WIRTSCHAFTS
UNIVERSITÄT
WIEN VIENNA
UNIVERSITY OF
ECONOMICS
AND BUSINESS

EFMD
EQUIS
ACCREDITED

Bachelor Thesis

Performance Evolution of In-Knowledge Graph Tasks: A Structured Literature Review

Berndt Marc Rausch

Date of Birth: 06.03.1998
Student ID: 11778962

Subject Area: Information systems

Studienkennzahl: h11778962

Supervisor: Prof. Dr. Axel Polleres

Co-Supervisor: Stefan Bachhofner, MSc. (WU)

Date of Submission: 30.August 2023

Department of Information Systems and Operations, Vienna University of Economics and Business, Welthandelsplatz 1, 1020 Vienna, Austria

DEPARTMENT FÜR INFORMATIONS-VERARBEITUNG UND PROZESS-MANAGEMENT DEPARTMENT OF INFORMATION SYSTEMS AND OPERATIONS

Contents

Acknowledgements

I am thanking the Institute for Data, Process, and Knowledge Management at the WU and our supervisor, Prof. Axel Polleres. I am grateful for the technological support that the language model ChatGPT delivers. It supported me with the resolution of additional structural questions and in the area of data analysis. Special thanks to my co-supervisor, Stefan Bachhofner, M.Sc., for the constructive feedback and the continuous support. I am very thankful for the support of my family, friends, and girlfriend. Without them, it would not be possible to finish my work on this bachelor's thesis.

Abstract

The goal of knowledge graph completion is to predict new information within a knowledge graph. Researchers increase their focus on knowledge graph embedding methods for the completion process and neglect traditional knowledge completion methods. Despite that, there is no direct comparison of the historical performance differences between embedding methods and traditional knowledge graph completion methods in literature. This includes the not researched extent of performance increase that the embedding method can potentially provide. To address this problem, we use a systematic literature review to study the performance of these two groups. We are interested in a historical performance comparison that indicates whether there is a difference between them. The analyses of our performance data results shows a better performance for none embedding methods than embedding methods. The results indicate that knowledge graph embedding methods provide no performance advantage and do not justify the greater extent of research investment in current literature.

Keywords: Knowledge graph, link prediction, triple classification, entity classification, entity resolution, embedding, completion.

Section/topic	#	Checklist item	Location(s) Reported
INFORMATION SOURCES AND METHODS			
Database name	1	Name each individual database searched, stating the platform for each.	4.1 Search strategy
Multi-database searching	2	If databases were searched simultaneously on a single platform, state the name of the platform, listing all of the databases searched.	4.1 Search strategy
Study registries	3	List any study registries searched.	N/a
Online resources and browsing	4	Describe any online or print source purposefully searched or browsed (e.g., tables of contents, print conference proceedings, web sites), and how this was done.	N/a
Citation searching	5	Indicate whether cited references or citing references were examined, and describe any methods used for locating cited/citing references (e.g., browsing reference lists, using a citation index, setting up email alerts for references citing included studies).	4.2 Study selection
Contacts	6	Indicate whether additional studies or data were sought by contacting authors, experts, manufacturers, or others.	N/a
Other methods	7	Describe any additional information sources or search methods used.	N/a
SEARCH STRATEGIES			
Full search strategies	8	Include the search strategies for each database and information source, copied and pasted exactly as run.	4.1 Search strategy
Limits and restrictions	9	Specify that no limits were used, or describe any limits or restrictions applied to a search (e.g., date or time period, language, study design) and provide justification for their use.	4.3 Data extraction
Search filters	10	Indicate whether published search filters were used (as originally designed or modified), and if so, cite the filter(s) used.	4.1 Search strategy
Prior work	11	Indicate when search strategies from other literature reviews were adapted or reused for a substantive part or all of the search, citing the previous review(s).	N/a
Updates	12	Report the methods used to update the search(es) (e.g., rerunning searches, email alerts).	4.1 Search strategy
Dates of searches	13	For each search strategy, provide the date when the last search occurred.	N/a
PEER REVIEW			
Peer review	14	Describe any search peer review process.	N/a
MANAGING RECORDS			
Total Records	15	Document the total number of records identified from each database and other information sources.	**440**
Deduplication	16	Describe the processes and any software used to deduplicate records from multiple database searches and other information sources.	4.4 Outline and scope

PRISMA-S: An Extension to the PRISMA Statement for Reporting Literature Searches in Systematic Reviews
Rethlefsen ML, Kirtley S, Waffenschmidt S, Ayala AP, Moher D, Page MJ, Koffel JB, PRISMA-S Group.
Last updated February 27, 2020.

1 Introduction

Everybody gets in contact with some knowledge graph applications in their daily modern lives. Big tech companies like Google or Amazon adopt these technologies and revolutionize the technological landscape. These companies build their own knowledge graphs through existing knowledge repositories [46, 12]. The data structure within a knowledge graph is a triple and looks like (Christoph Waltz, bornIn, Vienna) [22].

Over the past few years, research tends to focus on the generation or prediction of knowledge within a knowledge graph without the input of new external information. This process is knowledge graph completion, as its purpose is to get closer to a completion of a knowledge graph [80]. In-knowledge graph tasks predict new knowledge through prior existing ones, like (Christoph Waltz, ageIs, 66) leads to the prediction of (Christoph Waltz, bornIn, 1956) [59]. The knowledge graph completion process can be carrie out by embedding methods or none embedding methods.

Despite the continuously increasing amount of work in the research direction of knowledge graphs, the performance difference between embedding and none embedding methods represents a gap in the literature. This leads to a missing historic comparison that shows the performance disparity over time.

In this thesis, we aim to give a visual representation of the historic performance difference between embedding and none embedding methods. Our hypothesis is that embedding methods possess a slight performance advantage at the current stage, but not enough to explain the sole focus and switch to embedding methods in research.

The paper structures into the following sections: in Section 1, we present our acknowledgements. Section 2 presents an introduction to our topic and includes our research question and research method. Section 3 contains the background for knowledge graphs, in-knowledge graph tasks, traditional knowledge graph completion, and knowledge graph embedding. In Section 4, we introduce the related work for the systematic literature review. Section 5 is about the approach we took to visualize and compare our results. In Section 6, we present the results of our study. Section 7 discusses our results and future studies. In the last section, we summarize the most important findings into a conclusion.

1.1 Research question

This thesis contributes to the literature on knowledge graphs by studying the performance difference between knowledge graph embedding methods and methods that do not use embedding. In particular, we study whether there is a difference in performance for in-knowledge graph tasks between embedding methods and none embedding methods. The research questions in this thesis are:

> *Q1: Is there a difference in performance between embedding and none embedding methods for in-knowledge graph tasks?*
>
> *Q2: If there is one, to what extent does the performance differ between embedding and none embedding in-knowledge graph tasks, and is it justifying the additional investment that knowledge graph embedding entails?*

1.2 Research method

We study the research questions from above with a systematic literature review. As a guideline, we use the PRISMA or preferred reporting items, for systematic reviews and meta-analyses [42]. This research method enables us to determine, select, and link the present literature for knowledge graphs and knowledge graph completion. The process of a systematic literature review includes many steps. These steps are to identify, structure, and formulate the research questions. Then create the outline and scope for the search strategy, inclusion and exclusion criteria, and selection strategy. To conduct a systematic search of the literature with our selection of databases. Then we further reduce the number of articles. The next step is to filter the selection of papers according to the inclusion and exclusion criteria. Then assess the remaining full-text articles and include or exclude them from the qualitative synthesis. The last step is that we extract the relevant data from the remaining literature, like knowledge graph completion performance data and background information.

2 Background

This section presents background on Knowledge Graphs (KG), In-Knowledge Graphs (In-KG) tasks, Knowledge Graph Completion (KGC), Knowledge Graph Embedding (KGE) methods, and None Knowledge Graph Embedding (NKGE) methods.

2.1 Knowledge graph

The origin of KGs, or its synonym, knowledge base, dates back to the late 1960s with the development of expert systems [12, 27, 22]. KGs purpose is to collect real-world data and knowledge into a large-scale network and structure human knowledge as a multiple relational graph [38, 48, 11, 68]. In the 1980s, Tim Berners-Lee et al. suggested the Semantic Web. The Semantic Web is about the integration of relations between concepts [18, 27, 1]. Stokman and Vries proposed the modern knowledge graph in 1988 [27]. In 2012, Google's search engine use of KG technology made it popular [18, 27]. The current literature on KGs focuses on research in these three categories: knowledge graph representation, knowledge graph construction, and knowledge graph applications [12]. Big tech companies such as Google, Facebook, Amazon, etc. adopt KGs to build their own and utilize them in their business operations, such as search engines [46, 12].

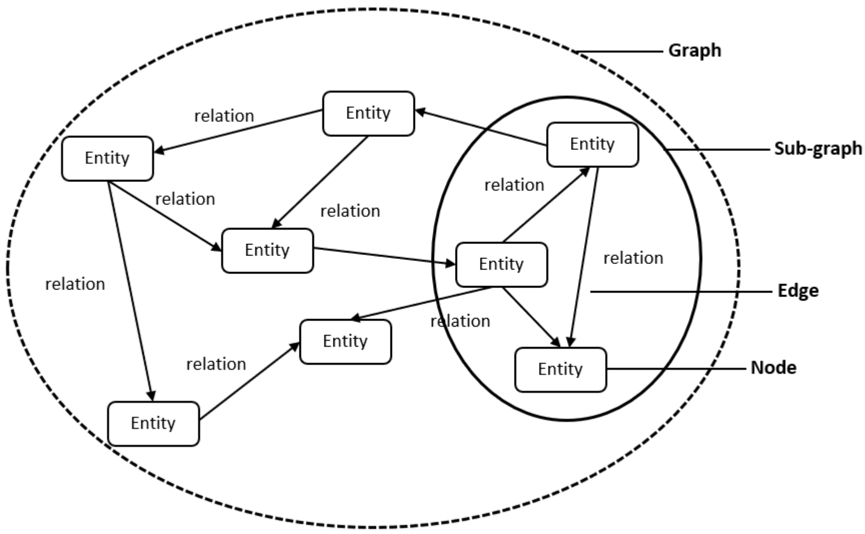

Figure 1: This is the structure of a graph. It consists of many nodes with edges as connections. A node is an entity, and the edge is the relationship between two entities. A graph contains many sub-graphs. A sub-graph within describes only part of the graph.

The Figure 1 represents an example of a KG structure. New applications for such KGs increase the adaption rate of companies. They solve real-world problems like semantic parsing, search engines, entity disambiguation, information extraction, or question answering [49, 84, 38, 39]. These applications are natural language processing tasks (NLP) and support the enhancement of AI applications [37, 67, 79, 7]. One use case for KGs is a modern search engine (e.g., Google or Bing). These search engines or other technologies like machine learning utilize the structure to improve search accuracy [46, 17]. This increases adoption and importance for companies and institutes [17]. The massive amount of linked data leads to a higher popularity of the technology [87]. User interest graphs are real-life applications of KGs. The basic information is from knowledge repositories, and they contain information about publicly known figures, e.g., actors, companies, and politicians [46].

The Google Knowledge Graph uses data from knowledge repositories like Freebase. Those repositories are also KGs, but are publicly available. This leads to KGs being built from other KGs [46]. These public repositories contain millions of entities and relations. The most popular in the literature are Freebase, DBpedia, YAGO, WordNet, NELL [86, 19, 65, 83]. Metaweb released the knowledge repository Freebase in 2007, and in 2010, Google bought it. It has around 3 billion facts and 50 million entities [18]. DBpedia, a community project by the Leipzig University and Free University of Berlin research team, released in 2007 as OpenLink software [18]. Yet another great ontology (YAGO) published by the Max Planck Institute in 2007, has 5 million facts about individuals, organizations, and locations [18]. As there are many different knowledge bases, there need to be classifications for each variant.

KGs classifies into several categories, in this paper we focus on 3 main variations containing sub-categories. The first classification is the visualization of information. The content displays either as textual, visual, or multi-modal. The next class is the domain scope. It splits between two sub-categories of general KG and a domain KG. The last one is about the timeliness of information. The supply of new information can either be static or dynamic. A static KG includes no new outside information, whereas a dynamic one constantly receives new outside input [12, 27]. This information, has the form of a triple, and we describe this structure of information storage in the next paragraph [13].

Figure 2: An example of a triple with the entities Vienna and Austria and the relation between them, capitalOf.

Triples The structure of knowledge graphs is a heterogeneous information network and consists of a set of triples [13, 17, 19, 18]. With Figure 1 we present a KG consisting of triples and a sub-graph. KGs are directed graphs [14, 85, 7]. Triples storage structured human knowledge [80]. Another name for triple is fact. A knowledge graph represents a fact through a subject and an object. The connection between them is called a predicate or relation [19, 65, 27]. This leads to (subject, predicate, object) [48, 68, 59]. The Figure 2 presents the triple (Vienna, capitalOf, Austria). Vienna is the subject with capitalOf as predicate and Austria as the object [50, 44, 38]. The descriptions or terms of a triple differ in the literature. One variation describes the subject as head entity or h, the relation or predicate between those entities as r, and the object as tail entity or t [16, 44]. These triples stating a fact as (h, r, t) and posses the same structure as (subject, predicate, object) [50, 84, 26, 71]. This means that an entity can either be a real-world object or an abstract concept with the relation between them [27]. The other common variant is with the name nodes for both subject and object and edge for relation. This structure of triples enables efficient representation of data in KGs. [65, 62, 21]. The relations of triples split into four classes with 1-to-1 relation, 1-to-N, N-to-1, and N-to-N [22]. In the next paragraph, we elaborate the difference between Resource description framework and labeled property graph.

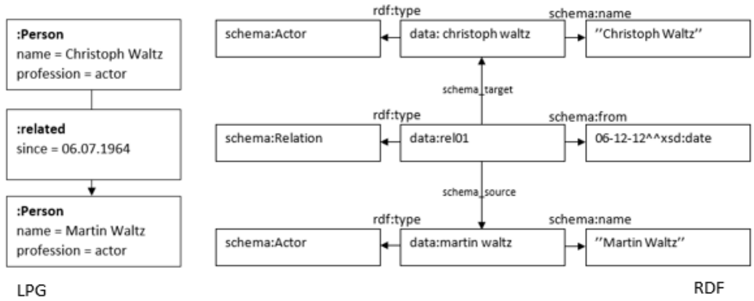

Figure 3: These two graphs represent a labelled property graph on the left and a resource description framework on the other side.

Resource description framework and labeled property graph Resource description framework (RDF) and labelled property graph (LPG) are information structures of KGs. In Figure 3 we present a visual comparison between the LPG graph and a RDF graph. The RDF methods use triples to store and represent knowledge [18, 58]. In our case, we present the relations of Christoph Waltz in a potential sub-graph as a RDF graph. A triple based on the RDF has the same structure as the triples above (subject, predicate, object) and supports semantics [18, 53, 51, 52]. RDF is one of the semantic web standards since 1999 [27]. Property graphs include properties or attributes of entities and relations. In our example, we have a :Person with the name = Christoph Waltz and the profession = actor. The :Person is the entity with the properties of name and profession. Another name for a property graph is an attribute graph. Entities in LPG have semantic descriptions, and relations possess types like since = 06.07.1964 [27]. In the next paragraph, we explain the construction of KGs.

The heterogeneous data construction of such sets of triples is a knowledge graph or knowledge base [50, 51]. As a dynamic KG includes external data, a static KG can only create new knowledge with a knowledge graph completion process utilizing the already existing triples [12, 63, 68]. The graph structure of KG contains information as triples. Relational reasoning utilizes existing triples to construct new potential relations between entities and create new triples within the KG. Different terms for relational reasoning are entity prediction, relational prediction, or knowledge graph completion [63]. Technologies like knowledge graph reasoning applications enhance the structure of these graphs. These applications use the existing entities to relation structure to predict new triples [63]. KGs increase in usage in modern situations. They are practical and essential, but have problems that can hinder their efficiency, like the high need for computing power, data sparseness, increasing size, and limited data storage space [4, 66, 73, 47]. The next section contains information about the knowledge graph completion process.

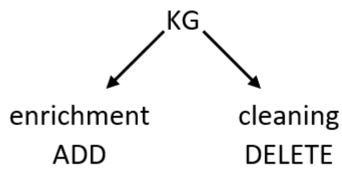

Figure 4: Knowledge graph completion has two categories of sub-tasks. The first is knowledge graph enrichment. This task adds triples to the knowledge graph. The second is cleaning, and it deletes false triples.

2.2 Knowledge graph completion

Knowledge graphs store millions of entities but still contain incomplete and incorrect facts [52, 22, 48, 68]. As the Figure 4 shows, the KGC process either enriches a KG or cleans it with certain in-KG tasks. To reduce these issues and improve the quality of KGs the KGC process predicts new triples to complete a KG [51, 62, 73, 48]. The prediction process needs the prior existing triples to find new ones [16]. The completion of a triple either predicts the head entity (?, r, t), relation (h, ?, t) or tail entity (h, r, ?) [22, 47, 37]. A scoring method is frequently used in the literature for KGC [24, 7]. It measures the plausibility of missing triple entities or relations and supports the completion process [71, 47]. Currently, the literature focuses on the KGC of static KGs [52, 73]. It is too costly and labor-intensive to manually add valid triples [80, 28]. This leads to the need for amortization like in-KG tasks [80, 14]. The next paragraph focuses on the different evaluation protocols for KGC.

Evaluation protocol To test and evaluate the performance of in-KG tasks on data-sets, the literature uses evaluation metrics for experiments. The main metrics are mean rank (MR), mean reciprocal rank (MMR), and HITS@K [1]. These metrics rank test triples to evaluate performance [16]. The evaluation uses a score function to rank triples [70]. MR is the most basic and describes average rank of these ranked test triples [1]. A lower score indicates better results [70]. MMR has fewer outliers than MR, and the scores are between 0 and 1. Results closer to 1 signify better performance [1]. The HITS@K has different settings, with the most common as HITS@10 [1]. HTIS@K measures the proportion of the test triples that are positive and rank in the k [70, 31]. The k can be a number, with the most common as 10 [70]. The standard evaluation protocol for classification tasks like EC or TC is mean average precision (MAP) [71, 70]. Other metrics are adjusted mean rank (AMR), receiver operating characteristic curve (AUC-ROC), and area under the precision-recall curve (AUC-PR) [1]. But AUC-ROC and AUC-PR need complete KGs and are hence not useful for in-KG tasks [1]. To improve upon the performance of the raw version, the literature filters valid triples and excludes corrupted ones [59]. This leads to the options of raw and filter version [16]. The literature uses filter more often as it has better performance [1]. In the upcoming subsection, we talk about the individual In-KG tasks and the way they contribute to the KGC process.

2.3 In-knowledge graph tasks

In-KG tasks are part of the entity and relation embedding scope, with the goal to complete and refine the KG [65, 59]. In other words, the tasks predict missing relations or entities [17, 82]. These In-KG tasks either complete a KG or refine them [79]. These tasks are link prediction, triple classification, entity classification, entity resolution, and entity prediction [65, 27, 68]. The tasks to reduce the incompleteness of KGs are link prediction and entity prediction [80]. The remaining tasks of triple classification refine a KG [28]. Link prediction is the first task we describe, and it is the most popular In-KG task in literature [5].

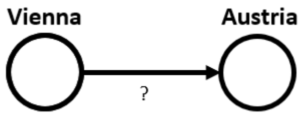

Figure 5: The link prediction task has the goal to predict the missing relation (?) between Vienna and Austria through a scoring method.

Link prediction The purpose of link prediction is to predict missing relations between two entities [68, 80, 5]. The entities to predict are either a missing head entity or a tail entity within a incomplete triple [9, 15, 6, 26]. This task adds additional information to the KG and increases completion and refinement [65]. The link prediction process uses a ranking or scoring system to predict the two entities with a matching relation [65]. The scoring methods return a list of ranked candidates based on their score [9, 59]. The process extracts incomplete triples within the KG. In our example (Figure 5) the triple is (Vienna, ?, Austria) with the relation between the entities missing. The scoring function or algorithm of link prediction trains through a KG [61]. It creates a list with a ranking of possible entities matching a relation. This leads to a prediction of a link between two entities [65, 63]. Another name for link prediction in the literature is relation prediction [29]. Link prediction of missing relation leads to the completion of the KG. The next in-KG task checks the correctness of these new triples.

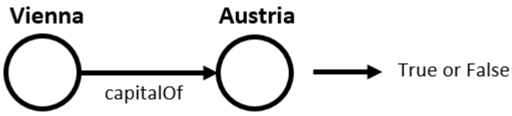

Figure 6: Triple classification or fact prediction calculates the correctness of a triple and classifies them as either True or False.

Triple classification Triple classification is another in-KG task. Triple classification classifies a triple as true or false [9, 66, 11, 68]. The literature considers this as a binary task as it is either true or false [85, 61]. A synonym for a true triple in the literature is golden triple [85]. The previous task predicts through a scoring method the missing relation as "capitalOf". The new triple is (Vienna, capitalOf, Austria) as Figure 6 above [85]. Now, triple classification calculates for every triple a score. The higher the score of a triple, the greater the chance that the fact is true and states the plausibility of triple [65, 27]. Triple classification's goal is to gather the correctness of unseen triples. This means it checks if the relation “capitalOf“ between the subject “Vienna“ and object “Austria“ is correct [4]. Triple classification or fact prediction checks if a triple is true and the next task classifies missing objects [63].

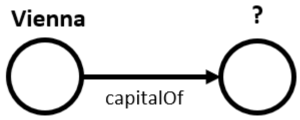

Figure 7: Entity classification predicts missing semantics for entities. For example, the results classify Vienna as city.

Entity classification The classification of entities puts entities into semantic classes, for example a person [65, 78]. Entity classification predicts matching labels of entities (Figure 7) [17, 8, 70, 43]. The in-KG task entity resolution checks if two different entities mean the same.

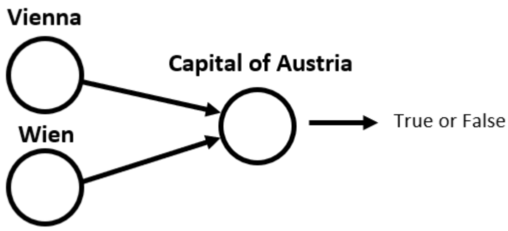

Figure 8: Entity resolution removes duplicate nodes. This cleaning task deletes entities with the same reference. In our example, the KG has the entities “Vienna“ and “Wien“, which refer both to the same city.

Entity resolution Entity resolution is a verifying process for two entities. It checks if nodes are referring to the same object or not (Figure 8) [65, 55, 23, 40]. This resolves the duplicate problem of nodes [65]. The last task of in-KGs is entity prediction.

Entity prediction The goal of link predictions is to find the missing relations between entities in the knowledge graphs. Entity prediction aims to locate missing relations. Entity prediction searches for incomplete triples. Triples only exist with one relation and one entity. Similar to link prediction, the missing entity will be found with the support of a scoring system. This scoring method scores the most probable missing entities [29]. The KGC uses these in-KG tasks. We split the KGC process into two variations. We describe one of them in the next part of the thesis.

2.4 Traditional knowledge graph completion

The traditional knowledge graph completion methods or none embedding methods are based on rule reasoning, probabilistic graph model, and graph calculation [12]. Graph calculation methods are mostly multi-hop whereas KGE are single-hop [33, 54, 35]. We present each of them in turn, with rule reasoning as the first method.

Rule reasoning With this method, new knowledge within a KG derives through statistical feature or rule reasoning (RR) and enables the KGC process. RR extracts and establishes these rules automatically through the use of semantics, or includes rules manually [10]. Rule-based reasoning extracts new knowledge through rules. The completion process uses this knowledge. Through semantics and deduction. RR can categorize nodes through this set of rules. It reason through rules to deduce new triples. Through these rules, RR can deduce that the entity Vienna and the entity Linz are the same category. The accuracy of RR depends on accurate rules and comprehensiveness. An advantage is that these rules are understandable and logical for humans and can be gathered automatically [41]. To generate these rules, the RR methods often use Markov logic network [20]. But Ii has many deficiencies like a need to complete rules to function, but these rules are hard to gain. The accuracy and completion in practice is not efficient and needs high computation power [79]. Chen et al., 2020 write that as KGs keep increasing in size the NKGE methods are no longer serviceable. RR methods are DeepPath, OWL2RL, and KGRL [12]. The probabilistic graph model, the second traditional KGC method, solves some shortcomings of rule reasoning.

Rule Reasoning Traditional knowledge graph completion
OWL2RL [12]
KGRL [12]
ProPPR [69]
DeepPath [57]

Table 1: This is a list of rule reasoning completion methods. They are none embedding and solve the completion process with a set of rules.

Probabilistic graph model The probabilistic graph model (PGM) utilizes graphs that show probabilistic relations. Through this relation, it uses probability modal to deal with uncertainty. The advantages are flexible topological structure, understandable, and clear semantics. It allows relational semantic interpretability. The Markov logic network and Bayesian network are mostly used for KGC within a PGM [64]. This model needs lower computational power than graph reasoning, but has a complex algorithm and is difficult to do with large-scale multi-relation KGs [12]. The last NKGE method we present is graph calculation.

Graph calculation Graph calculation (GC) or path ranking algorithm calculates with nodes and edges and uses the graph structure of KGs. GC utilizes the statistical characteristics of edges and nodes, outgoing degree and incoming degree of nodes, and the adjacent matrix to predict new relations and entities [64]. The literature calls this paths [88]. These paths rank the distance of the nearest path to for IN-KG tasks [69]. The learning phase includes feature extraction, feature calculation, and construction classifier [64]. Random walk is one method for feature extraction [64]. Methods of GC for KGC are Path Ranking Algorithm (PRA), Coupled Path Ranking Algorithm (CPRA), and several others with PRA as first method from 2013 [18, 60]. Reinforcement learning improves upon the random walk method and the including path solving or reasoning problem with Markov decision process (MDP) [32, 33, 60]. MDP uses a reward function to calculate the path and is sensitive to these rewards [31]. Methods that utilize this are DeepPath, MINVERVA and M-WALK [64, 33, 36]. Based on the literature, KGE improves upon the NKGE methods [12]. As path GC use random walk, the methods deal with noise, randomness and uncertainy [60]. Another problem is the inefficiency of computation as most GC methods needs to path through the whole graph [60]. In the next subsection we describe the different KGE methods, categorize them and list their advantages and disadvantages.

Graph Calculation Traditional knowledge graph completion
PRA [60]
MINVERVA [54]
M-Walk [32]
ADRL [64]
Neural LP [41]
DRUM [10]
MultiHopKG [36]
NTP-λ [54]

Table 2: Graph calculation or path ranking methods calculate the distance of paths to support the completion of a knowledge graph.

2.5 Knowledge Graph Embedding

The main aim of KGE is to hold the original graph structure in-tact while mapping a graph into low-dimensional vectors [19, 21, 73, 11]. This now embedded triple can also include the semantics from its original form [80]. Representation learning is the other definition that the literature uses for this category [55, 24]. The main focus of KGE in the literature is about triples and their textual information [17]. This solution of embedding triples for current knowledge graph problems enables a more efficient for storing and computation of data [4, 48, 16, 80]. This means that KGE utilizes vectors for the embedding to gain more efficiency with the computing processes [76]. Graph embedding, originally designed in the early 2000 to reduce the dimensionality of non-relational data [4]. Graph embedding sets individual nodes into low-dimensional space and compares the closeness of each node [4, 76]. Different embedding methods use different inputs besides triples.

There are different inputs and outputs for each graph embedding method. One input can be a triple from a KG, and an output of the embedded triple into a vector space. As literature states, node embedding as an output setting is the most researched. The output produces a representation of the input graph in the form of a low dimensional vector. Depending on the output granularity, node embedding, edge embedding, hybrid embedding, and whole-graph embedding are different graph embedding outputs [4]. Node embedding embeds similar nodes like Vienna and Linz, whereas whole-graph embedding the whole graph [4].

Knowledge graph embedding models need training before they can be used. There are two existing assumptions for the training. Either the open world assumption or the closed world assumption. The open world assumption states that there are only true facts within a knowledge graph and not observed facts are either missing or false [24]. The closed world assumption states that all not included facts are false [65]. We and many articles split the KGE methods into three main categories [36, 80, 14]. These are translation, semantic matching, and network representation learning [83, 18, 21]. The first model we cover is the translation.

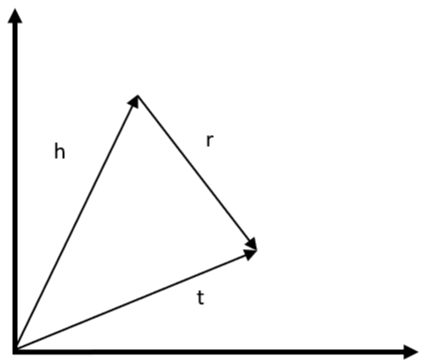

Figure 9: TransE embeds a triple into a low dimensional vector space. "h" and "t" are the entities and r is the relation of triple. It uses the distance between h and t for calculations to achieve in-KG tasks.

Translation model Translation based methods focus on semantic information of triples [17]. The most popular methods are TransE, TransH, TransR, and TransD [15, 56, 58]. The first and original translation based method is TransE [14]. The Figure 9 visualizes the embedding of a triple through the TransE method [65, 58]. TransE embeds relations as distance between head entity and tail entity and performs the best with 1-to-1 relations but struggles with more complex ones [9, 83, 26, 71]. Other translation models like TransH try to overcome this weakness and gain better results with 1-to-N, N-to-1, or N-to-N relations [71, 51, 56, 62]. TransH embeds relations through hyperplane and TransR matrix embedding [83, 26, 71]. The Table 3 contains all translation based methods of our literature that we include in the bachelor thesis.

Translation-based Knowledge graph embedding
TransE [65]
TransH [44]
TransR [9]
RotatE [82]
SE [19]
TorusE [15]
HAKE [80]
DualE [8]

Table 3: The translation-based embedding methods utilize the distance between embedded entities for the scoring of plausibility. This table present a list of common translation based methods.

Translation models focus on the search of valid triples based on a score function and try to minimize the loss function [59]. It scores the plausibility by calculating the distance between the embedded entities [65, 27]. Translation-based models mostly use the distance of the embedded triples to either score or compare similarity to check the plausibility of triples [27]. It utilizes the relations between entities and the structure of the KG to predict unseen facts. Translation models need a complex training for the algorithm and are more difficult than NKGE methods, but allows for simple, understandable results [12, 56]. To improve upon the strength of this method, semantic matching models include the semantic aspect into the embedding process.

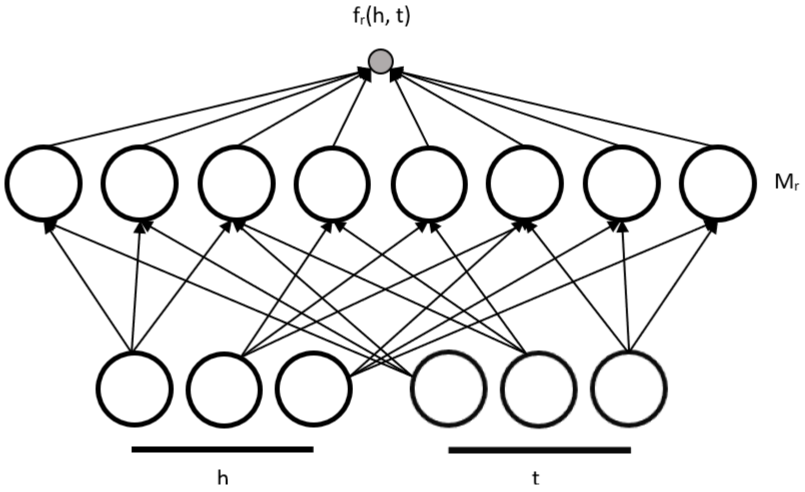

Figure 10: RESCAL is a semantic matching model that compares head and tail entities semantics to calculate a score and to process in-KG tasks. The final output f_r(h, t) present the score of a triple.

Semantic matching model The semantic matching model or bilinear models uses the semantic factor to compare similar entities and relations to predict new facts [12, 15, 27, 51]. These methods include the semantics on their score functions [80]. As Figure 10 shows the interaction of head entities and tail entities through RESCAL. RESCAL compares semantic pairs and semantics of entities through a scoring system and outputs a score of fact f_r(h, t) [12, 65]. RESCAL uses relational-type constraints to improve in-KG tasks like link prediction. Relational type-constraints describe the logic of relations by removing either the head or tail entity [83]. Within the Table 4 is the collection of semantic matching models.

Semantic Matching Knowledge graph embedding
RESCAL [66]
DistMult [1]
HolE [83]
ANALOGY [61]
SME [22]
BILINEAR [47]
TuckER [3]

Table 4: This list of semantic matching methods compare embedded triples semantics to calculate a score. The scoring supports the knowledge graph completion.

One problem of these type of model is the none efficient scaling with bigger KGs [14]. Methods for KGC within semantic matching model are SME, DistMult, HolE, RESCAL, and ANALOGY [22, 47, 48, 68]. The last KGC method is network representation learning.

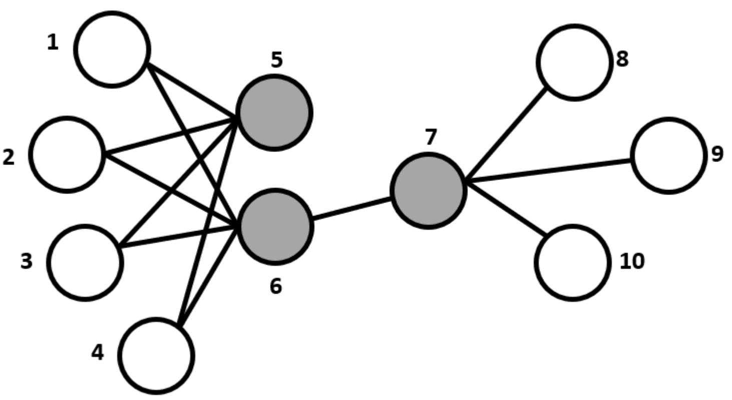

Figure 11: Network structure diagram embeds triples into tensor instead of vectors to perform a scoring function. The structure is comparable to a neural network.

Network representation learning Network representation learning (NRL) uses information of a KG to gain a high accuracy in predicting missing information for the KGC process [12]. The structure of layers is similar to a neural network [15, 14]. This information includes the characteristics of graph network structure, nodes, and relations. The Figure 11 visualizes this a structure. This category also includes scoring methods and functions [59]. The models embed the entities and the relations to tensors as input [59]. In comparison to the previous categories, the network representation models scale better as they use multiple layers for their calculations [59]. Another advantage is the high number of possible parameters that improve the expressiveness of the models [14]. In the recent literature, the focus increases for these models [80]. However, it struggles with too many different relation and overloads the data in training [14]. Models for network representation learning are DeepWalk, Node2vec, GCN and Line [12, 6, 8]. We put all baseline network representation learning models within Table 5. The next section summarizes our related work.

Network Representation Learning Knowledge graph embedding
DeepWalk [5]
LINE [12]
Node2vec [6]
ConvE [11]
GCN [48]
ConvKB [59]

Table 5: Network representation learning methods embed into tensors and are similar to neural networks.

3 Related work

The related work section for this systematic literature review refers to papers that focus on reviewing different aspects of in-KG tasks and knowledge completion methods.

Knowledge graph completion: A review Although there is extensive work on knowledge graph completion in the literature, there are few papers that compare these different methods and list their strength and weaknesses. This paper focuses on existing methods and divides them into traditional knowledge graph completion and deep learning-based knowledge graph completion [12]. Our approach is similar, as we divide the methods into KGE and NKGE. However, the paper is not comparing the performance between these methods. This means our papers list the KGC methods in an identical way, but we explore further and compare the historic performance difference between KGE and NKGE.

A survey on knowledge graphs: representation, acquisition, and applications The paper fixates on four main points. These are KGE, KGC, KG learning, and use cases. The goal is to summarize the current methodologies and applications of KGs [27]. Like in our thesis as we review the current methodologies of in-KG tasks and KGC methods.

A comprehensive survey of graph embedding: problems, techniques, and applications This literature review of graph embedding contains the basic concept of graph embedding, explains the subsequent problems and categorizes each embedding method. Similar to our work, there is also a comparison between the graph embedding methods and a list of advantages and disadvantages [4]. But, we also include the NKGE methods and the in-KG tasks and compare them through their individual historic performances.

ADRL: attention-based deep reinforcement learning framework for knowledge graph reasoning The scientific article compares their KGC method, a NKGE method, to other methods including both of KGE and NKGE. It is the only article from our initial inclusion without the expansion strategy that includes the traditional knowledge graph completion methods. The IN-KG tasks for their tests include LP and TC, the criterion of hits@10 [63].

Bringin light into the dark: A large-scale evaluation of knowledge graph embedding models under a unified framework The paper takes a similar approach like our work. But instead of comparing the NGKE methods to KGE it focuses on the different categories of only KGE. The article gives deep insight into each of the different variants and allows for a structured and understandable performance comparison [1].

4 Research Methodology

We use a systematic literature review in this thesis to solve the research questions. A set of limitations and a scope enables the best possible research process for answering the scientific questions. The scope and limitations reduce the selection of literature to 87 different scientific papers to represent the current status of knowledge graphs and knowledge graph embedding. We include 46 of these 87 papers for the performance evaluation.

4.1 Search strategy

In this subsection, we represent our search strategy for our systematic literature review. This includes the construction of our search string, our search strategy and the expansion of our search strategy.

	Search terms
Method	(“knowledge graph“) **OR** (“embedding“) **OR** **AND**
In-KG tasks	((“link prediction“) **OR** (“triple classification“) **OR** (“entity classification“) **OR** (“entity resolution“)) **OR**
Refinement	(“completion“)

Table 6: We build a search string through the combination of two search terms. Like a method with knowledge graph and refinement with completion to create the search string knowledge graph completion. The search string supports our research and improves the probability of finding matching literature.

Table 6 shows the search terms and search strings for our research process. The keywords are knowledge graph, link prediction, triple classification, entity classification, entity resolution, embedding, and completion. These are the initial search terms for our systematic literature review to find the best matching literature to answer our research questions. We combine the search terms of either one of method with one search term of in-KG tasks or refinement to build the best possible search string. The structure of one possible search string of method + In-KG tasks/Refinement looks like knowledge graph (method) completion (refinement). The next paragraph explains the structure of our search strategy.

Target databases	Scopus Google Scholar
Conferences	AAAI Conference on Artificial Intelligence (AAAI) and Knowledge Management (CIKM) The Web Conference (WWW)
Journals	Knowledge-Based Systems Journal of Web Semantics IEEE Access
Expansion strategy Conference	 Conference on Empirical Methods in Natural Language Processing (EMNLP) International Joint Conference on Artificial Intelligence (IJCAI) Conference on Neural Information Processing Systems (NeurIPS)

Table 7: Summary of the search string defines our selection of databases and conferences or journals. Literature we use in this bachelor thesis derives from these databases, conferences, and journals.

The summary of the search strategy is in Table 7. The search strategy includes the main databases for the search process, the target conferences and the target journals. The two databases are Scopus.com and Google Scholar. The conferences and journals with the most articles after the search process are the targets in our strategy. The target conferences are AAAI Conference on Artificial Intelligence (AAAI), International Conference on Information and Knowledge Management (CIKM), and The Web Conference (WWW). And the 3 target journals of this systematic literature review are Knowledge-Based Systems, IEEE Access and Journal of Web Semantics. 390 articles from the search process are from the Google Scholar database. The additional 26 articles come from Scopus. This leads to a total amount of

416 articles collected with the support of our search strategies in our initial search process. To accommodate a lack of data for traditional knowledge graph methods we initiate an expansion search.

Expansion search strategy The amount of data for a performance comparison over time between KGE and NGKE method through our initial search strategy is not enough. Therefore, we use an expansion to our search strategy and include an additional 51 papers to enable a performance comparison. For this expansion strategy, we use the Google Scholar database. We combine our search string (knowledge graph & completion) with all NKGE methods from our research. An example for a search string from our expansion search is (knowledge graph completion PRA). We add the conferences with Conference on Empirical Methods in Natural Language Processing (EMNLP), International Joint Conference on Artificial Intelligence (IJCAI), and Conference on Neural Information Processing Systems (NeurIPS) as these are the conference/journal with the most papers within our collection of literature. The new total amount of papers after the second search process is 440 the number of conferences and journals is 9. In the study selection, we explain our inclusion and exclusion criteria.

4.2 Study selection

The study selection includes the criteria of inclusion and exclusion, the explanation for our preliminary and post selection approach and the expansion to our study selection.

Inclusion criteria	IC-1: Terms fulfill the search string IC-2: Academic journal and conference papers IC-3: Papers written in English IC-4: Amount of citations at least 30 IC-5: Paper part of the selected journal or conference
Exclusion criteria for titles and abstract for full text	EC-1: Paper in PowerPoint format EC-2: Paper not focusing on knowledge graphs EC-3: Paper contents not relevant

Table 8: Summary of the selection strategy presents our preliminary criteria for our inclusion of literature. We list 5 inclusion and 3 exclusion criteria.

In the research phase we limit the scope and the number of literature as our selection strategy as shown in Table 8. The selection method has inclusion criteria and exclusion criteria. The preliminary phase of inclusion and exclusion support initial selection of papers before the full-text assessment. The preliminary inclusion criteria consist of matching the keywords or search string as inclusion criterion 1. The articles have to be part of an academic journal or needs to be a conference paper in the second inclusion criterion. The paper needs to have English as publishing language as the third criterion. The fourth criterion is that the minimum amount of citations for the selection of literature is 30 or more. The last inclusion criterion is that the paper is part of the 3 journals or 3 conferences mentioned above. The exclusion criteria split into two parts. The first part is for titles and abstract and the second part is after the full text research. The first exclusion is for papers with the structure of a PowerPoint presentation. The second for titles and the abstract, not focusing on the research topic of knowledge graphs. The full text exclusion criterion EC-3 is about the contents of the paper are not relevant to answer the research question.

The scope limits the literature by the usage of keywords and papers published by conference and journal papers. Another limitation is the number of citations of a scientific paper. The selection excludes paper with below 30 citations. The literature for this systematic literature review derives from the databases Scopus and Google Scholar. The reviewed scientific papers categorize into pre-included papers and pre-excluded papers. This strict limitation of the scope leads to the inclusion of 88 papers from the initial total 416 papers. The reduction of papers brings the amount of articles to a remaining 21.1 % of scientific research paper as main resource of knowledge for the construction of this paper. Within the full-text post exclusion criteria, the assessment of 88 papers leads to the exclusion of 26 paper and a total of 62 for the qualitative synthesis. The post excluded papers are either not focusing on in-KG tasks or the contents are not matching the purpose of this research paper. The next paragraph explains the expansion for our selection strategy.

Inclusion criteria	IC-1: Terms fulfill the expansion search string
	IC-2: Academic journal and conference papers
	IC-3: Papers written in English
	IC-4: Amount of citations at least 20
	IC-5: Paper part of the target journals or conference
Exclusion criteria	EC-1: Paper in PowerPoint format
	EC-2: Paper not focusing on knowledge graphs
	EC-3: Paper not focusing on in-KG tasks
	EC-4: Paper not including NKGE methods
	EC-5: Paper not including performance data of NKGE methods

Table 9: With the summary of the expansion selection strategy, we define criteria for the additional literature.

Expansion strategy in the selection phase The additional papers need to meet our requirements. We summarize these in Table 9. From the selection of 51 papers, we pre-select 28 through our scope of target conferences or journals. We include the next biggest conference or journal as a target, which is EMNLP, IJCAI, and NeurIPS. These 28 articles are either part of an already selected journal/conference or are from the EMNLP, IJCAI, and NeurIPS conferences. 3 are from the journal IEEE Access, 1 is from Knowledge-Based Systems, and 3 papers are from the conference AAAI. The remaining 21 are from the conferences of EMNLP with 14, 5 are from IJCAI, and the last 2 are from NeurIPS. 15 of these 21 additional papers are from our initial search. The other 7 derive from the expansion search process. The post exclusion removes 3 articles from the additional expansion selection. These 3 do not meet our post exclusion criteria of Table 9. They do not include matching content or performance data of in-KG tasks. The remaining 25 papers are part of the post inclusion. This leads to a total amount of 87 articles for this bachelor thesis and 46 from the 87 papers for the performance evaluation. And a total amount of 353 papers in the exclusion.

4.3 Data extraction

We collect performance data of In-knowledge graph tasks from our selection of papers and transfer them into tables to visualize and analyse our collection of data. The performance tables structure includes the digital object identifier (DOI) of each articles as a primary key. The x-axis is the publishing date of each data set and enables sorting through dates to research the performance difference over time. The y-axis represents the performance of each method. Every data point derives of either a KGE or NKGE. We guarantee this affiliation with a boolean. We specify if the data in the literature utilizes variance in the data sets to present the results. Furthermore, we add additional information for each entry that includes the specific in-KG task, the KGC method and the data-sets.

We enable a comparison between each task and KGE against NKGE with the visualization of performance data. This allows for a structured visual data table. This table shows the extent of difference between KGE and NKGE. The data-sets or databases for the comparison of data between literature are from WordNet or from Freebase [13, 26]. The limited subsets of these two are WN11, WN18 and WN18RR. Freebase has FB13, FB15k and FB15k-237 as subsets. The remaining databases the literature mainly uses are YAGO3, YAGO37, Wiki13k, NELL-995, DBP15k, Wikidata, WK31-15k and WK31-120k [13, 71, 56, 21]. For the comparison of our data we only use the subsets of WN18, Wn18RR, FB15K, and FB15K-237 due to not enough data from the literature of other data sets. As criterion for the performance, we select the hits@10 and Mean Rank for link prediction and MAP for triple classification. For the data collection, we insert the data into an Excel table. We then import the Excel table into python. We analyse the data and create plots through pandas. In the next paragraph, we discuss the challenges and limitations of these processes.

Challenges and limitations The biggest challenge we have to overcome is the scarcity of performance data for NKGE methods. As we mention above, the amount of papers that include NKGE methods into their performance comparison of in-KG tasks in our initial search process is 1. This leads to the necessity of the expansion strategy to compensate the lower amount of NKGE data and to guarantee a fair comparison between those two categories. However, even through the expansion strategy, the amount of data entries of

KGE to NKGE is still 8/1. This means that we have to limit the comparison of in-KG tasks to only LP and TC. The biggest limitation of this bachelor thesis is the exclusion of NKGE methods in KGE papers. And the hardest challenge to overcame is to enable fair and valid comparison of both categories with the unequal amount of data available. Within the results section, we present our findings and the performance of both KGC main categories.

4.4 Outline and scope

The first chapter starts with the introduction. It explains the problem and our hypothesis. In the subsections 1.1 and 1.2 we introduce the research questions and the research method. In Chapter 2 we focus on the background. The section contains information about KGs, KGC, In-KG tasks, NKGE and KGE. Section 3 is about related work and papers that present a similar approach to our work. In chapter 4, the research methodology, we list the steps we take to get to our results with the subsections search strategy, study selection, data extraction, and outline and scope. Within section 5 we present the results and findings of the systematic literature review. The results section includes the study characteristics and the performance comparison. Chapter 6 contains the discussion section. We include the scientific research trends for KGC, our interpretation of findings, address the limitations and strengths of our research method, answer our research questions and discuss future research. The next section is the conclusion. We summarize our relevant and significant findings. The reference chapter is the last section and lists the cited materials in the thesis.

We limit the thesis scope in our initial research phase by considering only 3 conferences and 3 journals (Figure 12). The 3 selected conferences are AAAI, International Conference on Information and Knowledge Management, and The Web Conference. The 3 journals to scope the selection of articles are Knowledge-Based Systems and Journal of Web Semantics. Through the expansion strategy we include the conferences EMNLP, IJCAI, and NeurIPS . The next section presents our results of our performance evaluation.

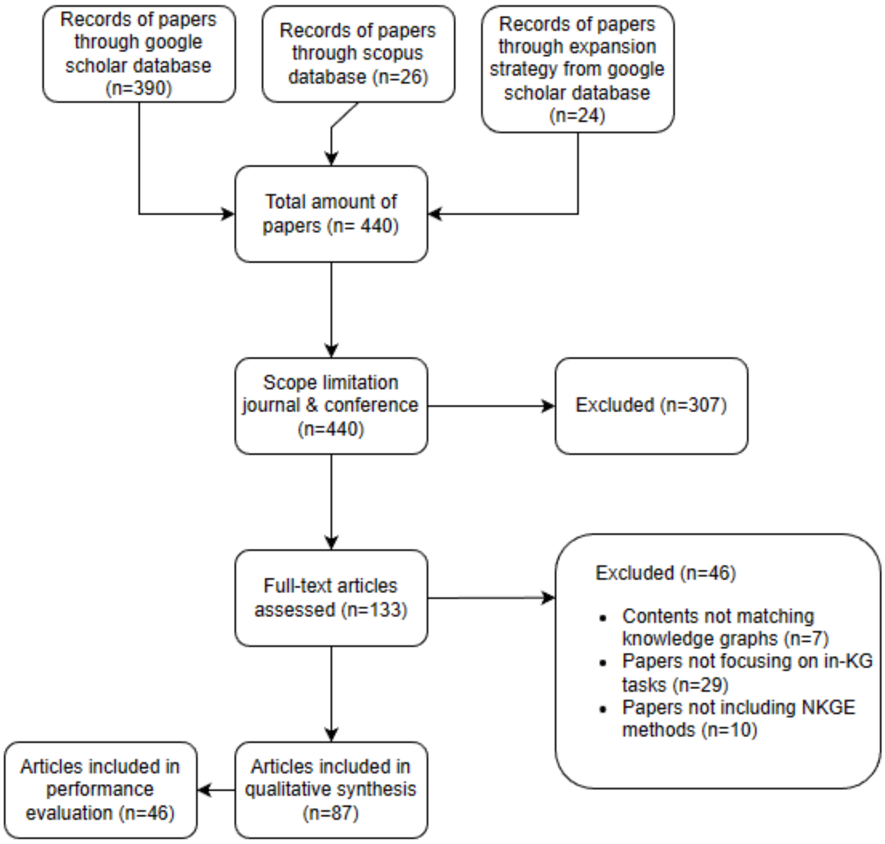

Figure 12: The flow diagram presents our processes and the amount of literature we collect for each step within our literature research. Our initial search includes Google Scholar with 390 papers and Scopus with 26. The expansion strategy adds 24 articles from Google Scholar. From 440 papers, 87 are in our qualitative synthesis. From the remaining papers, we use 46 for the performance evaluation. The remaining 41 articles within our qualitative synthesis selection do not possess the correct selection of data for the performance comparison.

5 Results

In the results section, we present our findings of the performance comparison of the knowledge graph completion categories. We collect the data of papers that match all our inclusion criteria. These papers are from the journals of Knowledge-Based Systems, IEEE access, and the conferences of AAAI, International Conference on Information and Knowledge Management, EMNLP, IJCAI, and NeurIPS. The collection of papers from the The Web Conference and the Journal of Web Semantics do not possess performance data for our performance evaluation. This section includes the study characteristics, findings, and our scientific questions. First, we explain the study characteristics of the papers in the next paragraph.

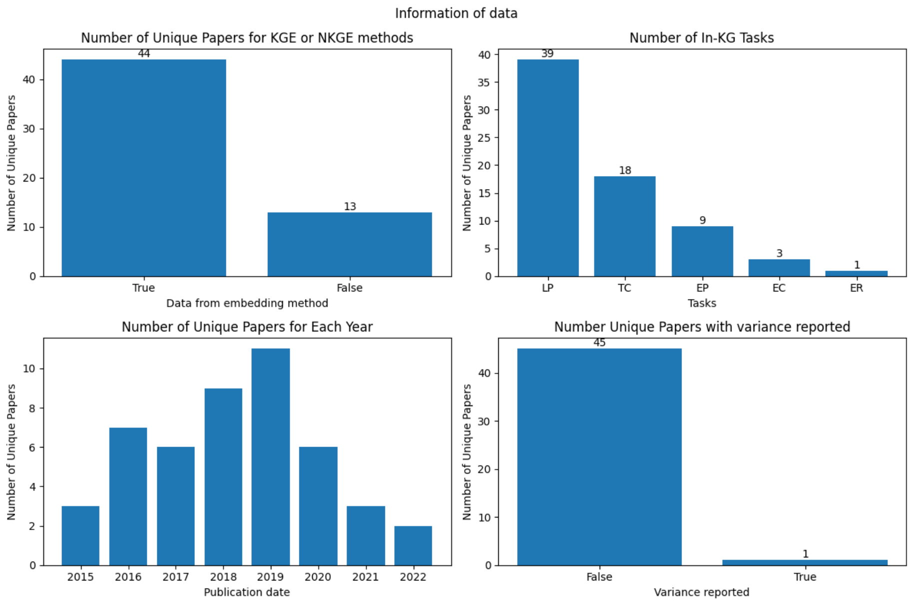

Figure 13: The statistics summary of our paper selection includes four figures. The first one shows the amount of papers with NKGE methods data. The second, the distribution of in-KG tasks. The third the number of paper publications for each year. And the last figure, the amount of papers that present their performance data with variance.

5.1 Study characteristics

In the study characteristics, we discuss relevant information about paper details for the data extraction. The total amount of papers we use to compare the performance is 46. As visible in Table 10, 18 of these papers are from the conference AAAI, 9 from EMNLP, 1 from International Conference on Information and Knowledge Management, 5 articles are from IJCAI, and 2 are from NeurIPS. The remaining papers are from the journals IEEE Access and Knowledge-Based Systems, with 7 and 4 papers. This means we do not have the necessary performance data for the performance evaluation from papers of The Web Conference and Journal of Web Semantics. The date of publication for these scientific articles range from 2015 to 2022. With the earliest release in 1.1.2015 and the latest at 5.5.2022. The total amount of unique data entries is 978. The first bar chart, "Number of KGE or NKGE methods" in Figure 13 visualizes the relation of entries between KGE and NKGE. The bar represents True as a KGE with 44 papers and False as a NKGE with 13. This means that some papers of NKGE include performance data for KGE methods. The second bar chart shows the usage of each in-KG task. The most prominent task is link prediction with 39 papers and triple classification as the second biggest with 18. The last three are entity prediction, entity classification, and entity resolution with 9, 3, and 1 papers. The first 2 bar charts can include papers of both KGE and NKGE methods and more than one In-KG-task within their performance data. The third bar chart in Figure 13 includes the total amount of paper published in each year. The year 2019 contains the most articles, with 11. The least amount of publications is in 2022 with 2. The last chart visualizes all articles that include a variance within their performance results. One entry within our 46 publications uses variance. We continue with the explanation of our paper attributes.

Conference	Number of papers
AAAI	18
EMNLP	9
International Conference on Information and Knowledge Management	1
IJCAI	5
NeurIPS	2
Journal	Number of papers
IEEE	7
Knowledge-Based Systems	4

Table 10: Number of papers for each conference and journal, which are included in the performance evaluation.

In Table 11 we include structural information of our performance graphs, plots and bar charts. The performance data we collect and use for our comparison are all from filter instead of raw. The filter option removes corrupted triples and delivers better performance than the raw version. Consequently, the literature prefers to use the filter variant for performance testing.

X-axis	Publication date
Y-axis	Performance data from papers
Primary key	Digital Object Identifier (DOI)
Column contents	
Embedding	true, false
Variance included	true, false
In-KG tasks	Link prediction (LP), Triple classification (TC), Entity prediction (EP), Entity classification (EC), Entity resolution (ER)
Methods	TransE, TransR, TransH, DKRL, SSP, PRA, DistMult, ConvE, MINERVA, M-Walk, ADRL, RESCAL, SE, SME, LFM, TransD, TranSparse, ConvKB, R-GCN, BILINEAR, ComplEX, ANALOGY, TorusE, RotatE, DCN, M-DCN, HolE, Unstructured, SACN, QuatE, SimplE, DualE, LMF, Random Walk, Goal-Directed Random Walk, TuckER, NeuralLP, NTP, DRUM, Multihop-KG, DeepPath, SLM, NTN, CKRL
Data-sets	WN18, WN18RR, FB15k, FB15k-237,
Evaluation protocols	hits@10, MAP

Table 11: Paper attributes presents the structure of an entry within our data table. Each entry has a DOI as primary key, a publication date for the x-axis, the performance for the y-axis. The column additionally includes if the data uses embedding or variance, which in-KG tasks it performs, KGC method, and on which data-set it tests.

We insert the data collection into a table with the columns DOI, publication date, embedding, variance reported, In-KG-Tasks, Method, Category, Top-k Criterion, Data-Sets, and Results. We input each performance result from an article into a row. This can lead to more entries from one paper. The publication date represents the latest version of papers. We use it as x-axis to enable a performance comparison over time. The DOI describes the digital object identifier and primary key for the results of each scientific paper. The embedding column represents a boolean that indicates the usage of a KGE method for this result. With the "variance included" column, we determine the amount of data that represent their results with a variance through true or false. The variance shows the spread of the performance score. An example would be the LP score of 98.2 with the variance of σ^2. This means scores spread around 2 point higher or lower from 98.2. In-KG tasks defines a task, like LP. Methods represent a specific KGE or NKGE method. The "Results" field contains the performance data. We describe the input for the data-sets column in the next paragraph.

Data-sets To enable a performance comparison between NKGE and KGE methods, we collect data from each paper through the qualitative synthesis. It is only possible to compare the performance for each method from the same data-set. This leads to the limitation of the data-sets most common in literature. Our performance comparison contains the 4 data-sets WN18, WN18RR, FB15K and FB15K-236. These data-sets are either subsets of WordNet or Freebase. Our first data set we describe is the WordNet family. WordNet has the two data sets WN18 and WN18RR [1]. In WordNet the relation is conceptual-semantic [1, 39]. The structure is build through a hierarchy with relations build as lexical relationships [3, 73, 22]. The WN18 data-set around 40,900 terms and 18 relationships [1]. WN18RR is another subset of WordNet. WN18RR is similar to WN18. It is a subset of WordNet and has the same amount of terms as WN18. The difference is the removal of inverse relations [3]. This subset has 11 relationships [1]. The first freebase data-set is FB15K. FB15K contains real world facts like about movies, awards, actors, and sports [3]. The amount of entities is around 15,000 with 1,345 relationships between them [1]. The last data-set we include is FB15K-237. This subset is comparable to the WN18RR variant, as the relations are inverse [1, 3]. This change leads to worse performance for simple models [3]. The next important topic we discuss is the evaluation protocol of our collection of performance data.

5.2 Performance comparison

We discuss the performance comparison of in-KG tasks between KGE and NKGE methods over time. The two tasks we present are link prediction and triple classification. For link prediction, we have enough data to visualize a graph for all four data-sets with WN18, WN18RR, FB15K, and FB15K-237. Triple classification has one plot with the data-set FB15K-237. The x-axis represents the years, and the y-axis contains information about the performance on the criterion and the performance results. The legend for the performance comparison graph includes "Embedding" and "Standard Deviation". The blue lines represent the performance for KGE and are "Embedding" and the light blue area visualizes the "Standard Deviation". The orange line stands for the NKGE data performance and the light orange area for the "Standard Deviation" of traditional methods. We start with the link prediction on the data set WN18 in the next paragraph.

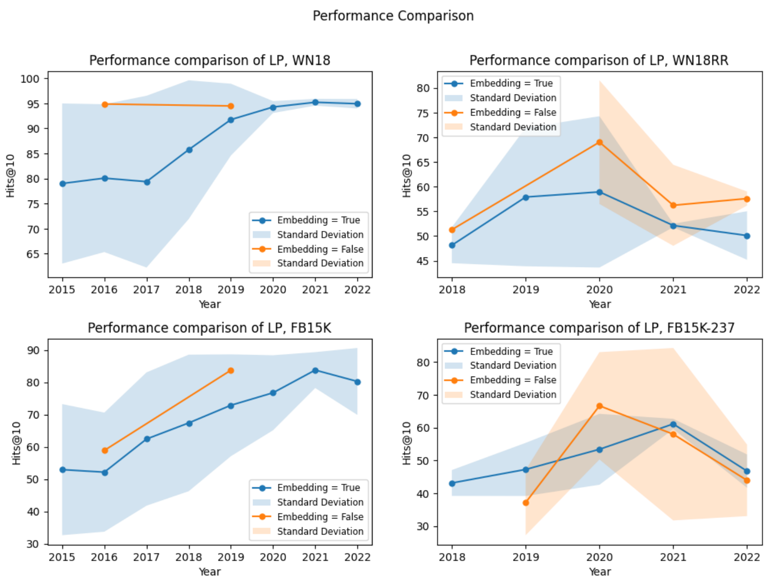

Figure 14: The performance comparison of the evaluation data is from the link prediction task and the criterion hits@10. These four figures visualize the performance within the data-sets WN18, WN18RR, FB15K, and FB15K-237.

Link prediction on WN18 The upper left graph in Figure 14 contains the link prediction performance of KGE and NKGE methods on the HITS@10 criterion. The comparison over time ranges from years 2015 to 2022. For the NKGE methods within link prediction on WN18, we possess not enough data to enable a render for the light orange standard deviation area. The performance of the KGE in 2015 shows a result of about 79. In the 2016 this performance rises to around 80 with a following slightly lower performance of 79 in 2017. The performance increases in 2018 and 2019 to 86 and 92. From 2020 to 2021 the performance increases to 94 and 95 and drops back in 2022 to 94. The lowest point for KGE is in 2015 and 2017 and the highest in 2021. The standard deviation for KGE between 2015 and 2019 is massive. Whereas in 2019 to 2022 it is narrow. NKGE has 2 data points, because of the lack of performance data for these methods within the WN18 data-set. The performance for NKGE is slightly decreasing between 2016 and 2019 from around 95 to 94. In 2016 NKGE methods are about 18.8% better in performance than KGE. In 2019 NKGE has around 2.2% better results. A comparison between the standard deviation is not possible due to the lack of data. In the performance graph for link prediction with the data set WN18, the NKGE methods possess better results in comparable years. With the next graph, we present the link prediction results on WN18RR.

Link prediction on WN18RR The graph for WN18RR on the in-KG task LP is the upper right on Figure 14. The data ranges from the years 2018 to 2022. The standard deviation for NKGE starts at 2020 as not enough data exists. The KGE results in 2018 are at 47 and increase around 21.3% in 2019 to 57. In the next year is a slowly increases in performance to 58. Then we register decreases to 53 in 2021 and 50 in 2022. NKGE starts with a better performance than the other category in 2018 at 51. In the year 2019 there is no data point. Then it increases to 69 in 2020 to its highest point. The next performance result for NKGE in 2021 lowers drastically to 56. In the last year we register another slow increase to 57. The results for both categories in link prediction on WN18RR shows that NKGE performs better in every year with an existing data point. At its highest point in 2020 the performance difference is 18.9% in favour of the traditional methods. Both methods have their lowest point in 2018 with 8.5% better results for NKGE. The trajectory of both lines within WN18RR are comparably similar. The KGE standard deviation is besides 2018 and 2022 very spread out and in 2021 there is no standard deviation. The blue area is drastically spread out in the years 2019 and 2020. The NKGE standard deviation measures only in 2020 to 2022. The data is similarly spread out to the blue variant for the years 2020 and 2021. But in 2022 the standard deviation is small. Overall the results show better performance for the NKGE methods with the lowest performance advantage in 2018 with 8.5% and the highest in 2020 with 18.9%. In the LP graph with the data-set WN18RR, NKGE perform better for each year. In the upcoming paragraph, we discuss the performance difference over time for LP on the FB15K data-set.

Link prediction on FB15K The graph for LP on FB15K is in the lower left corner from the Figure 14. The data comes from the years between 2015 and 2022. Similar to the previous two graphs, the lack of data leads to no visible standard deviation for the NKGE methods. Additionally, NKGE has only two data points. KGE start in 2015 on 52. It then decreases in the next year to 51. Between 2016 and 2021 there is a constant increase of performance that reaches 83. For 2022 there is another small reduction that ends at 80. The standard deviation for KGE is not as spread out as in the previous graphs, but still high. In the year 2021 possesses the standard deviation, is the smallest value for KGE. Besides the reduction of performance from 2015 to 2016 and 2021 to 2022, the performance has a continuous upwards trend. NKGE has a comparable performance increase in the 2016 to 2019, to KGE methods with 59 and 82. For the performance comparison of LP in FB15K, NKGE methods present better results than its competitor. In 2016 the traditional variants have up to 15.7% better results and 13.9% in 2019. The highest point is NKGE with 82 and the lowest is from KGE with 51. We present the results of the last LP comparison next.

Link prediction on FB15K-237 The last graph is in the lower right side in Figure 14. It compares both methods in LP on the data-set FB15K-237. The comparison runs over the time period 2018 to 2022. Both methods have a standard deviation. However, there is no data of NKGE methods in 2018. Embedding performance starts in 2018 with a mean performance of around 42. The results increase continuously until they reach their highest point in 2021 with 61. For the last year, there is a drastic drop to 47. The NKGE line starts from the lowest point in the entire graph, with 36. Then it reaches the highest point of the graph in 2020 with a performance of 67. For the remainder of the traditional methods, the performance decreases. It ends with a result of 44 in 2022. Within the FB15K-237 LP graph, the KGE methods have the lowest standard deviation of the Figure 14. In comparison, NKGE standard deviation has a higher spread. The KGE have a better performance for every year besides 2020. In that year NKGE has its highest point with 67. And the lowest performance is in 2019 from NKGE data. The last comparison we have is for the triple classification task.

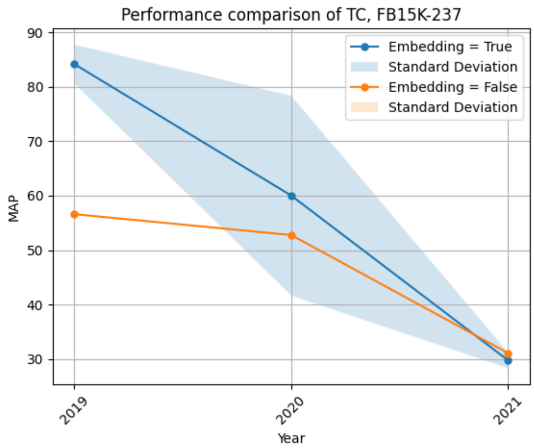

Figure 15: The performance comparison of the triple classification task is on the data-set FB15K-237 and the evaluation protocol MAP.

Triple classification on FB15K-237 The comparison over time for triple classification on the data-set FB15K-237 is visible in the Figure 15. The x-axis represents the publication years and the y-axis the criterion. The criterion for the in-KG task TC is MAP. The performance data ranges over a span of three years, from 2019 to 2021. For the NKGE category, there is no standard deviation present. The reason for this is the data sparsity in literature. KGE starts in 2019 with a performance of 84. It decreases in performance significantly in 2020 with 60 and in 2021 with 30. The standard deviation for KGE in TC is low for 2019 and 2021. But it is very high in 2020. The orange performance start at 57 in 2019. Sinks sightly in 2020 with 54 and reaches its worst results in 2021 with 32. The KGE performance through these years is better than the NKGE methods. KGE has both the best performing year and the weakest within our data evaluation in TC. The upcoming section discusses our performance results.

6 Discussion

In the discussion section, we present the most important findings of our research. This includes current trends within scientific articles of KGC, our interpretation of the results, strengths and limitations of our research method, and the answers to our research questions. We begin with the discussion of current trends.

Scientific research trends for KGC Within the current publications for KGC with the focus on KGE or NKGE there exist several observable trends within our data. We focus on the three most notable. We begin with the none existing comparison to NKGE methods within KGE papers, the complete exclusion of variance within performance results, and the lack of in-KG data besides LP and TC. First is the exclusion of NKGE within KGE articles. This means that in scientific papers with focus on KGE methods, they exclude NKGE methods from their test results. We must note that within our performance data papers that focus on KGE methods, do not include performance data of NKGE. In contrary to papers that concentrate on NKGE categories. These papers all include KGE results to compare their methods. The next trend within the current literature is the none existent usage of variance to represent their performance data. This is visible within the lower right plot of our Figure 13. A handful of papers include a reason for the exclusion [59]. Et al. Li is an exception. They list and discuss the variance within their results and are the sole inclusion within our data [31]. The last trend visible within our data is focus on LP and TC tasks. As the upper right figure in Figure 13 presents, those two in-KG tasks are 87.9% of our total data entries. Next we discuss our findings of the bachelor thesis.

Our interpretation of findings In this paragraph we present our main findings, we summarize our results, and we interpret the key findings. Our main results to answer our research questions include the number of paper that include either KGE or NKGE, the amount of in-KG tasks in the performance data, and the exclusion of reporting a variance. As previously mentioned, the relation of KGE to NKGE methods data within our selection of literature is a serious matter. We have 899 entries from 44 papers for the embedding categories and only 79 without. This is a critical difference, and it is noticeable that the NKGE methods are disregarded in the KGE literature. We must add that we needed to conduct an expansion strategy to be able to start a comparison. Another stark contrast in numbers is visible within the usage of in-KG tasks. The literature favours the LP task significantly more than others. It has more than twice as many entries in 39 papers within our data, with a 66.4% share in total. We must note that these numbers restrict the comparison of data and present a lack of research for the other in-KG tasks in recent literature. The last finding is the most surprising. Within our 978 entries from 46 papers, there is only one that includes the variance within their performance results representation. This is critical as it diminishes the credibility of the results. Next, we discuss our results of the performance comparison over time. In Figure 14, NKGE methods perform better in LP in all data-sets besides FB15K-237. In WN18, traditional KGE performs around 9.9% better. With around 12% the NKGE category has better results within WN18RR and 14.6% better performance for FB15K. Only in FB15K-237 the KGE has better results within the LP task with around 0.49%. In the TC task, KGE performance is 21.7% better than the KGE methods. With the next paragraph, we continue with the strengths and limitations of our research method.

Strengths and limitations of our research method Our systematic literature review process with PRISMA as guideline is a reliable and robust process. In the research process, the biggest problem we encountered, was the diminishing low amount of NKGE methods performance data within our initial selection. As solution we conducted an expansion strategy that allowed us to conduct a performance comparison. We speculate that the current extensive focus on KGE methods in the literature is the reason for the low amount of NKGE data. Another limitation is the usage of only two databases, with Google Scholar and Scopus.com. In the last paragraph of our discussion section, we answer our research questions.

Answering our research questions In this paragraph, we answer our research questions and explain the reasoning behind our answers. The first question Q1 asks about the performance difference in in-KG tasks between KGE and NKGE methods. As we mention in the previous paragraph, the results show that there is a difference in performance in favour of NKGE. However, we could only compare the in-KG tasks of LP and TC. The second question is about the extent of performance difference and if it justifies the additional focus that KGE methods receives in the literature. Our findings show that NKGE methods perform better in 3 of our 5 test cases. On the FB15K-237 data-set in LP, the KGE have a better overall result and within our TC test result. The extent between those two favours NKGE significantly with 9.9% WN18, 12% WN18RR, and 14.6% FB15K better performance in LP. The two test cases with better performance for KGE are TC with 21.8%

FB15K-237 and LP with 0.49% FB15K-237. These results lead us to the suggestion to increase the research focus for NKGE. The next subsection covers potential future research direction that derive from our results.

Future research Through our research in this bachelor thesis, we identified gaps in the KGC literature. This includes the lack of variance in current study performance results, the exclusion of NKGE in KGE research papers results, the direct comparison between the KGC categories, and the unequal distribution of performance evaluation of in-KG. To fill these gaps in literature, future studies could evaluate the reason for the current lack of depth. Another important future study we strongly suggest is a direct performance comparison of KGE and NKGE within experiments that include all in-KG tasks we list on not modified data-sets.

7 Conclusion

In our bachelor thesis, we present the performance comparison between KGE and NKGE within the two KGC in-KG tasks LP and TC over a historic time period. We focus on data from the data-sets WN18, WN18RR, FB15K, and FB15K-237 on the filter option. We present current trends within KGC literature. With the use of a systematic literature review, we present that NKGE outperforms the embedding categories within our results. Our data leads us to the conclusion, that the current weak focus on NKGE methods in literature is not justified. We support more transparency within the comparison knowledge graph completion methods and suggest that KGE papers should include NKGE methods.

References

[1] Mehdi Ali, Max Berrendorf, Charles Tapley Hoyt, Laurent Vermue, Mikhail Galkin, Sahand Sharifzadeh, Asja Fischer, Volker Tresp, and Jens Lehmann. Bringing light into the dark: A large-scale evaluation of knowledge graph embedding models under a unified framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 44(12):8825–8845, 2022.

[2] Ivana Balažević, Carl Allen, and Timothy Hospedales. Multi-relational poincaré graph embeddings. *Conference on Neural Information Processing Systems (NeurIPS)*, 32, 2019.

[3] Ivana Balažević, Carl Allen, and Timothy Hospedales. TuckER: Tensor factorization for knowledge graph completion. 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)*, pages 5185–5194, 2019.

[4] HongYun Cai, Vincent W. Zheng, and Kevin Chen-Chuan Chang. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering*, 30(9):1616–1637, 2018.

[5] Lei Cai and Shuiwang Ji. A multi-scale approach for graph link prediction. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pages 3308–3315, 2020.

[6] Lei Cai, Jundong Li, Jie Wang, and Shuiwang Ji. Line graph neural networks for link prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2022.

[7] Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In *The Web Conference (WWW)*, pages 151–161, 2019.

[8] Zongsheng Cao, Qianqian Xu, Zhiyong Yang, Xiaochun Cao, and Qingming Huang. Dual quaternion knowledge graph embeddings. *AAAI Conference on Artificial Intelligence (AAAI)*, 35(8):6894–6902, 2021.

[9] Liang Chang, Manli Zhu, Tianlong Gu, Chenzhong Bin, Junyan Qian, and Ji Zhang. Knowledge graph embedding by dynamic translation. *IEEE Access*, 5:20898–20907, 2017.

[10] Jiajun Chen, Huarui He, Feng Wu, and Jie Wang. Topology-aware correlations between relations for inductive link prediction in knowledge graphs. *AAAI Conference on Artificial Intelligence (AAAI)*, 35:6271–6278, 2021.

[11] Xuelu Chen, Muhao Chen, Weijia Shi, Yizhou Sun, and Carlo Zaniolo. Embedding uncertain knowledge graphs. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, pages 3363–3370, 2019.

[12] Zhe Chen, Yuehan Wang, Bin Zhao, Jing Cheng, Xin Zhao, and Zongtao Duan. Knowledge graph completion: A review. *IEEE Access*, 8:192435–192456, 2020.

[13] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 1811–1818, 2018.

[14] Takuma Ebisu and Ryutaro Ichise. TorusE: Knowledge graph embedding on a lie group. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[15] Takuma Ebisu and Ryutaro Ichise. Generalized translation-based embedding of knowledge graph. *IEEE Transactions on Knowledge and Data Engineering*, 32(5):941–951, 2019.

[16] Bahare Fatemi, Siamak Ravanbakhsh, and David Poole. Improved knowledge graph embedding using background taxonomic information. *AAAI Conference on Artificial Intelligence (AAAI)*, 33(01):3526–3533, 2019.

[17] Niannian Guan, Dandan Song, and Lejian Liao. Knowledge graph embedding with concepts. *Knowledge-Based Systems*, 164:38–44, 2019.

[18] Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. A survey on knowledge graph-based recommender systems. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[19] Shu Guo, Quan Wang, Bin Wang, Lihong Wang, and Li Guo. Sse: Semantically smooth embedding for knowledge graphs. *IEEE Transactions on Knowledge and Data Engineering*, 29(4):884–897, 2016.

[20] Shu Guo, Quan Wang, Lihong Wang, Bin Wang, and Li Guo. Jointly embedding knowledge graphs and logical rules. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 192–202, 2016.

[21] Shu Guo, Quan Wang, Lihong Wang, Bin Wang, and Li Guo. Knowledge graph embedding with iterative guidance from soft rules. *AAAI Conference on Artificial Intelligence (AAAI)*, 32, 2018.

[22] Xu Han, Zhiyuan Liu, and Maosong Sun. Neural knowledge acquisition via mutual attention between knowledge graph and text. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 32, 2018.

[23] Linus Hermansson, Tommi Kerola, Fredrik Johansson, Vinay Jethava, and Devdatt Dubhashi. Entity disambiguation in anonymized graphs using graph kernels. In *International Conference on Information and Knowledge Management (CIKM)*, pages 1037–1046, 2013.

[24] Marcel Hildebrandt, Jorge Andres Quintero Serna, Yunpu Ma, Martin Ringsquandl, Mitchell Joblin, and Volker Tresp. Reasoning on knowledge graphs with debate dynamics. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pages 4123–4131, 2020.

[25] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. Knowledge graph embedding via dynamic mapping matrix. 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)*, volume 1, pages 687–696, 2015.

[26] Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. Knowledge graph completion with adaptive sparse transfer matrix. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 985–991, 2016.

[27] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and S Yu Philip. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):494–514, 2022.

[28] Shengbin Jia, Yang Xiang, Xiaojun Chen, and Kun Wang. Triple trustworthiness measurement for knowledge graph. In *The Web Conference (WWW)*, pages 2865–2871, 2019.

[29] Yantao Jia, Yuanzhuo Wang, Xiaolong Jin, and Xueqi Cheng. Path-specific knowledge graph embedding. *Knowledge-Based Systems*, 151:37–44, 2018.

[30] Seyed Mehran Kazemi and David Poole. SimplE embedding for link prediction in knowledge graphs. *Conference on Neural Information Processing Systems (NeurIPS)*, 31, 2018.

[31] Ruiping Li and Xiang Cheng. DIVINE: A generative adversarial imitation learning framework for knowledge graph reasoning. 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)*, pages 2642–2651, 2019.

[32] Weidong Li, Xinyu Zhang, Yaqian Wang, Zhihuan Yan, and Rong Peng. Graph2seq: Fusion embedding learning for knowledge graph completion. *IEEE Access*, 7:157960–157971, 2019.

[33] Zixuan Li, Xiaolong Jin, Saiping Guan, Yuanzhuo Wang, and Xueqi Cheng. Path reasoning over knowledge graph: A multi-agent and reinforcement learning based method. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 929–936, 2018.

[34] Yuanfei Luo, Quan Wang, Bin Wang, and Li Guo. Context-dependent knowledge graph embedding. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1656–1661, 2015.

[35] Xin Lv, Yuxian Gu, Xu Han, Lei Hou, Juanzi Li, and Zhiyuan Liu. Adapting meta knowledge graph information for multi-hop reasoning over few-shot relations. 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)*, pages 3376–3381, 2019.

[36] Xin Lv, Xu Han, Lei Hou, Juanzi Li, Zhiyuan Liu, Wei Zhang, Yichi Zhang, Hao Kong, and Suhui Wu. Dynamic anticipation and completion for multi-hop reasoning over sparse knowledge graph. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[37] Changsung Moon, Paul Jones, and Nagiza F Samatova. Learning entity type embeddings for knowledge graph completion. In *International Conference on Information and Knowledge Management (CIKM)*, pages 2215–2218, 2017.

[38] Tam Thanh Nguyen, Thanh Trung Huynh, Hongzhi Yin, Vinh Van Tong, Darnbi Sakong, Bolong Zheng, and Quoc Viet Hung Nguyen. Entity alignment for knowledge graphs with multi-order convolutional networks. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[39] Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. Holographic embeddings of knowledge graphs. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2016.

[40] Hao Nie, Xianpei Han, Ben He, Le Sun, Bo Chen, Wei Zhang, Suhui Wu, and Hao Kong. Deep sequence-to-sequence entity matching for heterogeneous entity resolution. In *International Conference on Information and Knowledge Management (CIKM)*, pages 629–638, 2019.

[41] Pouya Ghiasnezhad Omran, Kewen Wang, and Zhe Wang. An embedding-based approach to rule learning in knowledge graphs. *IEEE Transactions on Knowledge and Data Engineering*, 33(4):1348–1359, 2021.

[42] Matthew J Page, Joanne E McKenzie, Patrick M Bossuyt, Isabelle Boutron, Tammy C Hoffmann, Cynthia D Mulrow, Larissa Shamseer, Jennifer M Tetzlaff, Elie A Akl, Sue E Brennan, Roger Chou, Julie Glanville, Jeremy M Grimshaw, Asbjørn Hróbjartsson, Manoj M Lalu, Tianjing Li, Elizabeth W Loder, Evan Mayo-Wilson, Steve McDonald, Luke A McGuinness, Lesley A Stewart, James Thomas, Andrea C Tricco, Vivian A Welch, Penny Whiting, and David Moher. The prisma 2020 statement: an updated guideline for reporting systematic reviews. *BMJ*, 372, 2021.

[43] Shichao Pei, Lu Yu, Robert Hoehndorf, and Xiangliang Zhang. Semi-supervised entity alignment via knowledge graph embedding with awareness of degree difference. In *The Web Conference (WWW)*, page 3130–3136, 2019.

[44] Aritran Piplai, Sudip Mittal, Anupam Joshi, Tim Finin, James Holt, and Richard Zak. Creating cybersecurity knowledge graphs from malware after action reports. *IEEE Access*, 2020.

[45] Wei Qian, Cong Fu, Yu Zhu, Deng Cai, and Xiaofei He. Translating embeddings for knowledge graph completion with relation attention mechanism. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4286–4292, 2018.

[46] Marco Rospocher, Marieke van Erp, Piek Vossen, Antske Fokkens, Itziar Aldabe, German Rigau, Aitor Soroa, Thomas Ploeger, and Tessel Bogaard. Building event-centric knowledge graphs from news. *Journal of Web Semantics*, 37:132–151, 2016.

[47] Haseeb Shah, Johannes Villmow, Adrian Ulges, Ulrich Schwanecke, and Faisal Shafait. An open-world extension to knowledge graph completion models. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, pages 3044–3051, 2019.

[48] Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. End-to-end structure-aware convolutional networks for knowledge base completion. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 3060–3067, 2019.

[49] Amit Sheth, Swati Padhee, and Amelie Gyrard. Knowledge graphs and knowledge networks: the story in brief. *IEEE Internet Computing*, 23(4):67–75, 2019.

[50] Baoxu Shi and Tim Weninger. Discriminative predicate path mining for fact checking in knowledge graphs. *Knowledge-Based Systems*, 104:123–133, 2016.

[51] Baoxu Shi and Tim Weninger. Proje: Embedding projection for knowledge graph completion. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 31, 2017.

[52] Baoxu Shi and Tim Weninger. Open-world knowledge graph completion. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 1957–1964, 2018.

[53] Dezhao Song, Frank Schilder, Shai Hertz, Giuseppe Saltini, Charese Smiley, Phani Nivarthi, Oren Hazai, Dudi Landau, Mike Zaharkin, Tom Zielund, et al. Building and querying an enterprise knowledge graph. *IEEE Transactions on Services Computing*, 12(3):356–369, 2017.

[54] George Stoica, Otilia Stretcu, Emmanouil Antonios Platanios, Tom Mitchell, and Barnabas Poczos. Contextual parameter generation for knowledge graph link prediction. *AAAI Conference on Artificial Intelligence (AAAI)*, 34(03), 2020.

[55] Zequn Sun, Chengming Wang, Wei Hu, Muhao Chen, Jian Dai, Wei Zhang, and Yuzhong Qu. Knowledge graph alignment network with gated multi-hop neighborhood aggregation. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pages 222–229, 2020.

[56] Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. Non-parametric estimation of multiple embeddings for link prediction on dynamic knowledge graphs. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

[57] Prayag Tiwari, Hongyin Zhu, and Hari Mohan Pandey. Dapath: Distance-aware knowledge graph reasoning based on deep reinforcement learning. *Neural Networks*, 135:1–12, 2021.

[58] Bayu Distiawan Trisedya, Jianzhong Qi, and Rui Zhang. Entity alignment between knowledge graphs using attribute embeddings. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, pages 297–304, 2019.

[59] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, Nilesh Agrawal, and Partha Talukdar. Interacte: Improving convolution-based knowledge graph embeddings by increasing feature interactions. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pages 3009–3016, 2020.

[60] Guojia Wan and Bo Du. Gaussianpath:a bayesian multi-hop reasoning framework for knowledge graph reasoning. *AAAI Conference on Artificial Intelligence (AAAI)*, 35(5):4393–4401, 2021.

[61] Peifeng Wang, Jialong Han, Chenliang Li, and Rong Pan. Logic attention based neighborhood aggregation for inductive knowledge graph embedding. *AAAI Conference on Artificial Intelligence (AAAI)*, 33(01):7152–7159, 2019.

[62] Peifeng Wang, Shuangyin Li, and Rong Pan. Incorporating gan for negative sampling in knowledge representation learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 32, 2018.

[63] Qi Wang, Yongsheng Hao, and Jie Cao. Adrl: An attention-based deep reinforcement learning framework for knowledge graph reasoning. *Knowledge-Based Systems*, 197:105910, 2020.

[64] Qi Wang, Yuede Ji, Yongsheng Hao, and Jie Cao. Grl: Knowledge graph completion with gan-based reinforcement learning. *Knowledge-Based Systems*, 209:106421, 2020.

[65] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743, 2017.

[66] Rui Wang, Bicheng Li, Shengwei Hu, Wenqian Du, and Min Zhang. Knowledge graph embedding via graph attenuated attention networks. *IEEE Access*, 8:5212–5224, 2019.

[67] Ruijie Wang, Yuchen Yan, Jialu Wang, Yuting Jia, Ye Zhang, Weinan Zhang, and Xinbing Wang. Acekg: A large-scale knowledge graph for academic data mining. In *International Conference on Information and Knowledge Management (CIKM)*, pages 1487–1490, 2018.

[68] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. Explainable reasoning over knowledge graphs for recommendation. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, pages 5329–5336, 2019.

[69] Zhuoyu Wei, Jun Zhao, and Kang Liu. Mining inference formulas by goal-directed random walks. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1379–1388, 2016.

[70] H. Xiao, M. Huang, L. Meng, and X. Zhu. Ssp: Semantic space projection for knowledge graph embedding with text descriptions. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 3104–3110, 2017.

[71] R. Xie, Z. Liu, J. Jia, H. Luan, and M. Sun. Representation learning of knowledge graphs with entity descriptions. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 2659–2665, 2016.

[72] R. Xie, Z. Liu, H. Luan, and M. Sun. Image-embodied knowledge representation learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3140–3146, 2016.

[73] Ruobing Xie, Zhiyuan Liu, Fen Lin, and Leyu Lin. Does william shakespeare really write hamlet? knowledge representation learning with confidence. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 32, 2018.

[74] Ruobing Xie, Zhiyuan Liu, Maosong Sun, et al. Representation learning of knowledge graphs with hierarchical types. In *International Joint Conference on Artificial Intelligence (IJCAI)*, volume 2016, pages 2965–2971, 2016.

[75] Shihui Yang, Jidong Tian, Honglun Zhang, Junchi Yan, Hao He, and Yaohui Jin. Transms: Knowledge graph embedding for complex relations by multidirectional semantics. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1935–1942, 2019.

[76] Liang Yao, Yin Zhang, Baogang Wei, Zhe Jin, Rui Zhang, Yangyang Zhang, and Qinfei Chen. Incorporating knowledge graph embeddings

into topic modeling. *AAAI Conference on Artificial Intelligence (AAAI)*, 31, 2017.

[77] Rui Ye, Xin Li, Yujie Fang, Hongyu Zang, and Mingzhong Wang. A vectorized relational graph convolutional network for multi-relational network alignment. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4135–4141, 2019.

[78] Donghan Yu, Chenguang Zhu, Yiming Yang, and Michael Zeng. Jaket: Joint pre-training of knowledge graph and language understanding. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 36, pages 11630–11638, 2022.

[79] Wen Zhang, Bibek Paudel, Liang Wang, Jiaoyan Chen, Hai Zhu, Wei Zhang, Abraham Bernstein, and Huajun Chen. Iteratively learning embeddings and rules for knowledge graph reasoning. In *The Web Conference (WWW)*, page 2366–2377, 2019.

[80] Zhanqiu Zhang, Jianyu Cai, Yongdong Zhang, and Jie Wang. Learning hierarchy-aware knowledge graph embeddings for link prediction. *AAAI Conference on Artificial Intelligence (AAAI)*, 34:3065–3072, 2020.

[81] Zhao Zhang, Fuzhen Zhuang, Meng Qu, Fen Lin, and Qing He. Knowledge graph embedding with hierarchical relation structure. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3198–3207, 2018.

[82] Zhao Zhang, Fuzhen Zhuang, Hengshu Zhu, Zhiping Shi, Hui Xiong, and Qing He. Relational graph neural network with hierarchical attention for knowledge graph completion. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pages 9612–9619, 2020.

[83] Zhaoli Zhang, Zhifei Li, Hai Liu, and Neal N Xiong. Multi-scale dynamic convolutional network for knowledge graph embedding. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[84] Xuejiao Zhao, Zhenchang Xing, Muhammad Ashad Kabir, Naoya Sawada, Jing Li, and Shang-Wei Lin. Hdskg: Harvesting domain specific knowledge graph from content of webpages. In *2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 56–67, 2017.

[85] Xiaofei Zhou, Qiannan Zhu, Ping Liu, and Li Guo. Learning knowledge embeddings by combining limit-based scoring loss. In *International*

Conference on Information and Knowledge Management (CIKM)*, pages 1009–1018, 2017.

[86] Ganggao Zhu and Carlos A Iglesias. Computing semantic similarity of concepts in knowledge graphs. *IEEE Transactions on Knowledge and Data Engineering*, 29:72–85, 2017.

[87] Ganggao Zhu and Carlos A Iglesias. Sematch: Semantic similarity framework for knowledge graphs. *Knowledge-Based Systems*, 130:30–32, 2017.

[88] Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal A. C. Xhonneux, and Jian Tang. Neural bellman-ford networks: A general graph neural network framework for link prediction. *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

APPENDICES

A PRISMA 2020 CHECKLIST

Section/topic	#	Checklist item	Location(s) Reported
INFORMATION SOURCES AND METHODS			
Database name	1	Name each individual database searched, stating the platform for each.	4.1 Search strategy
Multi-database searching	2	If databases were searched simultaneously on a single platform, state the name of the platform, listing all of the databases searched.	4.1 Search strategy
Study registries	3	List any study registries searched.	N/a
Online resources and browsing	4	Describe any online or print source purposefully searched or browsed (e.g., tables of contents, print conference proceedings, web sites), and how this was done.	N/a
Citation searching	5	Indicate whether cited references or citing references were examined, and describe any methods used for locating cited/citing references (e.g., browsing reference lists, using a citation index, setting up email alerts for references citing included studies).	4.2 Study selection
Contacts	6	Indicate whether additional studies or data were sought by contacting authors, experts, manufacturers, or others.	N/a
Other methods	7	Describe any additional information sources or search methods used.	N/a
SEARCH STRATEGIES			
Full search strategies	8	Include the search strategies for each database and information source, copied and pasted exactly as run.	4.1 Search strategy
Limits and restrictions	9	Specify that no limits were used, or describe any limits or restrictions applied to a search (e.g., date or time period, language, study design) and provide justification for their use.	4.3 Data extraction
Search filters	10	Indicate whether published search filters were used (as originally designed or modified), and if so, cite the filter(s) used.	4.1 Search strategy
Prior work	11	Indicate when search strategies from other literature reviews were adapted or reused for a substantive part or all of the search, citing the previous review(s).	N/a
Updates	12	Report the methods used to update the search(es) (e.g., rerunning searches, email alerts).	4.1 Search strategy
Dates of searches	13	For each search strategy, provide the date when the last search occurred.	N/a
PEER REVIEW			
Peer review	14	Describe any search peer review process.	N/a
MANAGING RECORDS			
Total Records	15	Document the total number of records identified from each database and other information sources.	**440**
Deduplication	16	Describe the processes and any software used to deduplicate records from multiple database searches and other information sources.	4.4 Outline and scope

PRISMA-S: An Extension to the PRISMA Statement for Reporting Literature Searches in Systematic Reviews
Rethlefsen ML, Kirtley S, Waffenschmidt S, Ayala AP, Moher D, Page MJ, Koffel JB, PRISMA-S Group.
Last updated February 27, 2020.

Figure 16: PRISMA checklist for our systematic literature review

B Statistics summary of paper selection with data entries

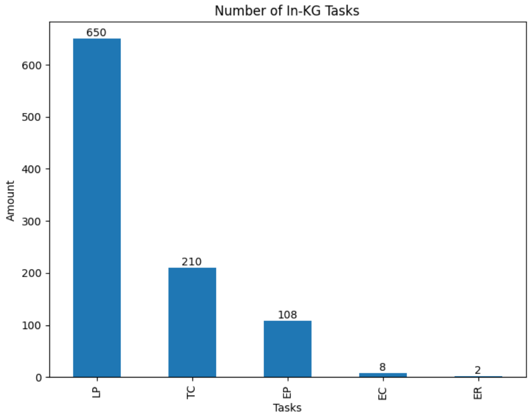

Figure 17: Number of data entries for each In-KG task within our performance data

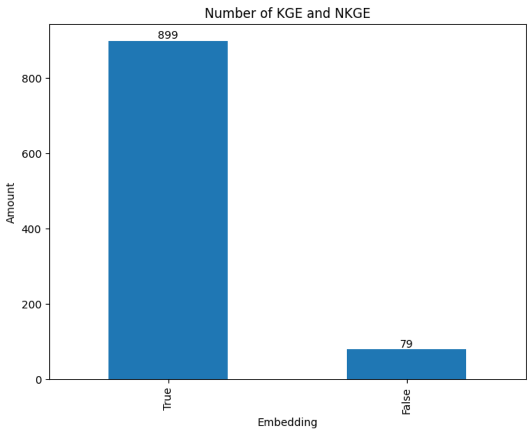

Figure 18: Number of data entries which include embedding methods within our performance data

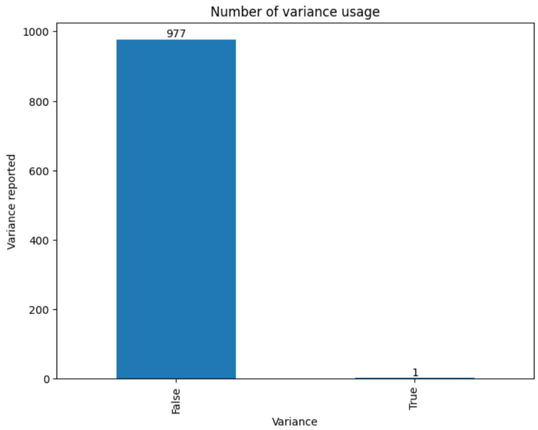

Figure 19: Number of data entries which include variance within our performance data

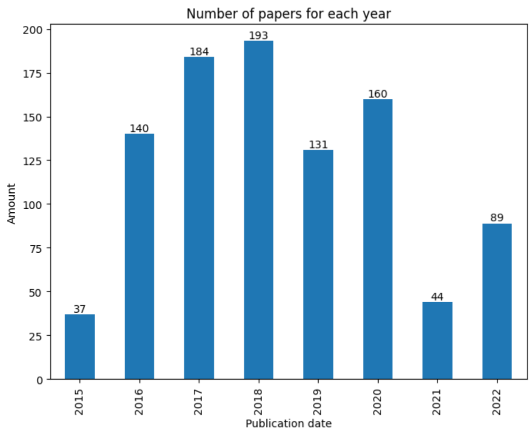

Figure 20: Number of data entries for each year within our performance data

C Collection of granular time series plots

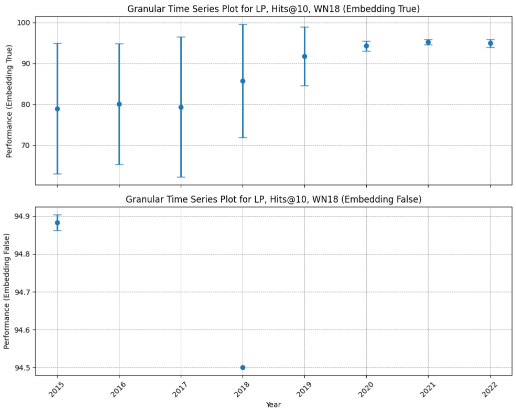

Figure 21: Granular time series plot for LP on the data set WN18

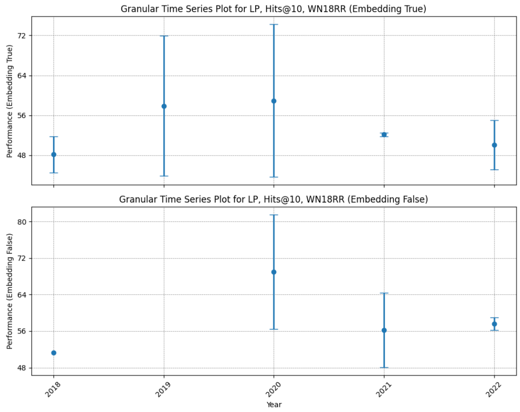

Figure 22: Granular time series plot for LP on the data set WN18RR

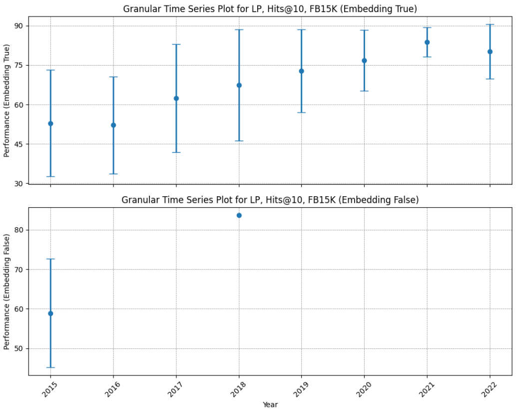

Figure 23: Granular time series plot for LP on the data set FB15K

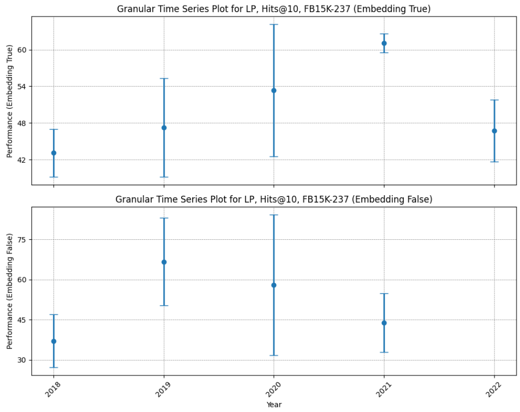

Figure 24: Granular time series plot for LP on the data set FB15K-237

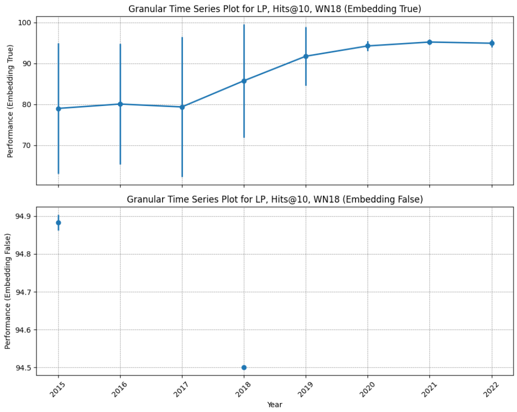

Figure 25: Line plot with error bar for LP on the data set WN18

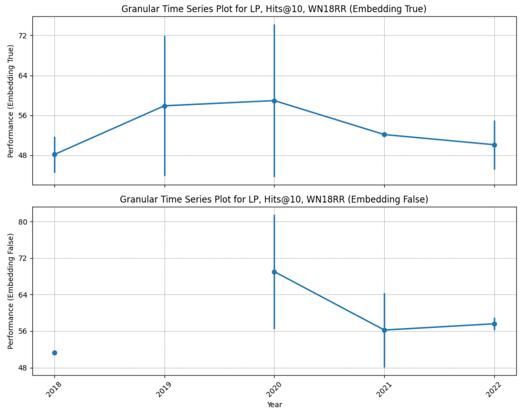

Figure 26: Line plot with error bar for LP on the data set WN18RR

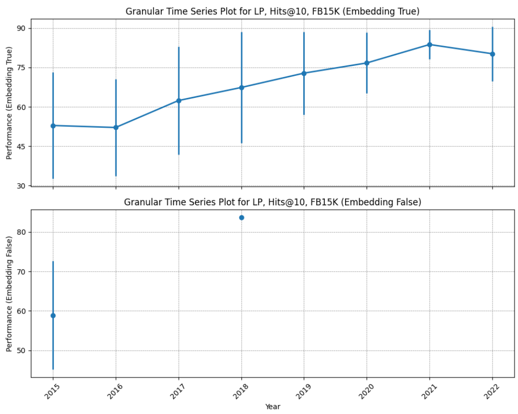

Figure 27: Line plot with error bar for LP on the data set FB15K

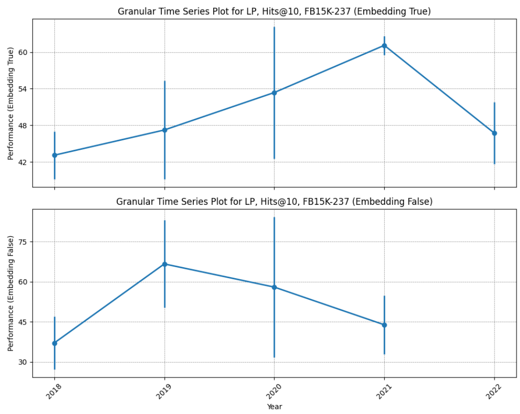

Figure 28: Line plot with error bar for LP on the data set FB15K-237

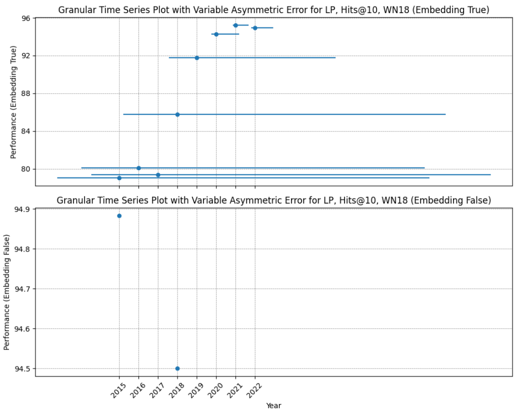

Figure 29: Asymmetric error for LP on the data set WN18

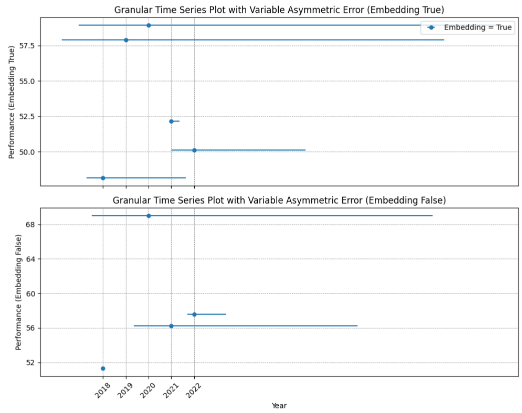

Figure 30: Asymmetric error for LP on the data set WN18RR

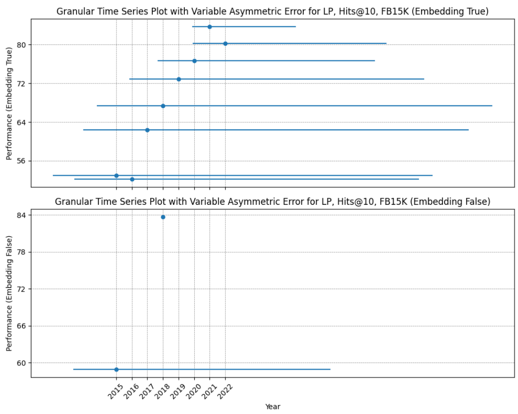

Figure 31: Asymmetric error for LP on the data set FB15K

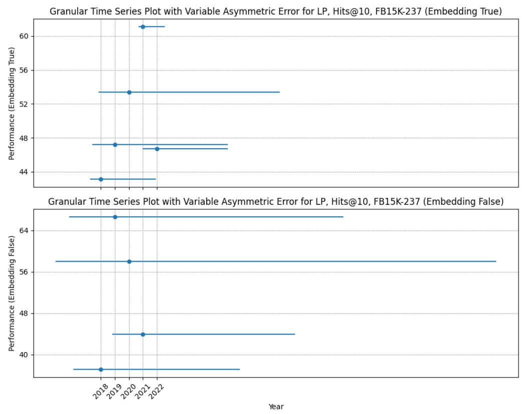

Figure 32: Asymmetric error for LP on the data set FB15K-237