 2.2 discusses QA systems using KGs. Subsection 2.3 explains why current methods for Query Expansion are often not good enough to find the proper answers. Subsection 2.4 explains how augmentation techniques come to fill this gap. Finally, in subsection 2.5, we explain the significant differences between augmentation and auto-completion and clarify how our research is positioned in the current state-of-the-art.

2.1 Question Answering

QA has been successfully applied in several domains, such as search engines and voice assistants. The reason is that QA can facilitate applications to access knowledge effectively. QA systems are generally considered collections of interconnected components, usually, through a pipeline, that automatically analyze various data sources in order to answer questions [19]. Building successful QA systems is considered challenging due to the inherent problems in managing considerable amounts of data [11]. In recent times, QA systems have become prevalent as opposed to systems that rely on ranking methods to provide different resources to find information related to a question [15].

The development of QA systems involves many and varied problems [8]. However, in this work, we focus on the problem of data redundancy. The notion of data redundancy in massive collections, such as large textual corpora, means that information is likely to be phrased in just one of the many ways it would be possible.

Finally, it is essential to emphasize that we are only concerned with KGQA systems in this research work. This means we are interested in QA systems aiming to exploit KGs, which store facts about the world in a structured format. This kind of system has become very popular recently due to the good results they can achieve compared to classic unstructured text-based systems [2].

2.2 Question Answering over Knowledge Graphs

Question Answering over Knowledge Graphs (KGQA) aims to find answers for natural language questions over a KG [27]. Recent KGQA approaches adopt a neural machine translation approach, where the natural language question is translated into a structured query language [24].

The main difficulty comes from the nature of human language and the fact that text suffers from ambiguity in most cases. In addition, sentences and questions about the same topic and case can be formulated differently [18]. Language is very dynamic, and people can ask a question in almost infinite different ways. So, in most cases, specifying and providing a precise answer to a question is complicated.

KGQA systems are intended to convert a question into a query to a given KG, thus avoiding the need to learn a graph-oriented query language [25]. However, an insufficient amount of data is typical when exploiting current KGs. This is because collecting such amount of data can be tedious and error-prone. An effective way to work with KGs that are not large enough is to reformulate the original query by adding some synonyms to expand the search space to find a suitable answer. However, as we will see below, these methods also have some disadvantages.

2.3 Insufficiency of solutions for Query Expansion

One of the most popular methods in the QA domain is query expansion. Query expansion tries to augment a search query with other terms such as relevant synonyms or semantically related terms [5, 20]. This process is widely used in information retrieval systems to improve the results and increase recall [28]. Users either ask questions with minimal keywords that do not reflect the user intention or are inexperienced in the topic they are searching for. Therefore, query expansion is done assuming that the structure of the user query reflects the user's real intention. Moreover, the chance of finding a meaningful answer is low if no standard form is shared between the original question and how the knowledge is represented in the KG. Therefore, exploring methods to augment KGs automatically and without structural constraints seems reasonable.

2.4 Data Augmentation

Data Augmentation (DA) is a set of techniques that can be used to artificially expand the size of a dataset by creating modified data from the existing one [9]. For example, in the machine learning field, it is usually good to use DA to prevent overfitting or when the initial dataset is too small to train on or achieve better performance. In this way, DA is crucial for many applications as accuracy increases with available data.

It is widely assumed that DA can significantly improve tasks such as classification and segmentation accuracy in many domains. However, the use of DA techniques regarding KGs is little explored. In the context of KGs, DA can be formally defined as

$D \leftarrow (D, D_{aug}), D_{aug} \leftarrow \{\, (o, p_r, s) \,|\, \forall x \in D, x = (s, p, o)\}$

Although there are some precedents, DA is not as popular in working with textual information as in other domains. The reason is that augmenting textual data is challenging due to the nuances of the natural language [13]. In this way, generating new data samples is one way to augment existing data, but there is a high risk of creating incorrect knowledge. However, there are other safer approaches (e.g., minor changes to existing data, deterministic transformations, and so on) for DA.

2.5 Differences between KG Augmentation and KG Auto-Completion

Most existing approaches use some methods for auto-completion of KGs [4]. This task is usually called KG auto-completion. It extensively focuses on tackling this issue by learning models, commonly known as link predictors, that can complete any fact with partial information. More recently, neural network-based methods, commonly called neural link predictors, have become state-of-the-art for KG completion tasks. However, since these models are supervised learners, their ability is directly tied to the available training data.

Moreover, in KG auto-completion, the aim is to complete a fact, e.g., *(Socrates, is-a, ?)*. KG augmentation does not try to guess unknown information. However, it starts from the idea of making changes to the existing facts, e.g., *(Socrates, is-a, philosopher)* $\rightarrow$ *(Socrates, is-a, thinker), (Socrates, was-a, philosopher), (philosophy, is-contributed-by, Socrates), (Socrates, was-a, classic thinker)*, etc. While KG auto-completion is a machine learning challenge subject to much uncertainty, KG augmentation is a data management challenge. It is much more deterministic, preserving knowledge by minimizing the possibility of making serious errors. At the same time, it facilitates the accommodation of different formulations of a question only at the expense of needing more secondary memory (which is currently relatively cheap).

In this work, we adopt an approach based on KG augmentation. KG augmentation aims to expand the data sources to be analyzed to improve the limits of existing KGs. This strategy allows leveraging vast volumes of data to provide insights, forecasts, and suggestions previously unattainable owing to a lack of relevant data, even if starting with very little [29].

2.6 Positioning in the state-of-the-art

Much research work has been devoted to KG auto-completion. A common approach is KG embedding, representing entities and relations in triples as real-valued vectors and assessing triples' plausibility with these vectors. However, most KG embedding models only use structure information in observed triple facts. Furthermore, syntactic and semantic information in large-scale data is not fully utilized, as KG embeddings only employ entity descriptions, relation mentions, or word co-occurrence with entities [30]. Our research shows for the first time a strategy for KG augmentation that improves the quality of KGQA systems thanks to the automatic enrichment of the knowledge contained in the KG. To do this, we will try to make changes that are not too intrusive to save the veracity of the represented knowledge.

Our approach should be helpful in open-domain QA. For example, when dealing with factoid questions over KGs. A standard practice to execute a query using SPARQL over the KG to extract answer entities. Query expansion for this purpose adds a layer of difficulty. Our hypothesis is that a fully augmented KG might facilitate finding the correct answers by being much more flexible in allowing different formulations of the same or closely-related question.

3 Automatic Knowledge Graph Augmentation

Although QA systems are valuable and sought in various fields, developing efficient ways for assessing questions and determining responses is a difficult challenge. The fundamental challenge stems from the nature of human language and the fact that most texts can be reformulated without loss. Furthermore, questions concerning the same topic and instance might be phrased differently. People can ask a question in an almost limitless number of different ways. As a result, defining and providing a specific answer to a question is complex. Automatic KG augmentation can substantially reduce human effort and ensures the quality of machine-generated data using the results of performance improvements. It can also extend the relational information of given concept sets, including additional knowledge for the intended input query.

We can define a Knowledge Graph $\mathcal{KG} = \{(sub, pred, obj)\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ as a set of triples of the form $(sub, pred, obj)$ such that $sub \in \mathcal{E}$, $pred \in \mathcal{R}$ and $obj \in \mathcal{E}$. In this way, $\mathcal{E}$ and $\mathcal{R}$ are the sets of all entities and relation types of $\mathcal{KG}$.

Figure 1 shows an example of a KG in which facts in the form of subject, predicate, and object are modeled to give rise to a DAG that semantically models information.

For a given $\mathcal{KG} = \{(\mathcal{V}, \mathcal{E}, \mathcal{X})\}$ where $\mathcal{V}$ is a set containing $|\mathcal{V}|$ nodes, $\mathcal{E}$ is the set of edges showing the links between nodes, and $\mathcal{X}$ is the attribute matrix. A KG augmentation $Aug(\cdot)$ aims to learn a mapping function $\Phi : \Phi(\mathcal{V}, \mathcal{E}, \mathcal{X}) \mapsto \mathcal{R}^{|\mathcal{V}| \cdot \mathcal{V}}$ that projects graph nodes to d dimension latent representation $\mathcal{Z}$, where $\Phi : Aug(\Phi')$ being Φ' the mapping function.

While it is true that, in graph-oriented queries, the essential nodes are usually extended with synonymous words to facilitate cases where users set the question in different forms, we have adopted a radically different approach in this work that overcome specific weaknesses, e.g., safe creation of new fact triplets. KG augmentation is a set of methods to artificially increase the size of a KG by generating new facts from existing data. This includes making minor changes to data or using heuristics to generate new facts [29]. KG applications, especially in data and knowledge engineering, continue to increase quickly. KG augmentation

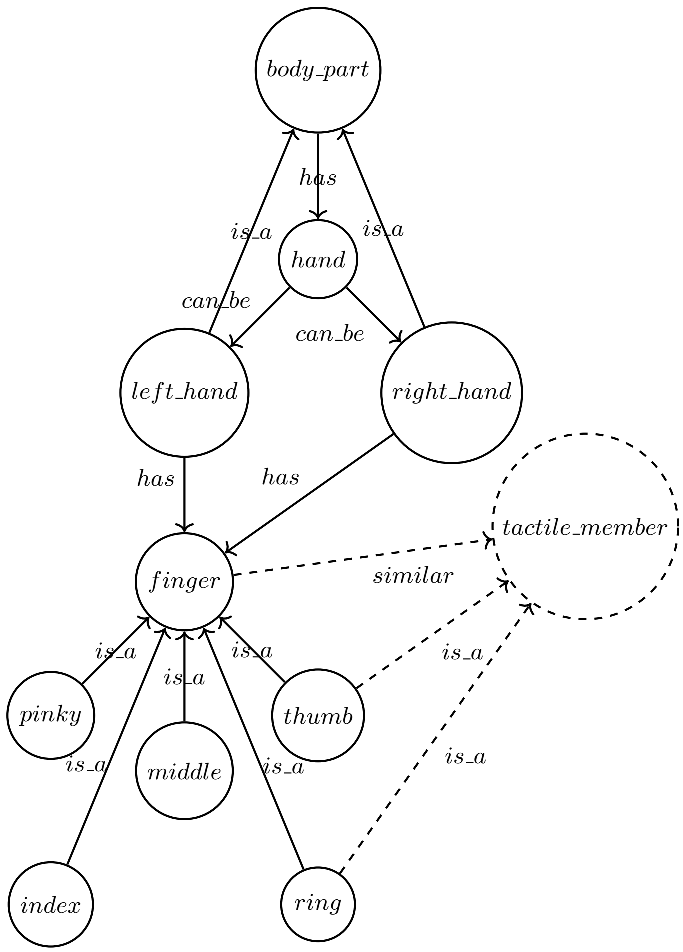

Fig. 1: Example of a Knowledge Graph representing some general-purpose knowledge. It consists of 12 fact triplets of type (subject, predicate, object). It can be easily and safely augmented with new facts that facilitates the answering of more questions. For example: (finger, equivalent, tactile_member), (ring, is_a, tactile_member), (thumb, is_a, tactile_member), etc. Such augmentation will be only at the expense of increased secondary memory consumption.

techniques may be an excellent tool to face the community's challenges. Manually curating and extending existing KG can lead to exhausting, error-prone, and costly processes. Transformations in KGs using augmentation techniques allow for reducing these operational costs. In this way, augmentation techniques make KGs more complete by creating diverse variations, and therefore, they facilitate finding the correct answer to a given question.

In this work, we consider four types of augmentation covering lexical, syntactic, and semantic aspects:

- Character-level augmentation
- Item-level augmentation
- Component-level augmentation
- An efficient combination of all of them

We will now explain each of them in detail. Subsequently, we will undertake an empirical evaluation that will allow us to explore the strengths and weaknesses of each of them.

3.1 Character-level augmentation

Character-level augmentation has been widely used because the augmentation mechanism is straightforward to realize. Compared to other augmentation strategies, these augmentation strategies generally have lower computational costs but can also achieve decent accuracy. Since such an augmentation strategy function does not consider graph structures, it can be used in large graphs. Accordingly, it has broader applications than the other strategies and can effectively improve the reliability of the KGQA systems.

The rationale behind this strategy is that variability is inherent to the human language. Therefore, this kind of KG augmentation can be understood as a way to proceed with a lemmatization process, determining the root of the words to prevent irregular forms (i.e., plurals, third persons, and so on).

3.2 Item-level augmentation

In order to enhance graph representation, item-level augmentation techniques generally employ information on target nodes and edges. Compared with character level augmentation, these techniques pay attention to the nodes in the graph. The universality is weaker than that of character-level strategies. Consequently, this strategy is more prevalent in recommendation systems and neural language processing. The most popular techniques in this category are synonym replacement, random token insertion, and random token deletion. Please note that this method is never applied to proper nouns because they usually have no substitute.

Synonym replacement It randomly chooses words from the subject or object that do not stop words. Replace each of these words with one of its synonyms selected at random. In order to do that, it should be possible to use word2vec [22] or contextual embeddings like BERT [6]. Dictionaries can also be used to find synonyms for the desired token from the original entity that must be replaced. The popular WordNet [23] is an example of a resource to be used here.

Random token insertion It consists of inserting a random word into the entity. Nevertheless, it is assumed that this insertion should not be a stop word. For these random insertions to make sense, they must be supported by a contextual embedding solution such as BERT [6], which can predict the words that usually appear next to a given one.

Random token deletion It consists of deleting a random word into the entity. It is assumed that the deletion should not be a stop word. For random deletion to make sense, it is preferable to proceed with adverbs or adjectives since they carry less semantic load than nouns.

3.3 Component-level augmentation

It consists of working on the level of a complete fact (subject, predicate, object). The most popular techniques in this category are: swapping and structure prediction.

Swapping It consists of choosing the subject and the object in the triplet and swapping their positions. In addition, the predicate must be adapted accordingly through a reversal operation. This operation is prevalent since it enables semantic swaps that preserve the global consistency [13].

Structure prediction Structured language in graph form helps achieve this augmentation capability. It can be done using any KG-auto completion technique since we will not rely on information already expressed.

3.4 An efficient combination of all of them

In augmentation research, a common technique combines several KG augmentation methods to achieve more diversified instances. Here, the combination can mean applying multiple separate or stacked methods. In this way, while the results of the two augmentation methods might differ significantly, combining both methods should produce good results.

It is essential to note that some combinations may be safer than others in preserving knowledge. For example, a reasonably safe combination can be made by choosing one type of augmentation from each branch, such that (*philosophers, like, reading)* could be replaced by *(read*, is-liked, thinkers)*. It can be seen how a character-level, synonym replacement, and swapping have been performed, respectively. The massive insertion of these triplets helps accommodate possible human operator queries. In addition, there is always the possibility of using more aggressive strategies if the original question cannot be satisfied. For example (*thinkers, enjoy, reading books)* whereby two synonym replacements and one random insertion have been considered.

4 Results

This section presents our results after subjecting our proposal to an extensive empirical study. So first, we describe the datasets we are going to work with. The configuration we have selected to perform the experiments, the empirical results we have obtained for the different variants of our strategy, and their appropriate comparison with baseline methods. Then, we show an empirical study on our performance when testing the strategy. Finally, we discuss the results we have obtained.

4.1 Datasets

The KG that we will use as a base to be augmented will be Wikidata[1]. This corpus consists of the entire Wikidata KG, and it takes 33 GB, approx. Although this is due to overhead, the amount of net data is significantly lower. At the same time, we have chosen some questionnaires (i.e., adapted pairs of questions and answers) on geography and history, which have been taken from a subset of the popular OpenTrivia benchmark dataset [14]. The reason for choosing this dataset is that it contains primarily questions in factoid format, which in principle, is the most suitable format expected for a KGQA-based system. Nevertheless, since ours is a general-purpose framework, there would be no restriction to operating on other single-response datasets.

4.2 Setup

The configuration we have chosen is as follows:

- *Character level*, we have chosen the Krovetz solution [16].
- *Item Level - Synonym Replacement*, using the synonyms ranked first in Wordnet.
- *Item Level - Random Insertion*, using the first prediction of BERT.
- *Item Level - Random Deletion*, a token is erased randomly when the entity has more than one token.
- *Component-level - Swapping*, subject and object are interchanged, and the predicate is switched to passive voice.

[1] https://www.wikidata.org/

– *Component Level - Structure prediction*, calculated with TransE from the Ampligraph library[2]
– *An efficient combination of all of them*, we calculate all possible permutations, choosing one method from each of the three families at each iteration. We report only the best result.

There is one crucial thing to keep in mind at this point. Augmentation techniques may or may not be associated with worse results than baseline. For example, work with the augmented copies only to find answers to questions not satisfied by the original KG. The result will never be inferior to the baseline. However, running the program will require much more time since it has to navigate between several augmented copies. Nevertheless, in no case will the human operator get worse results.

In another case, the queries are performed directly on the augmented KG to check if they can be satisfied. In these cases, the execution time will be very similar (although it depends significantly on the degree of augmentation of the original KG). However, it is quite true that the results may vary significantly concerning the baseline.

In this work, we take as a form of evaluation the first one. The answers are sought in augmented copies of the KG only if they could not be satisfied initially.

4.3 Empirical Evaluation

Below, we show our results after submitting our different strategies to an empirical evaluation. The baseline consists of using the original Wikidata KG. Furthermore, each of the following strategies works on the original Wikidata KG to check if such augmentation can facilitate better results. We also show the augmentation factor, which means the amount of memory space required to store the augmented KG, being 1.00 the original Wikidata KG.

Table 1 shows the results for the questions on general geography. Sometimes it is challenging to know specific data about geography. We now want to see if our proposal could help a human operator satisfactorily.

Method	**Score**
Baseline (Original Wikidata)	0.57
Augm-Character Level - Krovetz Lemmatization	0.62
Augm-Item Level - Synonym Replacement	0.66
Augm-Item Level - Random Insertion	0.57
Augm-Item Level - Random Deletion	0.47
Augm-Component Level - Swapping	0.57
Augm-Component Level - Structure prediction	0.62
Augm-Smart Combination	**0.67**

Table 1: Results for the subset of general geography from OpenTrivia

[2] https://github.com/Accenture/AmpliGraph/

Table 2 shows us the results on the questions about the history of humankind, regardless of date or geographical region. It is not easy for a human operator to store all this encyclopedic knowledge.

Method	Score
Baseline (Original Wikidata)	0.57
Augm-Character Level - Krovetz Lemmatization	0.62
Augm-Item Level - Synonym Replacement	0.66
Augm-Item Level - Random Insertion	0.57
Augm-Item Level - Random Deletion	0.46
Augm-Component Level - Swapping	0.57
Augm-Component Level - Structure prediction	**0.71**
Augm-Smart Combination	0.70

Table 2: Results for the subset of general history from OpenTrivia

Table 3 shows us the augmentation factor, which means the amount of memory space required to store the augmented KG, being 1.00 the original Wikidata KG. Please note that we have taken a conservative approach. For example, in the case of swapping, three operations could be performed on subject, predicate and object respectively. However, to guarantee the preservation of knowledge, we have only performed on one permutation at a time.

Method	Aug.
Baseline (Original Wikidata)	1.00
Augm-Character Level - Krovetz Lemmatization	0.98
Augm-Item Level - Synonym Replacement	2.53
Augm-Item Level - Random Insertion	1.26
Augm-Item Level - Random Deletion	0.86
Augm-Component Level - Swapping	1.01
Augm-Component Level - Structure prediction	1.33
Augm-Smart Combination	1.77

Table 3: Size of the Wikidata KG after undergoing augmentation operation

4.4 Discussion

Methods and techniques for automatically answering questions are in high demand. As a result, many solutions for QA have been developed to respond to this need. In this context, building models that seamlessly handle structured data (e.g., KG) has been a long-sought-after goal. The reason is that KGs allow overcoming limitations about the structure and semantics of the information they represent. We have seen how the research community has proposed efficient methodologies for analyzing questions and specifying proper answers using KGs with quite reasonable returns. In this way, the advantages inherent in the use of KG in contexts, such as the one we are dealing with here, can be summarized as follows:

- Improving KGQA accuracy by safely adding more data into the KGs
- Reducing costs of manually augmenting the KGs
- Facilitating answers to rare questions
- Preventing privacy issues (if needed)

The main issue faced when working with augmented KGs is that the model may need much more computation time to find no answer. Most KG augmentation learning methods focus on homogeneous graphs, including augmenting nodes, structural attributes, or models. However, the continuous development of KG learning methods is still difficult to handle the problem of heterogeneity in graph data. For this reason, we will now list the lessons we have learned from this research work:

- It is always possible to try different augmentation approaches and check which works better
- An aggregation of different augmentation methods is also a good idea
- It is possible to determine the optimal method combination for the best results
- Data augmentation in KGs does not always help to improve the performance

Furthermore, KG augmentation can be beneficial for increasing the accuracy of KGQAs. However, it does lead to an increase in the use of secondary memory. Nevertheless, this type of memory does not usually represent a high cost in recent times. In addition, it would be possible to look for data management techniques that optimize the consumption of resources. Furthermore, a possible strategy of combining techniques for query expansion and KG augmentation simultaneously could also help meet this challenge.

5 Conclusions

QA systems have become more critical than ever in recent years. The main reasons are the ongoing expansion in the available information and the necessity to help people get the information they need precisely, quickly, and efficiently. We have seen how KG augmentation is crucial for building more accurate and robust KGs. An appropriate pre-processing with data augmentation can help build state-of-the-art systems.

This research has presented our approach to using natural language generation for data-centric research to reduce the cost of leveraging building an augmented KG. Automatic knowledge augmentation for KGQA systems is not limited to a particular model but can be applied in different forms. Our approach demonstrates the positive effects of KG augmentation through comparative experiments using datasets belonging to different domains. Moreover, some guidelines for suitable and feasible KG augmentation strategies have been provided.

In future work, the community must consider the necessity to develop evaluation methodologies for measuring the quality of augmented KGs. As the use of augmentation methods increases, assessing their quality will be required. It is also important to consider that if an actual KG has biases, a KG augmented from it will also have those biases. So, the identification of a good KG augmentation approach is essential.

Furthermore, it is also necessary to consider that augmentation models are generally time-consuming and have some space complexity. As the number of nodes or edges increases in large-scale graphs, the augmentation factor will also increase. Nevertheless, until now, there has been no effective parallel solution for handling this issue. The problem of high cost, the selection of augmentation strategy, and the optimization of the augmentation model are the main problems that need to be faced in the future.

Acknowledgements

The authors thank the anonymous reviewers for their help in improving the work. This work has been supported by the Austrian Ministry for Transport, Innovation and Technology, the Federal Ministry of Science, Research and Economy, and the State of Upper Austria through the COMET center SCCH. And by the project FR06/2020 - International Cooperation & Mobility (ICM) of the Austrian Agency for International Cooperation in Education and Research (OeAD-GmbH). We would also thank 'the French Ministry of Foreign and European Affairs' and 'The French Ministry of Higher Education and Research' which support the Amadeus program 2020 (French-Austrian Hubert Curien Partnership – PHC) Project Number 44086TD.

References

1. H. K. Azad and A. Deepak. Query expansion techniques for information retrieval: A survey. *Inf. Process. Manag.*, 56(5):1698–1735, 2019.
2. J. Berant, A. Chou, R. Frostig, and P. Liang. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1533–1544. ACL, 2013.
3. M. Cannaviccio, L. Ariemma, D. Barbosa, and P. Merialdo. Leveraging wikipedia table schemas for knowledge graph augmentation. In *Proceedings of the 21st International Workshop on the Web and Databases, Houston, TX, USA, June 10, 2018*, pages 5:1–5:6. ACM, 2018.
4. Z. Chen, Y. Wang, B. Zhao, J. Cheng, X. Zhao, and Z. Duan. Knowledge graph completion: A review. *IEEE Access*, 8:192435–192456, 2020.

5. S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *J. Am. Soc. Inf. Sci.*, 41(6):391–407, 1990.
6. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
7. D. Diefenbach, T. P. Tanon, K. D. Singh, and P. Maret. Question answering benchmarks for wikidata. In N. Nikitina, D. Song, A. Fokoue, and P. Haase, editors, *Proceedings of the ISWC 2017 Posters & Demonstrations and Industry Tracks co-located with 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 23rd - to - 25th, 2017*, volume 1963 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2017.
8. E. Dimitrakis, K. Sgontzos, and Y. Tzitzikas. A survey on question answering systems over linked data and documents. *J. Intell. Inf. Syst.*, 55(2):233–259, 2020.
9. S. Y. Feng, V. Gangal, J. Wei, S. Chandar, S. Vosoughi, T. Mitamura, and E. H. Hovy. A survey of data augmentation approaches for NLP. In C. Zong, F. Xia, W. Li, and R. Navigli, editors, *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 968–988. Association for Computational Linguistics, 2021.
10. Q. Guo, F. Zhuang, C. Qin, H. Zhu, X. Xie, H. Xiong, and Q. He. A survey on knowledge graph-based recommender systems. *IEEE Transactions on Knowledge and Data Engineering*, 2020.
11. A. Hameurlain and F. Morvan. Big data management in the cloud: evolution or crossroad? In *Beyond Databases, Architectures and Structures. Advanced Technologies for Data Mining and Knowledge Discovery*, pages 23–38. Springer, 2015.
12. L. Hirschman and R. J. Gaizauskas. Natural language question answering: the view from here. *Nat. Lang. Eng.*, 7(4):275–300, 2001.
13. L. Huang, L. Wu, and L. Wang. Knowledge graph-augmented abstractive summarization with semantic-driven cloze reward. In D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5094–5107. Association for Computational Linguistics, 2020.
14. M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, 2017.
15. O. Kolomiyets and M. Moens. A survey on question answering technology from an information retrieval perspective. *Inf. Sci.*, 181(24):5412–5434, 2011.
16. R. Krovetz. Viewing morphology as an inference process. *Artif. Intell.*, 118(1-2):277–294, 2000.
17. Y. Lan, G. He, J. Jiang, J. Jiang, W. X. Zhao, and J. Wen. A survey on complex knowledge base question answering: Methods, challenges and solutions. In Z. Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 4483–4491. ijcai.org, 2021.
18. J. Martinez-Gil and J. M. Chaves-Gonzalez. Semantic similarity controllers: On the trade-off between accuracy and interpretability. *Knowledge-Based Systems*, page 107609, 2021.

19. J. Martinez-Gil, B. Freudenthaler, and A. M. Tjoa. A general framework for multiple choice question answering based on mutual information and reinforced co-occurrence. *Trans. Large Scale Data Knowl. Centered Syst.*, 42:91–110, 2019.
20. J. Martinez-Gil, R. Mokadem, J. Küng, and A. Hameurlain. A novel neurofuzzy approach for semantic similarity measurement. In M. Golfarelli, R. Wrembel, G. Kotsis, A. M. Tjoa, and I. Khalil, editors, *Big Data Analytics and Knowledge Discovery - 23rd International Conference, DaWaK 2021, Virtual Event, September 27-30, 2021, Proceedings*, volume 12925 of *Lecture Notes in Computer Science*, pages 192–203. Springer, 2021.
21. J. Martinez-Gil, R. Mokadem, F. Morvan, J. Küng, and A. Hameurlain. Interpretable entity meta-alignment in knowledge graphs using penalized regression: a case study in the biomedical domain. *Prog. Artif. Intell.*, 11(1):93–104, 2022.
22. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119, 2013.
23. G. A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, 1995.
24. A. Perevalov, D. Diefenbach, R. Usbeck, and A. Both. Qald-9-plus: A multilingual dataset for question answering over dbpedia and wikidata translated by native speakers. In *16th IEEE International Conference on Semantic Computing, ICSC 2022, Laguna Hills, CA, USA, January 26-28, 2022*, pages 229–234. IEEE, 2022.
25. T. Ploumis, I. Perikos, F. Grivokostopoulou, and I. Hatzilygeroudis. A factoid based question answering system based on dependency analysis and wikidata. In N. G. Bourbakis, G. A. Tsihrintzis, and M. Virvou, editors, *12th International Conference on Information, Intelligence, Systems & Applications, IISA 2021, Chania Crete, Greece, July 12-14, 2021*, pages 1–7. IEEE, 2021.
26. C. Shorten, T. M. Khoshgoftaar, and B. Furht. Text data augmentation for deep learning. *J. Big Data*, 8(1):101, 2021.
27. N. Steinmetz and K. Sattler. What is in the KGQA benchmark datasets? survey on challenges in datasets for question answering on knowledge graphs. *J. Data Semant.*, 10(3-4):241–265, 2021.
28. C. Xiong and J. Callan. Query expansion with freebase. In J. Allan, W. B. Croft, A. P. de Vries, and C. Zhai, editors, *Proceedings of the 2015 International Conference on The Theory of Information Retrieval, ICTIR 2015, Northampton, Massachusetts, USA, September 27-30, 2015*, pages 111–120. ACM, 2015.
29. S. Yu, H. Huang, M. N. Dao, and F. Xia. Graph augmentation learning. *arXiv preprint arXiv:2203.09020*, 2022.
30. Z. Zhao, T. Liu, S. Li, B. Li, and X. Du. Ngram2vec: Learning improved word representations from ngram co-occurrence statistics. In M. Palmer, R. Hwa, and S. Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 244–253. Association for Computational Linguistics, 2017.