HybridFC: A Hybrid Fact-Checking Approach for Knowledge Graphs

Umair Qudus(✉), Michael Röder, Muhammad Saleem, and Axel-Cyrille Ngonga Ngomo

DICE Group, Department of Computer Science, Universität Paderborn, Paderborn, Germany
{umair.qudus,michael.roeder,axel.ngonga}@uni-paderborn.de,
saleem@mail.uni-paderborn.de
https://dice-research.org/

Abstract. We consider fact-checking approaches that aim to predict the veracity of assertions in knowledge graphs. Five main categories of fact-checking approaches for knowledge graphs have been proposed in the recent literature, of which each is subject to partially overlapping limitations. In particular, current text-based approaches are limited by manual feature engineering. Path-based and rule-based approaches are limited by their exclusive use of knowledge graphs as background knowledge, and embedding-based approaches suffer from low accuracy scores on current fact-checking tasks. We propose a hybrid approach—dubbed HybridFC—that exploits the diversity of existing categories of fact-checking approaches within an ensemble learning setting to achieve a significantly better prediction performance. In particular, our approach outperforms the state of the art by 0.14 to 0.27 in terms of Area Under the Receiver Operating Characteristic curve on the FactBench dataset. Our code is open-source and can be found at https://github.com/dice-group/HybridFC.

Keywords: Fact checking · Ensemble learning · Knowledge graph veracity

1 Introduction

Knowledge graphs (KGs) are an integral part of the Web. A recent crawl of 3.2 billion HTML pages found over 82 billion RDF statements distributed over roughly half of the Web pages that were crawled.[1] The increasing adoption of RDF at Web scale is further corroborated by the Linked Open Data cloud, which now contains over 10,000 KGs with more than 150 billion assertions and 3 billion entities.[2] Large-scale KGs like WikiData [30], DBpedia [2], Knowledge Vault [13], and YAGO [43] contain billions of assertions, and describe millions of entities. They are being used as background knowledge in a growing number of applications, including healthcare [26], autonomous chatbots [1], and in-flight entertainment [31]. However, it is well established that current KGs are partially incorrect. For example, roughly 20% of DBpedia's assertions are

[1] http://webdatacommons.org/structureddata/2021-12/stats/stats.html.

[2] https://lod-cloud.net/.

U. Sattler et al. (Eds.): ISWC 2022, LNCS 13489, pp. 462–480, 2022.
https://doi.org/10.1007/978-3-031-19433-7_27

assumed to be false in the literature [20,39]. Fostering the further uptake of KGs at Web scale hence requires the development of highly accurate approaches that are able to predict the veracity of the assertions found in KGs in an automated fashion. We call such approaches fact-checking approaches.

In general, fact checking can be understood as the task of computing the likelihood that a given assertion is true [6]. Various categories of automatic approaches have been proposed for this task. These categories include but are not limited to text-based [20, 46], path-based [9,19,41,45,48], rule-based [16,17,27], and embedding-based [7,29] approaches. State-of-the-art instantiations of these categories of approaches are faced with a set of common limitations. In particular,

(1) Current text-based approaches rely on manual feature engineering [20,39,46], which is time-consuming, and has been shown to be suboptimal w.r.t. their prediction performance by representation learning approaches [5].
(2) Path-based approaches rely on the availability of (short) paths in the KG between the entities that are part of the given assertion [48].
(3) Approaches that rely on KGs as background knowledge, i.e., path-, rule- and embedding-based approaches, have to take the open-world assumption (OWA) into account when determining the veracity of the given assertion [48].
(4) Embedding-based approaches [42] encounter limitations with respect to their accuracy [22] as well as their scalability [50].

We alleviate these limitations by exploiting the principles of diversity and accuracy known from ensemble learning. Our approach, dubbed HybridFC, overcomes the drawbacks of individual categories of approaches by leveraging the advantages of other categories of approaches. For example, we replace the manual feature engineering of the text-based approaches by exploiting embeddings. To the best of our knowledge, we are the first to propose the combination of text-, path- and embedding-based fact-checking approaches in an ensemble learning setting.

The contributions of this work are as follows:

- We use pre-trained KG embedding and sentence transformer models, and take advantage of transfer learning to reuse them for the task of fact checking.
- We study the performance of different fact-checking approaches in isolation and in combination, and show that the joint use of multiple categories of approaches within an ensemble learning setting often leads to an improved performance.
- We benchmark our approach on two recent fact-checking datasets, i.e., FactBench and BirthPlace/DeathPlace (BD). Our experiments suggest that our hybrid approach outperforms other text-, path-, rule- and embedding-based approaches by at least 0.14 area under the curve (AUROC) on average on the FactBench dataset. It is ranked 3rd on the smaller BD dataset.

The rest of this paper is structured as follows. In Sect. 2, we introduce the notation required to understand the rest of the paper. In Sect. 3, we give related work and motivate our work using a real-world example. In Sect. 4, we present HybridFC. Thereafter, the evaluation datasets and metric used are presented in Sect. 5. We then discuss our results in Sect. 6. In Sect. 7, we present an ablation study of our approach. Finally, we conclude and discuss potential future work in Sect. 8.

2 Preliminaries

In this section, we define the terminology and notation used throughout this paper. We build upon the definition of fact checking for KGs suggested in [46]:

Definition 1 (Fact Checking). *Given an assertion, a reference KG $\mathcal{G}$, and/or a reference corpus, fact checking is the task of computing the likelihood that the given assertion is true or false [46].*

Throughout this work, we rely on RDF KGs:

Definition 2 (RDF Knowledge Graph). *An RDF KG $\mathcal{G}$ is a set of RDF triples $\mathcal{G} \subseteq (\mathbb{E} \cup \mathbb{B}) \times \mathbb{P} \times (\mathbb{E} \cup \mathbb{B} \cup \mathbb{L})$, where each triple $(s, p, o) \in \mathcal{G}$ comprises a subject s, a predicate p, and an object o. $\mathbb{E}$ is the set of all RDF resource IRIs, $\mathbb{B}$ the set of all blank nodes, $\mathbb{P} \subseteq \mathbb{E}$ the set of all RDF predicates, and $\mathbb{L}$ the set of all literals [47].*

In our approach, we use multiple representations of RDF KGs. In addition to their representation as sets of assertions, we also exploit representations in continuous vector spaces, called embeddings [10,50].

Definition 3 (KG Embeddings). *A KG embedding function φ maps a KG $\mathcal{G}$ to a continuous vector space. Given an assertion (s, p, o), $\varphi(s), \varphi(p)$, and $\varphi(o)$ stand for the embedding of the subject, predicate, and object, respectively. Some embedding models map the predicate embedding into a vector space that differs from the space wherein $\varphi(s)$ and $\varphi(o)$ are mapped. For those models, we use $\varphi^*(p)$ to denote predicate embeddings.*

Different embedding-based approaches use different scoring functions to compute embeddings [50]. The approaches considered in this paper are shown in Table 1.

Table 1. Scoring functions of different embedding-based approaches used in this paper. $\otimes$ stands for the quaternion multiplication, $\mathbb{R}$ for the space of real numbers, $\mathbb{H}$ for the space of quaternions, $\mathbb{C}$ for the complex numbers, Re for the real part of a complex number, Im for the imaginary part of a complex number, conv for the convolution operator, $\overline{\varphi(o)}$ for the complex conjugate of $\varphi(o)$, q is the length of embedding vectors, $\cdot$ for the dot product and $||\cdot||_2$ for the L2 norm.

Approach	Scoring function	VectorSpace	Regularizer
TransE	$\|\|(\varphi(s) + \varphi(p)) - \varphi(o)\|\|_2$	$\varphi(s), \varphi(p), \varphi(o) \in \mathbb{R}^q$	L2
ComplEx	$\mathrm{Re}\left(< \varphi(s), \varphi(p), \overline{\varphi(o)} >\right)$	$\varphi(s), \varphi(p), \varphi(o) \in \mathbb{C}^q$	Weighted L2
QMult	$\varphi(s) \otimes \varphi(p) \cdot \varphi(o)$	$\varphi(s), \varphi(p), \varphi(o) \in \mathbb{H}^q$	Weighted L2
ConEx	$\mathrm{Re}(\langle \mathrm{conv}(\varphi(s), \varphi(p)), \varphi(s), \varphi(p), \overline{\varphi(o)} \rangle)$	$\varphi(s), \varphi(p), \varphi(o) \in \mathbb{C}^q$	Dropout, BatchNorm

Definition 4 (Sentence Embedding Model). *A sentence embedding model maps the natural language sentence t to a continuous vector space [37]. Let b be the embedding function and let $T = (t_1, \ldots, t_k)$ be a list of k sentences. We create the embedding vector for T by concatenating the embedding vectors of the single sentences.*

3 Related Work

We divide the existing fact-checking approaches into 5 categories: text-based [20,46], path-based [41,47], rule-based [16,17,27], KG-embedding-based [7,24,29], and hybrid approaches [14,15,28]. In the following, we give a brief overview of state-of-the-art approaches in each category along with their limitations.

3.1 Text-Based Approaches

Approaches in this category validate a given assertion by searching for evidence in a reference text corpus. FactCheck [46] and DeFacto [20] are two instantiations of this category. Both approaches search for pieces of text that can be used as evidence to support the given assertion by relying on RDF verbalisation techniques. TISCO [39] relies on a temporal extension of DeFacto. All three approaches rely on a set of manually engineered features to compute a vectorial representation of the texts they retrieved as evidence. This manual feature engineering often leads to a suboptimal vectorial representation of textual evidence [5]. In contrast, we propose the use of embeddings to represent pieces of evidence gathered from text as vectors. First, this ensures that our approach is aware of the complete piece of textual evidence instead of the fragment extracted by previous approaches. Second, it removes the need to engineer features manually and hence reduces the risk of representing text with a possibly suboptimal set of manually engineered features.

3.2 Path-Based Approaches

Path-based approaches generally aim to validate the input assertion by first computing short paths from the assertion's subject to its object within the input KG. These paths are then used to score the input assertion. Most of the state-of-the-art path-based approaches, such as COPAAL [47], Knowledge stream [41], PRA [19], SFE [18], and KG-Miner [40] rely on RDF semantics (e.g., class subsumption hierarchy, domain and range information) to filter useful paths. However, the T-Box of a large number of KGs provides a limited number of RDFS statements. Furthermore, it may also be the case that no short paths can be found within the reference KG, although the assertion is correct [47]. In these scenarios, path-based approaches fail to predict the veracity of the given assertion correctly.[3]

3.3 Rule-based Approaches

State-of-the-art rule-based models such as KV-Rule [25], AMIE [16,17,27], OP [8], and RuDiK [34] extract association rules to perform fact checking or fact prediction on KGs. To this end, they often rely on reasoning [27,44]. These approaches are limited by the knowledge contained within the KG, and mining rules from large-scale KGs can be a very slow process in terms of runtime (e.g., OP takes ≥ 45 hours on DBpedia [27]).

[3] For the assertion *award*_00135 from the FactBench, COPAAL produces a score of 0.0 as it is unable to find a path between the assertion's subject and its object.

3.4 Embedding-Based Approaches

Embedding-based approaches use a mapping function to represent the input KG in a continuous low-dimensional vector space [7,12,21,24,29,42,49]. For example, Esther [42] uses compositional embeddings to compute likely paths between resources. TKGC [21] checks the veracity of assertions extracted from the Web before adding them to a given KG. The veracity of assertions is calculated by creating a KG embedding model and learning a scoring function to compute the veracity of these assertions. In general, embedding-based approaches are mainly limited by the knowledge contained within the continuous representation of the KG. Therefore, these approaches encounter limitations with respect to their accuracy in fact-checking scenarios [22] as well as their scalability when applied to large-scale KGs [50].

3.5 Hybrid Approaches

While the aforementioned categories have their limitations, they also come with their own strengths. Consider the assertion in Listing 1.1. The text-based approach FactCheck cannot find evidence for the assertion. A possible reason might be that West Hollywood is not mentioned on the Wikipedia page of Johnny Carson. However, COPAAL finds evidence in the form of corroborative paths that connect the subject and the object in DBpedia. For example, the first corroborative path in this particular example from FactBench [20] encodes that if two individuals share a death place, then they often share several death places. While this seems counter-intuitive, one can indeed have several death places by virtue of the part-of relation between geo-spatial entities, e.g., one's death places can be both the Sierra Towers and West Hollywood. In our second example shown in Listing 1.2, COPAAL is not able to find any relevant paths between the subject and the object. This shows one of the weaknesses of COPAAL which does not perform well for rare events, e.g., when faced with the `:award` property [47]. In contrast, TransE [7] is able to classify the assertion as correct. These examples support our hypothesis that there is a need for a hybrid solution in which the limitations of one approach can be compensated by the other approaches.

Listing 1.1. Example 1 (correct, `death-00129.ttl` in the FactBench dataset [20]).

```
PREFIX dbr:  <http://dbpedia.org/resource/>
PREFIX dbo:  <http://dbpedia.org/ontology/>
Assertion:  dbr:Johnny_Carson dbo:deathPlace dbr:West_Hollywood,_California

FactCheck Result: Score: 0.0
Proofs: [no proofs found]
========================================================
COPAAL Result: Score: 0.99
Proofs: evidence paths:[
evidence path 1: "predicate path: dbo:deathPlace/^dbo:deathPlace/dbo:deathPlace",
evidence path 2: "predicate path: dbo:deathPlace/^dbo:recordedIn/dbo:recordedIn",
...]
```

Listing 1.2. Example 2 (correct, `award-00135.ttl` in the FactBench dataset [20]).

```
Assertion:  dbr:T._S._Eliot dbo:award dbr:Nobel_Prize_in_Literature

COPAAL Result: Score: 0.0
Proofs: evidence paths: [no paths found]
========================================================
TransE Result: Score: 0.90
```

FACTY [28], ExFaKT [15], and Tracy [14] are hybrid approaches that exploit structured as well as textual reference knowledge to find the human-comprehensible explanations for a given assertion. ExFaKT and Tracy[4] make use of rules mined from the KG. A given assertion is assumed to be correct if it fulfills all conditions of one of the mined rules. These conditions can be fulfilled by facts from the KG or by texts retrieved from the Web. The output of these approaches is not a veracity score. Rather, they produce human-comprehensible explanations to support human fact-checkers. Furthermore, these approaches are not designed for ensemble learning settings. They incorporate a text search merely to find support for the rules they generate. As such, they actually address different problem statements than the one addressed herein. FACTY leverages textual reference and path-based techniques to find supporting evidence for each triple, and subsequently predicts the correctness of each triple based on the found evidence. Like Tracy and ExFaKT, FACTY only combines two different categories and mainly focuses on generating human-comprehensible explanations for candidate facts. To the best of our knowledge, our approach is the first approach that uses approaches from three different categories with the focus on automating the fact-checking task.

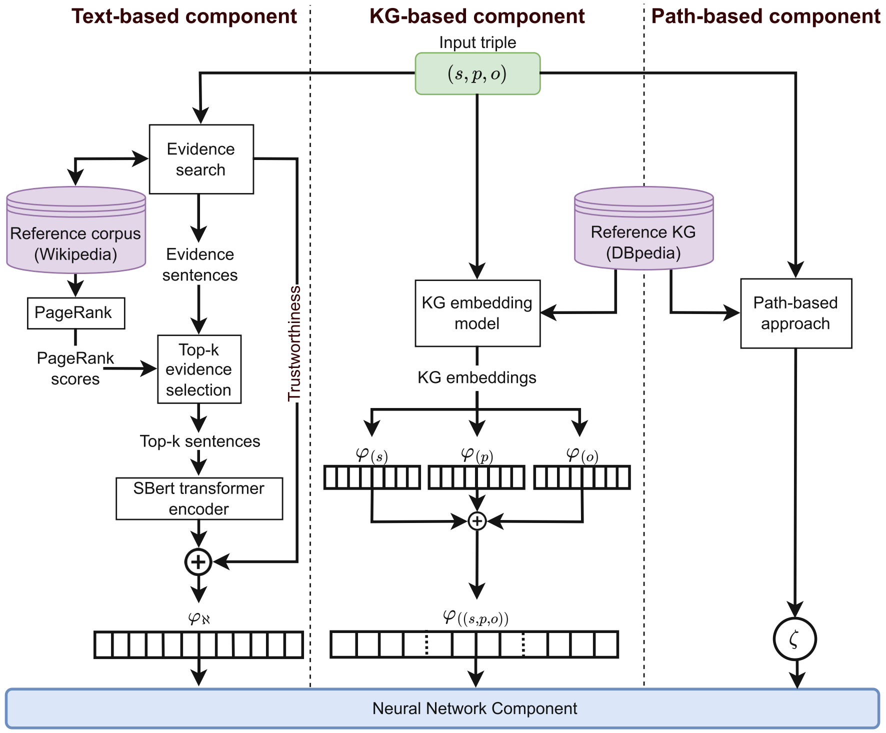

Fig. 1. Architecture of HybridFC. The purple color represents reference knowledge. The green color marks the input assertion. KG stands for knowledge graph. (Color figure online)

[4] https://www.mpi-inf.mpg.de/impact/exfakt#Tracy.

4 Methodology

The main idea behind our approach, HybridFC, is to combine fact-checking approaches from different categories. To this end, we created components for a text-based, a path-based and a KG embedding-based fact-checking algorithm. Figure 1 depicts a high-level architecture of our approach. We fuse the results from the three components and feed them into a neural network component, which computes a final veracity score. In the following, we first describe the three individual components of our approach in detail. Thereafter, we describe the neural network component that merges their results.

4.1 Text-Based Component

Text-based approaches typically provide a list of scored text snippets that provide evidence for the given assertion, together with a link to the source of these snippets and a trustworthiness score [20,46]. The next step is to use machine learning on these textual evidence snippets to evaluate a given assertion. In HybridFC, we refrain from using the machine learning module of text-based approaches. Instead, we compute an ordering for the list of text snippets returned by text-based approaches. To this end, we first determine the PageRank scores for all articles in the reference corpus [35] and select evidence sentences. Our evidence sentence selection module is based on the following hypothesis: ”Documents (websites) with higher PageRank score provide better evidence sentences”. Ergo, once provided with scored text snippets by a text-based approach, we select the top-k evidence sentences coming from documents with top-k PageRank scores. To each text snippet, we assign the PageRank score of its source article. Then, we sort the list of text snippets and use the k snippets with the highest PageRank score.

We convert each of the selected snippets t_i into a continuous vector representation using a sentence embedding model. We concatenate these sentence embeddings along with the trustworthiness scores [32] of their respective sources to create a single vector $\varphi_{\aleph}$. In short:

$$\varphi_{\aleph} = \bigoplus_{i=1}^{k} (b(t_i) \oplus \tau_i)\,, \tag{1}$$

where $\oplus$ stands for the concatenation of vectors, $b(t_i)$ is the sentence embeddings of t_i and τ_i is the trustworthiness score of t_i. Our approach can make use of any text-based fact-checking approach that provides text snippets and a trustworthiness score, and allows us to compute PageRank score. Moreover, we can use any sentence embedding model. For our experiments, we adapt the state-of-the-art text-based approach FactCheck [46] as a text-based fact checking approach, and make use of a pre-trained SBert Transformer model for sentence embeddings [37].

4.2 Path-Based Component

Path-based approaches determine the veracity of a given assertion by finding evidence paths in a reference KG. Our path-based component can make use of any existing path-based approach that takes the given assertion as input together with the reference KG and creates a single veracity score ζ as output. This veracity score is the result of our path-based component. Within our experiments, we use the state-of-the-art unsupervised path-based approach COPAAL [48].

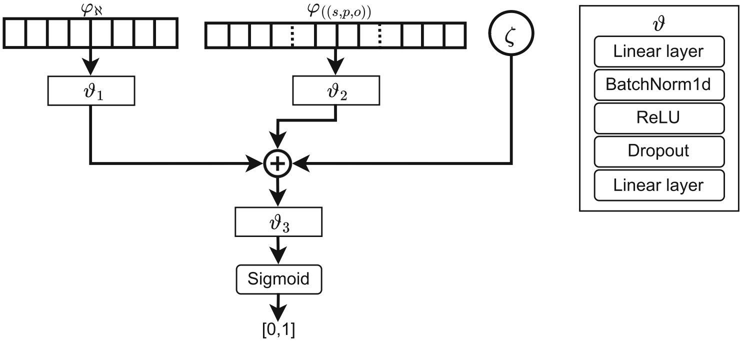

Fig. 2. Left: Overview of the architecture of HybridFC's neural network component. Right: Every ϑ_i is a multi-layer perceptron module.

4.3 KG Embedding-Based Component

KG embedding-based approaches generate a continuous representation of a KG using a mapping function. Based on a given KG embedding model, we create an embedding vector for a given assertion (s, p, o) by concatenating the embedding of its elements and define the embedding mapping function for assertions $\varphi((s, p, o))$ as follows:

$$\varphi((s, p, o)) = \varphi(s) \oplus \varphi(p) \oplus \varphi(o). \tag{2}$$

In our approach, we can make use of any KG embedding approach that returns both entities and relations embeddings. However, only a few approaches provide pre-trained embeddings for large-scale KGs (e.g., DBpedia). We use all approaches that provide pre-trained embeddings for DBpedia entities and relations in our experiments.

4.4 Neural Network Component

The output of the three components above is the input to our neural network component. As depicted in Fig. 2, the neural network component consists of three multi-layer perceptron modules that we name ϑ_i.[5] Each of these modules consists of a `Linear` layer, a `Batch Normalization` layer, a `ReLU` layer, a `Dropout` layer and a final `Linear` layer. The output of the text-based component $\varphi_{\aleph}$ is fed as input to the first module. The output of the KG embedding-based component $\varphi((s, p, o))$ is fed to the second module. The output of the 2 modules and the veracity score ζ of the path-based component are concatenated and fed to the third module. The result of the third module is used as input to

[5] During a first evaluation a simpler approach with only one multi-layer perceptron module (i.e., without ϑ_1 and ϑ_2) showed an insufficient performance.

a sigmoid function σ, which produces a final output in the range [0, 1]. The calculation of the final veracity ω score for the given assertion can be formalized as follows:

$$\omega = \sigma\left(w_\sigma^T \vartheta_3\left(\vartheta_1(\varphi_\aleph) \oplus \vartheta_2(\varphi((s,p,o))) \oplus \zeta\right)\right), \tag{3}$$

where w_σ is a weight vector that is multiplied with the output vector of the third module. Each of the three multi-layer perceptron modules (ϑ_i) is defined as follows for an input vector x:

$$\vartheta_i = W_{5,i} \times D_p(ReLU(W_{3,i} \times (BN(W_{1,i} \times x)))), \tag{4}$$

where x is an input vector, $W_{j,i}$ is the weight matrix of an affine transformation in the j-th layer of the multi-layer perceptron, × represents the matrix multiplication, `ReLU` is an activation function, D_p stands for a `Dropout` layer [51], and BN represents the `Batch Normalization` [23]. The latter is defined in the following equation:

$$BN(x') = \beta + \gamma \frac{x' - \mathrm{E}[x']}{\sqrt{\mathrm{Var}[x']}}, \tag{5}$$

where, x' is the output vector of the first `Linear` layer and the input to the `Batch Normalization`, and $\mathrm{E}[x']$ and $\mathrm{Var}[x']$ are the expected value and variance of x', respectively. β and γ are weight vectors, which are learned during the training process via backpropagation to increase the accuracy [23]. Furthermore, given the output of the `Linear` layer x as input to the `Dropout` layer D_p, the output $\bar{x}$ is computed as:

$$\begin{cases} \bar{x} & = D_p(x) \\ \bar{x}_i & = \delta_i x_i, \end{cases} \tag{6}$$

where each δ_i follows the Bernoulli distribution of parameter p, i.e., δ is 1 with probability p, and 0 otherwise.

5 Experimental Setup

We evaluate our approach by comparing it with seven state-of-the-art fact-checking approaches. In the following, we first describe the datasets we rely upon. Then, we describe our experimental setting.

5.1 Datasets

Fact-Checking Datasets. In our experiments, we use two recent fact-checking datasets that are often used in the literature [20,46,47]: FactBench and BirthPlace/DeathPlace (BD). We use these datasets because they comprise entities of DBpedia, which is (i) large, and (ii) for which multiple pre-trained embedding models are available.

We only use a subset of the original FactBench dataset because it was created in 2014, and is based on DBpedia version 3.9 [20]. Ergo, some of the facts it contains are outdated. For example, (`:B.Obama`, `:presidentOf`, `:USA`) was a correct assertion when the benchmark was created but is currently incorrect (without the date information). We performed the following list of changes to obtain the benchmark used herein:

Table 2. Overview of all correct facts used in our experiments. The train and test sets (train/test) are from the 2 benchmark datasets FactBench and BD from [46].

	Property	|Sub|	|Obj|	Comment
FactBench	`:birthPlace`	75/75	67/65	Birth place (city)
	`:deathPlace`	75/75	54/48	Death place (city)
	`:award`	75/75	5/5	Winners of nobel prizes
	`:foundationPlace`	75/75	59/62	Foundation place and timeof software companies
	`:author`	75/75	75/73	Authors of science fiction books (one book/author)
	`:spouse`	74/74	74/74	Marriages between actors(after 2013/01/01)
	`:starring`	22/21	74/74	Actors starring in a movie
	`:subsidiary`	54/50	75/75	Company acquisitions
BD	`:birthPlace`	51/52	45/35	Birth place (city)
	`:deathPlace`	52/51	42/38	Death place (city)

Table 3. Overview of the number of wrong assertions in the different categories of the train and test set (train/test) from the 2 benchmark datasets FactBench and BD [46].

	Category	|Assertions|	Comment
FactBench	Domain	1000/985	Replacing s with another entity in the domain of p
	Range	999/985	Replacing o with another entity in the range of p
	DomainRange	990/989	Replacing s or o based on the domain and range of p, resp
	Property	1032/997	Replacing s and o based on p connectivity
	Random	1061/1031	Randomly replacing o or s with other entities
	Mix	1025/1024	Mixture of above categories
BD	`type-based`	206/206	Replacing s or o of different RDF type

– We removed the date category from wrong assertions.
– We removed all assertions with Freebase entities.
– We removed the : *team* predicate, because there were many false positives in this category of assertions, since nearly all players changed their teams meanwhile.

Our second evaluation dataset, dubbed BirthPlace/DeathPlace (short DB) [46], aims to overcome a limitation of the FactBench dataset. It only contains assertions pertaining to birth and death places. The dataset was created based on the observation that some fact-checking approaches only check if the subject and object have a relation to each other while the type of the relation, i.e., whether it matches the property of the given assertion, is not always taken into account. Hence, all subjects and objects within the BD dataset have a relation to each other. This ensures that an approach only performs well on this dataset if it takes the type of the relation in assertions into account.

An overview of the two benchmarking datasets used in our evaluation in terms of the number of true and false assertions in training and testing sets, predicates, and some details about the generation of those assertions are presented in Tables 2 and 3. Note that both datasets were designed to be class-balanced. Hence, we do not need to apply any method to alleviate potential class imbalances in the training and test data. However, we want to point out that the BD dataset provides less training examples than FactBench.

Reference Corpus. Our text-based component makes use of a reference corpus. We created this corpus by extracting the plain text snippets from all English Wikipedia articles and loading them into an Elasticsearch instance. We used the dump from March 7th, 2022. For the Elasticsearch[6] index, we used a cluster of 3 nodes with a combined storage of 1 TB and 32 GB RAM per node.

5.2 Evaluation Metric

As suggested in the literature, we use the area under the receiver operator characteristic curve (AUROC) to compare the fact-checking results [25,46,47]. We compute this score using the knowledge-base curation branch of the GERBIL framework [33,36].

5.3 Setup Details and Reproducibility

Within the sentence embedding module, we use a pre-trained SBert model.[7] Furthermore, we set $k = 3$ in the sentence selection module. The size of the sentence embedding vectors generated by SBert is 768, and the trustworthiness score values against each sentence vector, which leads to $|\varphi_{\aleph}| = (3 \times 768) + 3 = 2307$.

We use embeddings from 5 KG embedding models, where pre-trained DBpedia embeddings are available[8]. These models include: TransE [7], ConEx [12], QMult [11], ComplEx [49], and RDF2Vec [38]. For the FactBench dataset, we do not include experiments using RDF2Vec embeddings, because these embeddings were generated using a different version of DBpedia (i.e., 2015-10) and missing embeddings of multiple entities (i.e., 40/1800).[9] However, we included RDF2Vec embedding in the BD dataset comparison. Different KG embedding models provide embedding vectors with different lengths. For example, the TransE model used within our experiment maps each entity and each relation to a vector with 100 dimensions. This leads to a total size for $\varphi_{(s,p,o)}$ of 300.

We use the Binary Cross Entropy (BCE) as loss function for training our neural network component. We set the maximum number of epochs to 1000 with a batch size of 1/3 of the training data size. The training may have to be stopped earlier in case the neural network component starts to overfit. To this end, we calculate the validation

[6] https://www.elastic.co/.

[7] We ran experiments with all available pre-trained models (not shown in the paper due to space limitations) from the SBert homepage (https://www.sbert.net/docs/pretrained_models.html) and found that `nq-distilbert-base-v1` worked best for our approach.

[8] A large number of KG embedding algorithms [12,42,49] has been developed in recent years. However, while many of them show promising effectiveness, their scalability is often limited. For many of them, generating embedding models for the whole DBpedia is impractical (runtimes > 1 month). Hence, we only considered the approaches for which pre-trained DBpedia embeddings are available.

[9] Fair comparison could not be possible with missing entities, which constitute many assertions.

loss every 10th epoch and if this loss does not decrease for 50 epochs, the training is stopped.

All experiments are conducted on a machine with 32 CPU cores, 128 GB RAM and an NVIDIA GeForce RTX 3090. We provide hyperparameter optimization, training, and evaluation scripts on our project page for the sake of reproducibility.

5.4 Competing Approaches

We compare HybridFC in different configurations to FactCheck [46], COPAAL [47], and KV-Rule [25], which are the state-of-the-art approaches of the text-, path- and rule-based categories, respectively. We also compare our results to those four KG embedding-based approaches for which pre-trained DBpedia embedding models are available. We employ these models for fact checking by training the neural network module ϑ_2 of our approach based only on the output of the KG-based component. The output of this neural network module is then directly used as input for the final sigmoid function. We do not compare our results with results of the hybrid approaches mentioned in Sect. 3 because ExFaKT and Tracy mainly focus on generating human-comprehensible explanations and do not produce the veracity score, and FACTY focuses on calculating the veracity of assertions containing long-tail vertices (i.e., entities from less popular domains, for example, cheese varieties).

6 Results and Discussion

Tables 4 and 5 show the AUROC scores for the different hybrid and competing approaches on the FactBench train and test datasets, respectively. We can see that HybridFC performs best when it uses the TransE embedding model. This is not unexpected as TransE is one of the simplest embedding models that supports property composition: Given two properties p_1 and p_2, TransE entails that $\varphi(p_1 \circ p_2) \approx \varphi(p_1) + \varphi(p_2)$. With TransE as its embeddings model, HybridFC significantly outperforms all competing approaches on the test data.[10].

Note that FactCheck does not achieve the performance reported in [46] within our evaluation. This is due to (i) the use of a different English Wikipedia as reference corpus—Syed et al. showed that they achieve better results with the larger ClueWeb corpus—and (ii) the fact that we had to remove triples from the FactBench dataset.

The overall performance of COPAAL is better than the performance of FactCheck, ConEx, QMult and KV-Rule on the test set. However, we observe large performance differences with respect to the different properties. While COPAAL achieves the second best AUROC scores after HybridFC for 6 out of the 8 properties it struggles to achieve good results for `:award` and `:author`. These experimental results suggest that our approach makes good use of the diversity of the performance of the approaches it includes. In particular, it seems to rely on COPAAL's good performance on most of the properties while being able to complement COPAAL's predictions with that of other algorithms for properties on which COPAAL does not perform well.

[10] We use a Wilcoxon signed rank test with a significance threshold $\alpha = 0.05$.

Table 4. Area under the curve (AUROC) score on different categories of FactBench train sets. T stands for text-based approach, P for path-based approach, R for rule-based approaches, and KG-emb for KG-embedding-based approaches.

		Domain	Range	DomainRange	Mix	Random	Property	Avrg.
T	FactCheck [46]	0.69	0.69	0.68	0.65	0.68	0.57	0.66
P	COPAAL [47]	0.67	0.67	0.68	0.65	0.69	0.68	0.67
R	KV-Rule [25]	0.57	0.57	0.58	0.58	0.63	0.63	0.59
KG-emb	TransE [7]	0.67	0.61	0.78	0.66	0.92	0.97	0.76
	ConEx [12]	0.64	0.67	0.68	0.86	0.96	0.88	0.78
	ComplEx [49]	0.78	0.66	0.74	0.80	0.98	0.97	0.82
	QMult [11]	0.83	0.73	0.75	0.86	0.97	0.98	0.85
HybridFC	TransE	**0.94**	**0.94**	**0.96**	**0.90**	**0.99**	**0.99**	**0.95**
	ConEx	0.81	0.79	0.81	0.74	0.82	0.80	0.79
	ComplEx	0.94	0.94	0.94	0.86	0.95	0.97	0.93
	QMult	0.90	0.89	0.89	0.81	0.91	0.94	0.89

Table 5. Area under the curve (AUROC) score on different categories of FactBench test sets; the abbreviations are: T/Text-based approaches, P/Path-based approaches, R/Rule-based approaches, and KG-emb/KG embedding-based approaches.

		Domain	Range	DomainRange	Mix	Random	Property	Avrg.
T	FactCheck [46]	0.67	0.67	0.66	0.61	0.66	0.59	0.64
P	COPAAL [47]	0.67	0.68	0.68	0.65	0.69	0.69	0.68
R	KV-Rule [25]	0.57	0.57	0.57	0.58	0.61	0.62	0.59
KG-emb	TransE [7]	0.63	0.60	0.63	0.64	0.87	0.96	0.72
	ConEx [12]	0.50	0.50	0.50	0.52	0.60	0.60	0.54
	ComplEx [49]	0.58	0.58	0.52	0.62	0.86	0.95	0.69
	QMult [11]	0.57	0.62	0.55	0.69	0.84	0.93	0.70
HybridFC	TransE	**0.80**	**0.80**	**0.81**	**0.78**	**0.95**	**0.99**	**0.86**
	ConEx	0.77	0.78	0.79	0.71	0.80	0.70	0.75
	ComplEx	0.75	0.76	0.74	0.72	0.93	0.97	0.81
	QMult	0.69	0.73	0.71	0.69	0.91	0.94	0.77

On the BD dataset, KV-rule outperforms all other approaches on the test split (Table 6). COPAAL achieves the second best score, closely followed by the TransE-based HybridFC variant. The results confirm that the unsupervised fact-checking approaches COPAAL and KV-rule achieve good results for the `:birthPlace` and `:deathplace` properties. A closer look at the results reveals two main reasons for the lower result of the TransE-based HybridFC variant on the test dataset. First, FactCheck fails to extract pieces of evidence for most of the assertions. Second, FactCheck, the embedding-based approaches as well as the HybridFC variants are supervised approaches and suffer from the small size of the train split of the BD dataset. This is

Table 6. Area under the curve (AUROC) scores on the BD dataset; the abbreviations are: T stands for text-based approaches, P for path-based approaches, R for rule-based approaches, KG-emb for KG-embedding-based approaches.

	T	P	R	KG-emb					HybridFC				
	FactCheck [46]	COPAAL [47]	KV-Rule [25]	TransE [7]	ConEx [12]	ComplEx [49]	QMult [11]	RDF2Vec [38]	TransE	ConEx	ComplEx	QMult	RDF2Vec
Train	0.51	0.67	0.76	0.69	0.50	0.73	0.60	0.67	0.80	0.51	0.74	0.60	0.74
Test	0.49	0.70	**0.81**	0.54	0.50	0.54	0.55	0.62	0.69	0.50	0.57	0.58	0.68

Table 7. Results of our ablation study on the FactBench test set and BD dataset. D stands for Domain, R for Range, DR for DomainRange, Ran. for Random, Prop. for Property, and Avg. for average. TC stands for text-based component, PC for path-based component, EC for embedding-based component, and the symbol + indicates the combination of 2 components. Best performances are bold, second-best are underlined.

(a) FactBench test set

	D	R	DR	Mix	Ran.	Prop.	Avg.
TC	0.76	0.77	0.76	0.69	0.77	0.64	0.73
PC	0.68	0.69	0.69	0.65	0.70	0.69	0.68
EC	0.63	0.61	0.62	0.64	0.86	0.97	0.72
TC+EC	0.76	0.78	0.76	0.74	0.92	0.98	0.82
TC+PC	0.77	0.77	0.77	0.7	0.79	0.67	0.74
PC+EC	0.71	0.7	0.69	0.72	0.89	0.97	0.78
HybridFC	**0.80**	**0.80**	**0.81**	**0.78**	**0.95**	**0.99**	**0.86**

(b) BD dataset

	Train	Test
TC	0.59	0.56
PC	0.67	**0.70**
EC	0.69	0.56
TC+EC	0.79	0.65
TC+PC	0.67	0.64
PC+EC	0.74	0.66
HybridFC	**0.80**	0.69

confirmed by our observation that the neural network component tends to overfit during the training phase.

7 Ablation Study

Our previous experiments suggest that HybridFC performs best in combination with TransE. Hence, we use it as default setting throughout the rest of the paper and overload HybridFC to mean HybridFC with TransE embeddings. To evaluate the contribution of the different components of HybridFC to its performance, we rerun our evaluation for each component (i.e., text-based (TC), path-based (PC), and embedding-based (EC)) individually and as pairwise combination of different components (TC+PC, TC+EC, PC+EC). The results for the FactBench test and the BD datasets are shown in Tables 7a and 7b.[11] The results suggest that the individual path-based and embedding-based components achieve results similar to those of COPAAL and TransE, respectively. Our text-based component achieves better results than FactCheck. On the FactBench test datasets, the combination of two components leads to better results than the single components. Similarly, HybridFC, i.e., the combination of all three components, leads to significantly better results than all pairwise combinations, where significance is measured using a Wilcoxon signed rank test with a p-value threshold of 0.05. Here, our null hypothesis is that the performances of the approaches compared are sampled from the same distribution. For the BD dataset, the pairwise combinations of components suffer from the same overfitting problem as HybridFC. Overall, our results in Table 7a suggest that our text component commonly achieves the highest average performance on datasets that provide enough training data. The text component is best supplemented by the embedding-based component. HybridFC outperforming all combinations of two components on FactBench suggests that in cases in which HybridFC is trained with enough training data, each of the three components contributes to the better overall performance of HybridFC.

8 Conclusion

In this paper, we propose HybridFC–a hybrid fact-checking approach for KGs. HybridFC aims to alleviate the problem of manual feature engineering in text-based approaches, cases in which paths between subjects and objects are unavailable to path-based approaches, and the poor performance of pure KG-embedding-based approaches by combining these three categories of approaches. We compare HybridFC to the state of the art in fact checking for KGs. Our experiments show that our hybrid approach is able to outperform competing approaches in the majority of cases. As future work, we will exploit the modularity of HybridFC by integrating rule-based approaches. We also plan to explore other possibilities to select the best evidence sentences.

Supplemental Material Statement

- The source code of HybridFC, the scripts to recreate the full experimental setup, and the required libraries can be found on GitHub.[12]
- Datasets used in this paper and the output generated by text-based and path-based approaches on these datasets are available at Zenodo [3].
- Pre-trained embeddings for these datasets are also available at Zenodo [4].

Acknowledgments. The work has been supported by the EU H2020 Marie Skłodowska-Curie project KnowGraphs (no. 860801), the German Federal Ministry for Economic Affairs and Climate Action (BMWK) funded project RAKI (no. 01MD19012B), and the German Federal Ministry of Education and Research (BMBF) funded EuroStars projects 3DFed (no. 01QE2114B) and FROCKG (no. 01QE19418). We are also grateful to Daniel Vollmers and Caglar Demir for the valuable discussion on earlier drafts.

[11] Due to space limitation we exclude the results of FactBench train set. These results are available on our GitHub page.

[12] Source code: https://github.com/dice-group/HybridFC.

References

1. Athreya, R.G., Ngonga Ngomo, A.C., Usbeck, R.: Enhancing community interactions with data-driven chatbots-the dbpedia chatbot. In: Companion Proceedings of the The Web Conference 2018, pp. 143–146. WWW 2018, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE (2018). https://doi.org/10.1145/3184558.3186964
2. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: a nucleus for a web of open data. In: Aberer, K., et al. (eds.) ASWC/ISWC -2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76298-0_52
3. Authors, A.: Mypublications dataset. https://doi.org/10.5281/zenodo.6523389
4. Authors, A.: Pre-trained embeddings for fact-checking datasets. https://doi.org/10.5281/zenodo.6523438
5. Bengio, Y., Courville, A., Vincent, P.: Representation learning: a review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell. **35**(8), 1798–1828 (2013)
6. Boland, K., Fafalios, P., Tchechmedjiev, A., Dietze, S., Todorov, K.: Beyond facts - a survey and conceptualisation of claims in online discourse analysis, March 2021. https://hal.mines-ales.fr/hal-03185097, working paper or preprint
7. Bordes, A., Usunier, N., Garcia-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2. NIPS 2013, pp. 2787–2795, Curran Associates Inc., Red Hook, NY, USA (2013)
8. Chen, Y., Goldberg, S., Wang, D.Z., Johri, S.S.: Ontological pathfinding: mining first-order knowledge from large knowledge bases. In: Proceedings of the 2016 International Conference on Management of Data. SIGMOD 2016, New York, NY, USA, pp. 835–846. Association for Computing Machinery (2016). https://doi.org/10.1145/2882903.2882954
9. Ciampaglia, G.L., Shiralkar, P., Rocha, L.M., Bollen, J., Menczer, F., Flammini, A.: Computational fact checking from knowledge networks. PLoS ONE **10**(6), 1–13 (2015). https://doi.org/10.1371/journal.pone.0128193
10. Dai, Y., Wang, S., Xiong, N.N., Guo, W.: A survey on knowledge graph embedding: approaches, applications and benchmarks. Electronics **9**(5) (2020). https://doi.org/10.3390/electronics9050750
11. Demir, C., Moussallem, D., Heindorf, S., Ngomo, A.C.N.: Convolutional hypercomplex embeddings for link prediction. In: Asian Conference on Machine Learning, pp. 656–671. PMLR (2021)
12. Demir, C., Ngomo, A.-C.N.: Convolutional complex knowledge graph embeddings. In: Verborgh, R., et al. (eds.) ESWC 2021. LNCS, vol. 12731, pp. 409–424. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77385-4_24
13. Dong, X.L., et al.: Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In: The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2014, New York, NY, USA, 24–27 August, pp. 601–610, 2014 (2014). http://www.cs.cmu.edu/nlao/publication/2014.kdd.pdf, evgeniy Gabrilovich Wilko Horn Ni Lao Kevin Murphy Thomas Strohmann Shaohua Sun Wei Zhang Geremy Heitz
14. Gad-Elrab, M.H., Stepanova, D., Urbani, J., Weikum, G.: Exfakt: a framework for explaining facts over knowledge graphs and text. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining. WSDM 2019, New York, NY, USA, pp. 87–95. Association for Computing Machinery (2019). https://doi.org/10.1145/3289600.3290996
15. Gad-Elrab, M.H., Stepanova, D., Urbani, J., Weikum, G.: Tracy: tracing facts over knowledge graphs and text. In: The World Wide Web Conference. WWW 2019, pp. 3516–3520, New York, NY, USA. Association for Computing Machinery (2019). https://doi.org/10.1145/3308558.3314126

16. Galárraga, L., Teflioudi, C., Hose, K., Suchanek, F.M.: Fast rule mining in ontological knowledge bases with AMIE+. VLDB J. **24**(6), 707–730 (2015). https://doi.org/10.1007/s00778-015-0394-1
17. Galárraga, L.A., Teflioudi, C., Hose, K., Suchanek, F.: Amie: association rule mining under incomplete evidence in ontological knowledge bases. In: Proceedings of the 22nd International Conference on World Wide Web. WWW 2013, pp. 413–422, New York, NY, USA. Association for Computing Machinery (2013). https://doi.org/10.1145/2488388.2488425
18. Gardner, M., Mitchell, T.: Efficient and expressive knowledge base completion using subgraph feature extraction. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1488–1498 (2015)
19. Gardner, M., Talukdar, P., Krishnamurthy, J., Mitchell, T.: Incorporating vector space similarity in random walk inference over knowledge bases. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 397–406, Doha, Qatar. Association for Computational Linguistics, October 2014. https://doi.org/10.3115/v1/D14-1044
20. Gerber, D., et al.: Defacto-temporal and multilingual deep fact validation. Web Semant. **35**(P2), 85–101 (2015). https://doi.org/10.1016/j.websem.2015.08.001
21. Huang, J., et al.: Trustworthy knowledge graph completion based on multi-sourced noisy data. In: Laforest, F., et al. (eds.) WWW 2022: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25–29, 2022, pp. 956–965. ACM (2022). https://doi.org/10.1145/3485447.3511938
22. Huynh, V.P., Papotti, P.: Towards a benchmark for fact checking with knowledge bases. In: Companion Proceedings of the The Web Conference 2018, pp. 1595–1598. WWW 2018, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee (2018). https://doi.org/10.1145/3184558.3191616
23. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37. ICML 2015, pp. 448–456. JMLR.org (2015)
24. Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: 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), pp. 687–696. Association for Computational Linguistics, Beijing, China, July 2015. https://doi.org/10.3115/v1/P15-1067
25. Kim, J., Choi, K.s.: Unsupervised fact checking by counter-weighted positive and negative evidential paths in a knowledge graph. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 1677–1686. International Committee on Computational Linguistics, Barcelona, Spain (Online), December 2020. https://doi.org/10.18653/v1/2020.coling-main.147
26. Kotonya, N., Toni, F.: Explainable automated fact-checking for public health claims. arXiv preprint arXiv:2010.09926 (2020)
27. Lajus, J., Galárraga, L., Suchanek, F.: Fast and exact rule mining with AMIE 3. In: Harth, A., et al. (eds.) The Semantic Web, pp. 36–52. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49461-2_3
28. Li, F., Dong, X.L., Langen, A., Li, Y.: Knowledge verification for long-tail verticals. Proc. VLDB Endow. **10**(11), 1370–1381 (2017). https://doi.org/10.14778/3137628.3137646
29. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 29 (2015)
30. Malyshev, S., Krötzsch, M., González, L., Gonsior, J., Bielefeldt, A.: Getting the most out of Wikidata: semantic technology usage in Wikipedia's knowledge graph. In: Vrandečić, D.,

et al. (eds.) The Semantic Web - ISWC 2018, pp. 376–394. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00668-6_23

31. Malyshev, S., Krötzsch, M., González, L., Gonsior, J., Bielefeldt, A.: Getting the most out of wikidata: semantic technology usage in Wikipedia's knowledge graph. In: Vrandečić, D., et al. (eds.) ISWC 2018. LNCS, vol. 11137, pp. 376–394. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00668-6_23
32. Nakamura, S., et al.: Trustworthiness analysis of web search results. In: Kovács, L., Fuhr, N., Meghini, C. (eds.) ECDL 2007. LNCS, vol. 4675, pp. 38–49. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74851-9_4
33. Ngonga Ngomo, A.C., Röder, M., Syed, Z.H.: Semantic web challenge 2019. Website (2019). https://github.com/dice-group/semantic-web-challenge.github.io/. Accessed 30 March 2022
34. Ortona, S., Meduri, V.V., Papotti, P.: Rudik: rule discovery in knowledge bases. Proc. VLDB Endow. **11**(12), 1946–1949 (2018). https://doi.org/10.14778/3229863.3236231
35. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical Report 1999–66, Stanford InfoLab, November 1999. http://ilpubs.stanford.edu:8090/422/, previous number = SIDL-WP-1999-0120
36. Paulheim, H., Ngonga Ngomo, A.C., Bennett, D.: Semantic web challenge 2018. Website (2018). http://iswc2018.semanticweb.org/semantic-web-challenge-2018/index.html. Accessed 30 March 2022
37. Reimers, N., Gurevych, I.: Sentence-BERT: sentence embeddings using Siamese BERT-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 3982–3992. Association for Computational Linguistics, Hong Kong, China, November 2019. https://doi.org/10.18653/v1/D19-1410
38. Ristoski, P., Paulheim, H.: RDF2Vec: RDF graph embeddings for data mining. In: Groth, P., et al. (eds.) ISWC 2016. LNCS, vol. 9981, pp. 498–514. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46523-4_30
39. Rula, A., et al.: Tisco: temporal scoping of facts. Web Semant. **54**(C), 72–86 (2019). https://doi.org/10.1016/j.websem.2018.09.002
40. Shi, B., Weninger, T.: Discriminative predicate path mining for fact checking in knowledge graphs. Know.-Based Syst. 104(C), 123–133 (2016). https://doi.org/10.1016/j.knosys.2016.04.015
41. Shiralkar, P., Flammini, A., Menczer, F., Ciampaglia, G.L.: Finding streams in knowledge graphs to support fact checking. In: 2017 IEEE International Conference on Data Mining (ICDM), pp. 859–864 (2017). https://doi.org/10.1109/ICDM.2017.105
42. da Silva, A.A.M., Röder, M., Ngomo, A.-C.N.: Using compositional embeddings for fact checking. In: Hotho, A., et al. (eds.) ISWC 2021. LNCS, vol. 12922, pp. 270–286. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-88361-4_16
43. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Proceedings of the 16th International Conference on World Wide Web, pp. 697–706. ACM (2007)
44. Sultana, T., Lee, Y.: Efficient rule mining and compression for RDF style kb based on horn rules. J. Supercomput. (2022). https://doi.org/10.1007/s11227-022-04519-y
45. Sun, Y., Barber, R., Gupta, M., Aggarwal, C.C., Han, J.: Co-author relationship prediction in heterogeneous bibliographic networks. In: 2011 International Conference on Advances in Social Networks Analysis and Mining, pp. 121–128 (2011). https://doi.org/10.1109/ASONAM.2011.112
46. Syed, Z.H., Röder, M., Ngonga Ngomo, A.C.: Factcheck: validating RDF triples using textual evidence. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management. CIKM 2018, New York, NY, USA, pp. 1599–1602. Association for Computing Machinery (2018). https://doi.org/10.1145/3269206.3269308

47. Syed, Z.H., Srivastava, N., Röder, M., Ngomo, A.C.N.: Copaal - an interface for explaining facts using corroborative paths. In: ISWC Satellites (2019)
48. Syed, Z.H., Srivastava, N., Röder, M., Ngomo, A.N.: COPAAL - an interface for explaining facts using corroborative paths. In: Suárez-Figueroa, M.C., Cheng, G., Gentile, A.L., Guéret, C., Keet, C.M., Bernstein, A. (eds.) Proceedings of the ISWC 2019 Satellite Tracks (Posters & Demonstrations, Industry, and Outrageous Ideas) co-located with 18th International Semantic Web Conference (ISWC 2019), Auckland, New Zealand, October 26–30, 2019. CEUR Workshop Proceedings, vol. 2456, pp. 201–204. CEUR-WS.org (2019). http://ceur-ws.org/Vol-2456/paper52.pdf
49. Trouillon, T., Welbl, J., Riedel, S., Gaussier, E., Bouchard, G.: Complex embeddings for simple link prediction. In: International Conference on Machine Learning, pp. 2071–2080 (2016)
50. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: a survey of approaches and applications. IEEE Trans. Knowl. Data Eng. **29**(12), 2724–2743 (2017). https://doi.org/10.1109/TKDE.2017.2754499
51. Watt, N., du Plessis, M.C.: Dropout algorithms for recurrent neural networks. In: Proceedings of the Annual Conference of the South African Institute of Computer Scientists and Information Technologists, New York, NY, USA, pp. 72–78. SAICSIT 2018, Association for Computing Machinery (2018). https://doi.org/10.1145/3278681.3278691