TEMPORALFC: A Temporal Fact Checking approach over Knowledge Graphs

Umair Qudus(✉) [1], Michael Röder [1], Sabrina Kirrane [2], and Axel-Cyrille Ngonga Ngomo [1]

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

[2] Institute for Information Systems and New Media,
Vienna University of Economics and Business, Vienna, Austria
sabrina.kirrane@wu.ac.at

Abstract. Verifying assertions is an essential part of creating and maintaining knowledge graphs. Most often, this task cannot be carried out manually due to the sheer size of modern knowledge graphs. Hence, automatic fact-checking approaches have been proposed over the last decade. These approaches aim to compute automatically whether a given assertion is correct or incorrect. However, most fact-checking approaches are binary classifiers that fail to consider the volatility of some assertions, i.e., the fact that such assertions are only valid at certain times or for specific time intervals. Moreover, the few approaches able to predict when an assertion was valid (i.e., time-point prediction approaches) rely on manual feature engineering. This paper presents TEMPORALFC, a temporal fact-checking approach that uses multiple sources of background knowledge to assess the veracity and temporal validity of a given assertion. We evaluate TEMPORALFC on two datasets and compare it to the state of the art in fact-checking and time-point prediction. Our results suggest that TEMPORALFC outperforms the state of the art on the fact-checking task by 0.13 to 0.15 in terms of Area Under the Receiver Operating Characteristic curve and on the time-point prediction task by 0.25 to 0.27 in terms of Mean Reciprocal Rank. Our code is open-source and can be found at https://github.com/dice-group/TemporalFC.

Keywords: temporal fact checking · ensemble learning · transfer learning · time-point prediction · temporal knowledge graphs.

1 Introduction

The transition from an industrial civilization to an information and knowledge society during the last few decades has been fast-paced [12]. The adoption of the World Wide Web is largely to thank for this transformation. A similar uptake of knowledge graphs for the creation and management of knowledge has occurred in many communities around the world [19]. This uptake most certainly holds in the semantic web community, where knowledge is commonly represented in the form of RDF knowledge graphs (KGs). The

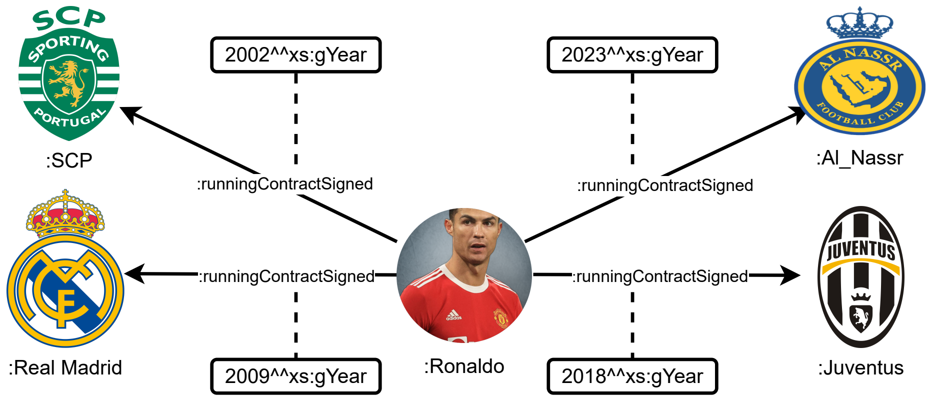

Fig. 1: A temporal knowledge graph excerpt from a large knowledge graph. The dotted line shows the time point for a given fact (in granularity of a year). Filled black lines are edges (aka predicates) with labels, and the rest are nodes of different RDF classes, which include Person, City, County, Award, and University.

Linked Open Data Stats[3], which already holds over 9,000 KGs with more than 149 billion assertions and 3 billion entities, further supports the growing adoption of the Resource Description Framework (RDF) at Web scale [13]. WikiData [34], DBpedia [2], Knowledge Vault [11], and YAGO [49] are examples of large-scale KGs that include billions of assertions and describe millions of entities. They are used as background information in an increasing number of applications, such as in-flight entertainment [35], autonomous chatbots [1], and healthcare [27]. However, current KGs may not be fully correct. for instance, the literature assumes that roughly 20% of DBpedia's claims are erroneous [19,44]. To encourage the further adoption of KGs on a large scale on the Web, approaches that can automatically forecast the truthfulness of the assertions included in KGs must be developed. Such methods are what we refer to as fact-checking approaches.

There are several fact-checking approaches designed to verify assertions in KGs and compute their veracity scores [51,19,52,50,23]. However, assertions can be volatile and the majority of existing approaches do not take any temporal aspects into account. For example, Figure 1, which we use as a running example in this paper, shows that the assertion (`:Ronaldo`, `:runningContractSigned`, `:SCP`) is not accurate without information about the year since `:Ronaldo` also signed contracts with other teams at later points in time[4]. Ergo, temporal information is critical for validating volatile assertions. However, very little attention has been given to the temporal aspect of KGs when dealing with the task at hand. Temporal-DeFacto [19] is at the time of writing the only temporal fact-checking method that looks at this aspect of KGs along with their verifica-

[3] `http://lodstats.aksw.org/`

[4] From here on, we work with IRIs. The prefixes for these IRIs that we use are "xs" and ":". The xmlns schema is identified by the URI-Reference `http://www.w3.org/2001/XMLSchema/#` and is associated with the prefix 'xs'. Furthermore, we use ":" prefix for literals.

tion. However, Temporal-DeFacto relies on tedious manual feature engineering [19,51], which has been shown to be sub-optimal w.r.t. their prediction performance by representation learning approaches [4]. The approaches T-TRANSE [30], T-COMPLEX [6] and T-DYHE [36] from the knowledge base completion domain concentrate on time-point prediction as well. These approaches focus on the time prediction task and encounter limitations with respect to their Mean Reciprocal Rank (MRR) scores as well as their scalability. Furthermore, they do not consider the fact-checking aspect [36].

We alleviate the limitations of the aforementioned approaches by proposing a neural network-based approach that utilizes transfer learning (i.e., it uses pre-trained embeddings created from a Temporal Knowledge Graph TKG) for fact-checking and time-point prediction tasks. Since temporal information is also critical, our system predicts the year in which an assertion was true, along with its veracity score. For example, if an assertion (`:Ronaldo`, `:runningContractSigned`, `:SCP`) is given as input, our system not only validates the statement, it also predicts the year in which the assertion was true.

The main contributions of our work are as follows:

- We employ transfer learning to repurpose pre-trained TKG embeddings for the fact-checking and time-point prediction tasks.
- We present an open-source neural network-based approach for detecting the temporal scope of assertions.
- We evaluate our approach on two datasets—DBpedia124K and Yago3K—and compare it to the state of the art of time-point prediction and temporal fact-checking tasks. Our approach outperforms other approaches in the time-point prediction task by 0.25 and 0.27 MRR and temporal fact-checking task by 0.13 to 0.15 in terms of Area Under the Receiver Operating Characteristic curve (AUROC).

The rest of this paper is organized as follows. The notations necessary to comprehend the remainder of the paper are introduced in Section 2. In Section 3, we provide an overview of the related work. We present our proposed approach in Section 4. After that, Section 5 describes the experimental setup. The results are discussed in Section 6. In Section 7, we conclude and discuss potential future work.

2 Preliminaries

Fact checking and related terms have a variety of definitions that come from different fields, such as journalism [25,32,26], natural language processing [39], and KGs [53,29,45]. We adopt the definition of fact-checking for KGs provided in [51] as follows:

Definition 1 (Fact Checking). *Fact checking implies calculating the likelihood that an assertion is true or false in the presence of a reference KG $\mathcal{G}$, and/or a reference corpus [51].*

We utilize RDF TKGs throughout the entirety of this work.

Definition 2 (Temporal Knowledge Graph (TKG)). *A TKG $\mathcal{TG}$ is a collection of RDF quadruples $\mathcal{TG} \subseteq (\mathbb{E} \cup \mathbb{B}) \times \mathbb{P} \times (\mathbb{E} \cup \mathbb{B} \cup \mathbb{L}) \times \mathbb{T}$, where each quadruple $(s, p, o, t) \in \mathcal{TG}$*

Table 1: Scoring functions of different embedding-based approaches used in this paper. $*$ stands for the Dihedron multiplication, $\otimes$ stands for the quaternion multiplication, $\mathbb{R}$ for the space of real numbers, $\mathbb{H}$ *for the space of quaternions,* $\mathbb{D}$ *for the space of Dihedrons*, $\mathbb{C}$ for the complex numbers, Re for the real part of a complex number, $<>$ for componentwise multi-linear dot product e.g., $< a, b, c > := \sum_k a_k b_k c_k$, conv for the convolution operator, $\overline{\varphi(\circ)}$ for the complex conjugate of $\varphi(\circ)$, q is the length of embedding vectors, and $\|\cdot\|_2$ for the L2 norm.

	Approach	Scoring function	Vector space
TKGE	T-TransE	$-\|(\varphi(s)+\varphi(p)+\varphi(t))-\varphi(o)\|_2$	$\varphi(s), \varphi(p), \varphi(o), \varphi(t) \in \mathbb{R}^q$
	T-ComplEx	$\mathrm{Re}\left(< \varphi(s), \varphi(p), \varphi(t), \overline{\varphi(o)} >\right)$	$\varphi(s), \varphi(p), \varphi(t), \varphi(o) \in \mathbb{C}^q$
	T-DyHE	$-\|\varphi(s) * \varphi(p)_{1,2} + \varphi(t) - \varphi(o)\|_2$	$\varphi(s), \varphi(p)_{1,2}, \varphi(t), \varphi(o) \in \mathbb{D}^q$
KGE	TransE	$\|(\varphi(s)+\varphi(p))-\varphi(o)\|_2$	$\varphi(s), \varphi(p), \varphi(o) \in \mathbb{R}^q$
	ComplEx	$\mathrm{Re}\left(< \varphi(s), \varphi(p), \overline{\varphi(o)} >\right)$	$\varphi(s), \varphi(p), \varphi(o) \in \mathbb{C}^q$
	QMult	$\varphi(s) \otimes \varphi(p) \cdot \varphi(o)$	$\varphi(s), \varphi(p), \varphi(o) \in \mathbb{H}^q$
	ConEx	$\mathrm{Re}(\langle \mathrm{conv}(\varphi(\mathrm{s}), \varphi(\mathrm{p})), \varphi(\mathrm{s}), \varphi(\mathrm{p}), \overline{\varphi(\mathrm{o})} \rangle)$	$\varphi(s), \varphi(p), \varphi(o) \in \mathbb{C}^q$

consists of a subject (s), a predicate (p), an object (o), and a time point (t). $\mathbb{E}$ *is the set of all RDF resource IRIs (Internationalized Resource Identifier),* $\mathbb{P} \subseteq \mathbb{E}$ *is the set of all RDF predicates,* $\mathbb{L}$ *is the set of all literals,* $\mathbb{B}$ *is the set of all blank nodes, and* $\mathbb{T}$ *is the set of all time points [52,36].*

In this study, we treat each year as a single point in time. The time-point prediction is defined as follows:

Definition 3 (Time-point prediction). *Given an assertion* (s, p, o)*, the task of time-point prediction is to predict the time-point* t *to form a correct quadruple* (s, p, o, t)*, where* t *is a specific point in time that represents the occurrence of a predicate* p *with respect to* s *and* o *[36].*

In our running example, a time-point prediction algorithm should predict 2002 for the assertion (`:Ronaldo`, `:runningContractSigned`, `:SCP`).

Definition 4 (Temporal Knowledge Graph Embeddings (TKGE)). *A TKGE embedding function* φ *maps a* $\mathcal{TG}$ *to a continuous vector space. Given a quadruple* (s, p, o, t)*,* $\varphi(s)$*,* $\varphi(p)$*,* $\varphi(o)$*, and* $\varphi(t)$ *stand for the embedding of the subject, predicate, object, and time point, respectively [28].*

Different knowledge graph embedding (KGE) and temporal knowledge graph embedding -based approaches use different scoring functions to compute embeddings [55]. The approaches considered in this paper are shown in Table 1.

3 Related Work

The research covered in this paper relates to two key areas of study: fact checking and time-point prediction of assertions in KGs. The most recent methods in each area are briefly described below, along with their limitations.

3.1 Fact checking

The goal of fact checking is to determine which subset of a given set of assertions from a KG may be trusted [40]. Fact-checking approaches can broadly be divided into three categories: those that utilize unstructured textual sources [51,19], those that utilize structured information sources [52,50,23], and those that are hybrid and use both [42].

In the first category, approaches validate a given assertion by searching evidence in a reference text corpus. There are two examples of this category: FactCheck [51] and DeFacto [19]. Both approaches are based on RDF verbalization techniques to find textual excerpts that can be used as evidence for the stated assertion. Both approaches compute a vector representation of the texts they retrieve as evidence based on a set of manually created features.

In the second category, there are three sub-categories of approaches: 1. path-based, 2. rule-based, and 3. embedding-based. By automatically computing short paths from the subject of the assertion to its object inside the reference KG, path-based approaches seek to validate the input assertion. The input assertion is then scored using these paths. Most path-based approaches, like COPAAL [52], Knowledge stream [46], PRA [18], SFE [17], and KG-Miner [45], filter out meaningful paths using RDF semantics (e.g., class subsumption hierarchy, domain and range information). However, the T-Box of several KGs provides a limited number of RDFS statements. Furthermore, no short paths may be found within the reference KG, although the assertion is correct [52]. In these situations, path-based approaches fall short of accurately predicting whether the provided assertion is true. Rule-based approaches such as KV-Rule [23], AMIE [15,14,29], OP [7], and RuDiK [38] extract association rules from KGs to perform the fact-checking task. These approaches are constrained by the knowledge found in the KG, and mining rules from huge KGs can be a long and tedious process (e.g., OP takes ≥ 45 hours on DBpedia [29]). Embedding-based approaches express the input KG in a continuous high-dimensional vector space via a mapping function [21,5,31,10,54,47]. For example, Esther [47] computes likely paths between resources using compositional embeddings. By developing a KG embedding model and learning a scoring function, the veracity of these statements is computed. In general, the information included in the continuous representation of the KG is the fundamental constraint on embedding-based techniques. Ergo, when used with large-scale KGs, these approaches have limitations in terms of both their scalability and accuracy in fact-checking scenarios.

The third category is more pertinent to the work presented herein. To the best of our knowledge, the only state-of-the-art hybrid approach that takes full advantage of the variety of available fact-checking approach categories in an ensemble learning environment is called HybridFC [42]. By integrating the aforementioned categories of approaches, HybridFC seeks to address the issues of: 1. manual feature engineering in text-based approaches, 2. circumstances when paths between subjects and objects are not available to path-based approaches, and 3. the poor performance of pure KG-embedding-based approaches. However, HybridFC does not use time information along with the assertions in a TKG. In comparison, TEMPORALFC overcomes this limitation.

3.2 Time-point prediction

Knowledge Graphs (KGs) with an added temporal component are the focus of Temporal Knowledge Graph Embedding (TKGE) models. Quadruples are created from a triple-based representation. The majority of the early TKGE models were constructed on top of KGEs that already existed. One of the first TKGEs to project the subject, predicate, and object embeddings to a time space is the HyTE [8] model. HyTE uses TRANSE on the projected embeddings for the final scoring of the newly predicted facts. T-TRANSE [30] and TA-TRANSE [16] are two further TKGEs that have been proposed as expansions of TRANSE. The ConT model, which is an extension of the Tucker [3] KGE, is the other cutting-edge approach among TKGEs. For the encoding of TKGs, a number of adaptations to DistMult [58] have also been proposed, including TDistMult [33] and TA-DistMult [16]. Recurrent neural networks (RNNs) are the foundation of these models, and they capture the entity embeddings for the subject and object entities. Another RNN-based TKGE called RE-NET uses unique patterns from historical data between entities to capture pair-wise knowledge in the form of (subject, predicate) or (object, predicate) pairs [22]. The main difficulty with these models is that they carry over the flaws of the base models upon which they are built. For instance, the TKGEs that were constructed on top of TRANSE have problems encoding relational patterns. Recently, the TeRo model [57] was developed to address these issues with the pre-existing TKGEs regarding the inference of relational patterns. TeRo partially overcomes some of the limitations of other models; however, it does not focus on time-point prediction in TKGEs. Instead, it uses the time dimension solely for a better relation prediction. The T-COMPLEX model [6] is the temporal iteration of the COMPLEX-N3 model, which achieves better results in the relation prediction task compared with the previous approaches. For learning and predicting time-points, the DYHE embedding model uses dihedron algebra. Dihedron algebra is a rich 4D algebra of hyper-complex spaces. To the best of our knowledge, DYHE is the only approach for which the authors report the results of the time-point prediction task.

Our proposed approach performs both tasks—fact checking and time-point prediction—for a given assertion. To the best of our knowledge, Temporal-DeFacto [19] is the only state-of-the-art approach that covers both tasks as well.

4 Methodology

We propose TEMPORALFC, an approach that addresses the fact-checking and time-point prediction tasks. It takes the quadruple of a TKG as input and comprises three components as depicted in Figure 2. First, a pre-trained `TKGE model` is used to generate TKGE vectors of the input quadruple. Second, a `fact-checking` component classifies the (`s`, `p`, `o`) part of the quadruple as true or false. *We designed this fact-checking component as an extension of HybridFC and,* in contrast to the related work, our component takes all 4 TKGE vectors from the first component into account. If an assertion is classified as true, the (`s`, `p`, `o`) TKGE vectors of the input quadruple are used as input for the third `time-point prediction` component. The `time-point prediction` component comprises a neural network, which predicts the year in which the given assertion should be true. *Our ensemble of the two latter components is in line* *with the task definition of Temporal-DeFacto [19], i.e., our approach checks whether the assertion is true at any point in time and then performs the time-point prediction task for the given assertion.* We describe all these components in more detail in the following.

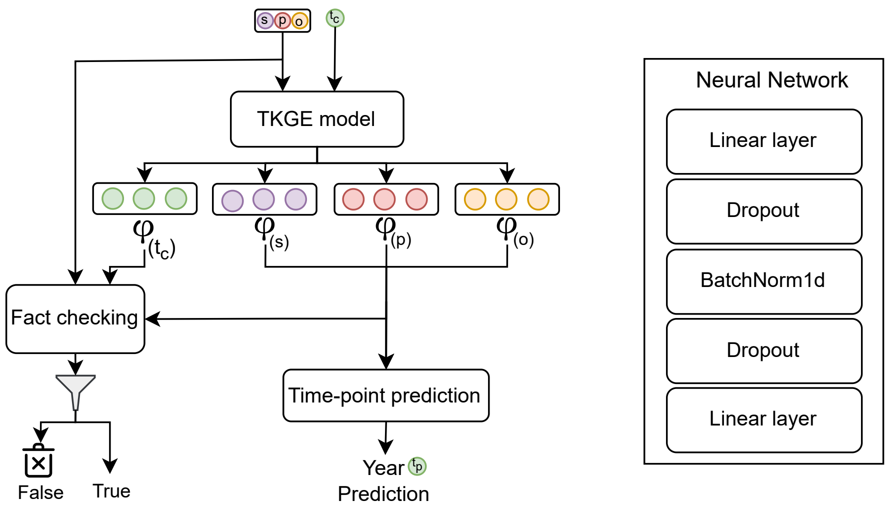

Fig. 2: Left: Overview of the architecture of TEMPORALFC. Right: A multi-layer perceptron module (ϑ_i) that is used in the fact-checking and the time-point prediction components of TEMPORALFC.

4.1 TKGE model

We use pre-trained temporal embedding vectors, which are generated from a given TKGE model, in our approach. Given a quadruple $(\mathtt{s}, \mathtt{p}, \mathtt{o}, \mathtt{t})$, $\varphi(\mathtt{s})$, $\varphi(\mathtt{o})$, $\varphi(\mathtt{p})$, and $\varphi(\mathtt{t})$ stand for the embedding of the subject, predicate, object, and time point, respectively. Initially, we form an embedding vector for a given quadruple $(\mathtt{s}, \mathtt{p}, \mathtt{o}, \mathtt{t})$ by concatenating the embedding of its first three elements and defining the embedding mapping function $\varphi(\mathtt{s,p,o})$ for assertions $(\mathtt{s}, \mathtt{p}, \mathtt{o})$ as follows [42]:

$$\varphi(s, p, o) = \varphi(s) \oplus \varphi(p) \oplus \varphi(o)\,, \tag{1}$$

where $\oplus$ stands for the concatenation of vectors. $\varphi(\mathtt{s,p,o})$ is used in both of the following components, while the time embedding vector $\varphi(\mathtt{t})$ is only used in the `fact-checking` component.

From our running example, given (`:Ronaldo`, `:runningContractSigned`, `:SCP`, `2002`) as input quadruple, we first gather the embedding of each component using a pre-trained TKGE model. In the next step, we transform the embeddings of φ(`:Ronaldo`), φ(`:runningContractSigned`), and φ(`:SCP`) into a concatenated embedding vector φ(`:Ronaldo,:runningContractSigned,:SCP`), which is the input to both the `time-point-prediction` and the `fact-checking` components. The `fact-checking` component utilizes $\varphi(\texttt{2002})$ as an additional input.

4.2 Fact checking

Our fact-checking component addresses the fact-checking task according to Definition 1. It relies on a modified architecture of the HybridFC approach [42], which is the current state-of-the-art algorithm for the fact-checking task. HybridFC takes a triple $(\texttt{s}, \texttt{p}, \texttt{o})$ as input and retrieves three different types of evidence—embeddings-, text-, and $\mathcal{G}$-based evidence. While the embedding-based evidence comprises a representation of the input assertion in the knowledge graph, the textual evidence is gathered from a reference corpus. The search for textual evidence transforms the given assertion into a search query and extracts those pieces of text from the retrieved documents that contain all terms of the search query. The pieces of text are sorted in descending order based on the PageRank of their respective document. Then, the top-k pieces of text are selected and transformed into embedding vectors using a pre-trained sentence embedding model. The result of these evidence retrieval steps are:

1. $\varphi(\texttt{s,p,o})$,
2. a vector $\varphi_{\aleph}$ comprising the concatenated embedding vectors of the top-k evidence sentences, and
3. the veracity score ζ of the input assertion from a path-based fact-checking algorithm.

From our running example, $\varphi(\texttt{:Ronaldo,:runningContractSigned,:SCP})$ is the embedding-based evidence retrieval output. An example for a textual evidence could be "Ronaldo began his senior career with Sporting CP (SCP)" retrieved from the reference corpus as one of the outputs for the given triple and transformed into an embedding vector. This vector is concatenated with the embedding vectors of other pieces of textual evidence to form $\varphi_{\aleph}$. The $\mathcal{G}$-based evidence retrieval utilizes an existing path-based approach. Such an approach searches for paths between `:Ronaldo` and `:SCP` in the reference $\mathcal{G}$ and utilizes them to calculate a single veracity score ζ. For example, COPAAL [52] returns a veracity score of 0.69 for the given triple based on DBpedia as reference graph.

Furthermore, HybridFC contains 3 multi-layer perceptron modules (ϑ_1, ϑ_2 and, ϑ_3). Each of the three multi-layer perceptron modules (ϑ_i) is defined as follows for an input vector x:

$$\vartheta_i = W_{5,i} \times D(ReLU(W_{3,i} \times (BN(W_{1,i} \times x)))) \,, \tag{2}$$

where $W_{j,i}$ is the weight matrix of an affine transformation in the j-th layer of the multi-layer perceptron, `ReLU` is an activation function, D stands for a Dropout layer [56], $\times$ represents the matrix multiplication, and BN represents the Batch Normalization [20]. The Batch Normalization and Dropout layers are defined as follows.

Given $x\prime$ as input, the batch normalization is formally defined as:

$$BN(x') = \beta + \gamma \frac{x' - \mathrm{E}[x']}{\sqrt{\mathrm{Var}[x']}} \,, \tag{3}$$

where, $\mathrm{E}\left[x'\right]$ and $\mathrm{Var}\left[x'\right]$ 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 [20].

Given x as input to the Dropout layer D, the elements of the layer's output vector $\bar{x}$ are computed as:

$$\bar{x}_i = \delta_i x_i \,, \tag{4}$$

where each δ_i is sampled from a Bernoulli distribution with parameter r, i.e., δ is 1 with probability r, and 0 otherwise.

In our approach, we add a fourth input vector that comprises the embedding vectors of a pre-trained TKGE model of the given quadruple. After adding the time embeddings $\varphi(\mathtt{t})$, the resultant equation of the final neural network component of HybridFC is as follows:

$$\omega = \sigma\left(w_\sigma^T \vartheta_3\left(\vartheta_1(\varphi_\aleph) \oplus \vartheta_2(\varphi(s,p,o) \oplus \zeta \oplus \varphi(t))\right)\right) \,, \tag{5}$$

where ω is the final veracity score of our fact-checking part, $\oplus$ stands for the concatenation of vectors, and w_σ is a weight vector that is multiplied with the output vector of the third module ϑ_3. For the given quadruple from our running example, we input $\varphi(\mathtt{2002})$ as the fourth input, and the fact-checking component was able to correctly classify it by producing the final veracity score of 0.95.

4.3 Time-point prediction

The time-point prediction component predicts the time-point in a certain range of years to form a correct quadruple $(\mathtt{s}, \mathtt{p}, \mathtt{o}, \mathtt{t})$ from the given assertion $(\mathtt{s}, \mathtt{p}, \mathtt{o})$. The output concatenated vector $\varphi(\mathtt{s,p,o})$ of the first component is fed as input to a multi-layer perceptron. This multi-layer perceptron consists of a Linear Layer, a Dropout layer, a Batch Normalization layer, a second Dropout layer, and a final Linear layer. It can be formalized as follows:

$$\gamma(s,p,o) = W_5 \times D(BN(D(W_1 \times \varphi(s,p,o)))) \,. \tag{6}$$

$\gamma(s,p,o)$ is a vector of size n, where n is the number of years in the targeted range of years. This vector is normalized to transform its values into probabilities and the vector into a distribution. The year with the highest predicted probability is returned as the predicted year and is the final output for the time-point prediction task. In our running example, for the given assertion (`:Ronaldo`, `:runningContractSigned`, `:SCP`), our time-prediction component predicts the correct year as 2002.

5 Experiments

Our evaluation has two objectives: we want to measure TEMPORALFC's abilities to a) discern between true and false assertions, and b) identify the appropriate point in time for a given, true assertion. In the following, we describe the experiment designs.

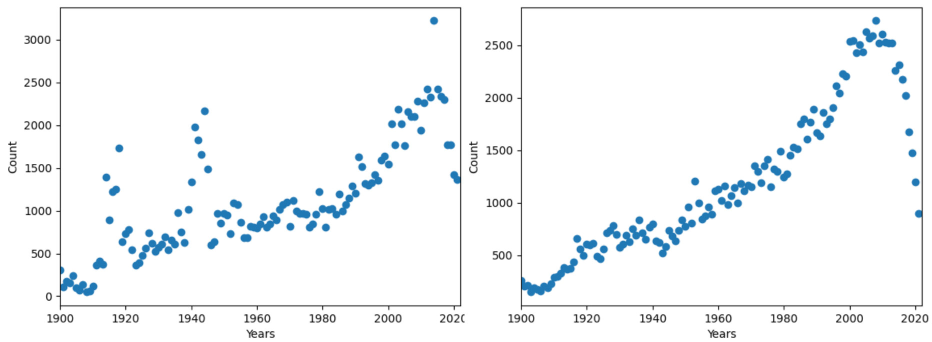

Fig. 3: The distribution of quadruples over years for the training sets of DBpedia124k (left) and Yago3k (right).

Table 2: Overview of updated DBpedia124k and Yago3k datasets used in experiments; the abbreviations are: Q/Quadruples, Ent./Entities, Rel./Relations, Tim./Time stamps, |.|/count.

<table>
<tr><th>Dataset</th><th>|Train-Q|</th><th>|Valid-Q|</th><th>|Test-Q|</th><th>|Ent.|</th><th>|Rel.|</th><th>|Tim.|</th></tr>
<tr><td>DBpedia124k</td><td>145k</td><td>44k</td><td>44k</td><td>124k</td><td>7</td><td>123</td></tr>
<tr><td>Yago3k</td><td>7k</td><td>2k</td><td>2k</td><td>3k</td><td>8</td><td>195</td></tr>
</table>

5.1 Datasets

We reuse two TKGs, DBpedia124k and Yago3k, from [36]. Due to the small size and incorrect IRIs in the original datasets, we update it by running the queries used to generate these datasets again on recent versions of the DBpedia and Yago datasets. Due to the Temporal-DeFacto requirement and, consequently, to ensure a fair comparison, we filtered all quadruples and kept those with year information between 1900 and 2022. We dubbed the resultant dataset with DBpedia124k and Yago3k due to the number of entities present in them. The statistics of the resultant datasets are shown in Table 2. Furthermore, we do not use the FactBench dataset [19] because it is based on older versions of DBpedia (i.e., 2013-10) and Freebase (i.e., 2013-08), which contain many entities (650/1813) for which $\mathcal{TG}$ model failed to produce embedding vectors.[5]

Our fact-checking 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.[6] We use the English Wikipedia dump from March 7th, 2022. For the Elasticsearch index, we use a cluster of 3 nodes with a combined storage of 1 TB and 32 GB RAM per node. Figure 3 shows the frequency of time points for the DBpedia124k and Yago3k train sets.

The process of generating negative examples for the fact-checking task requires more effort than generating positive examples [19]. For the purpose of examining the

[5] Fair comparison could not be possible with missing entities, which constitute many assertions.
[6] `https://www.elastic.co/`

contribution of the assertion $(\texttt{s}, \texttt{p}, \texttt{o})$ part of the quadruple and the time-point (t) part of the quadruple in the overall result, we generated two sets of negative examples: 1. The assertion-based negative example set is generated using the same strategy as defined in [28]. 2. The time-point-based negative example set is generated by randomly replacing time points as suggested in [19].

5.2 Evaluation metrics

We use common measures to evaluate the performance of the different approaches on the two tasks. When evaluating the fact-checking task, we rely on the area under the receiver operator characteristic curve (AUROC) [23,52,51]. We use the GERBIL framework to calculate this score [41,37].

The time prediction task is based on queries of the form $(\texttt{s}, \texttt{p}, \texttt{o}, \texttt{?})$, which are generated by removing the correct time point t from a set of test quadruples. The evaluated systems predict scores for all years that we have within our dataset. We rank the years according to the predicted scores and use the MRR, Hits@1, and Hits@3 to determine the ranking quality. In addition, we use the accuracy metric of multi-class classification to measure system performance in cases where only the highest-ranked year is considered.

5.3 Setup Details and Reproducibility

TEMPORALFC is designed to work with any pre-trained TKGE model to generate embedding vectors. *However, throughout our experiments, we solely use pre-trained embedding models because training large embedding models has a high footprint in terms of cost, energy, and CO_2 emissions [48].* Within our evaluation, we use a pre-trained T-DYHE model, since this approach has been reported to outperform other approaches for the time-point prediction task [36]. The size of each embedding vector is set to $q = 100$. The range of years is between 1900 and 2022.[7] The loss functions for training our multi-layer perceptron are the *Binary Cross Entropy Loss* for the fact-checking task and the *Cross Entropy Loss* for the time-point prediction task. *We chose the Binary Cross Entropy Loss because it is widely used for binary classification and is well-suited for models with a sigmoid activation function in the output layer [42,47,52]. We chose the Cross Entropy Loss for analogous reasons on multi-class classification [8,33].* With a batch size equal to one third of the training data, we set the maximum number of epochs to 1000. We calculate the validation loss after every 10th epoch and stop the training earlier if the loss is not reduced for 50 epochs to avoid overfitting.[8] Throughout our experiments, we use Adam [24] optimizer. For the fact-checking component, we use a pre-trained SBert model for sentence vector generation.[9] Furthermore, we set $k = 3$

[7] `https://doi.org/10.5281/zenodo.7913193`

[8] We report the parameters that were used to achieve the results reported in this study. Nevertheless, the user has the option to modify these parameters to suit her personal preferences. Visit the project home page to get the complete list of parameters.

[9] Among all the available pre-training models from the SBert webpage (`https://www.sbert.net/docs/pretrained_models.html`), we select `nq-distilbert-base-v1` for our approach (as suggested in [42]).

in the sentence selection module. SBert generates sentence embedding vectors of 768, which leads to $|\varphi_{\aleph}| = (3 \times 768) + 3 = 2307$.[10]

All experiments are executed on a computer with 32 CPU cores, 128 GB of RAM, and an NVIDIA GeForce RTX 3090. For the sake of reproducibility, we uploaded scripts for hyperparameter optimization, training, and evaluation to our project homepage.

5.4 Competing approaches

We compare our approach TEMPORALFC with HybridFC [42], FactCheck [51], Temporal-DeFacto [19], COPAAL [52], and KV-Rule [23], which are the state-of-the-art approaches in the hybrid, text-, path-, and rule-based categories of the fact-checking task. We also compare our results to the four KG embedding-based approaches TRANSE, CONEX, COMPLEX, and QMULT, which show the most effective performance for the fact-checking tasks [42].

For the time-point prediction task, we compare our approach with Temporal-DeFacto and the top-performing temporal embedding-based approaches: T-DYHE [36], T-TRANSE [30], and T-COMPLEX [28]. For the embedding-based approaches, we use the parameter configuration reported in [36].

6 Results and Discussion

In this section, we discuss the results we obtained in our evaluation. All results along with the scripts to reproduce the results are also available at the project homepage. First, we evaluate TEMPORALFC on the fact-checking task. Thereafter, we compare and evaluate the time-point prediction task with the state-of-the-art approaches.

6.1 Fact checking

Tables 3 and 4 show the results for the fact-checking task on the training and test data, respectively. In comparison to other approaches, hybrid approaches perform best.[11]

[10] *We ran experiments with other values of k, i.e., 1, 2, 3, and 5 and found that $k = 3$ worked best for our approach. We cannot present comprehensive results in this paper due to space limitations. However, they can be found in our extended, green open-access version of the paper.*

[11] Our results are also available on the GERBIL benchmarking platform [43]:
1. Using assertion-based negative examples:
   `http://w3id.org/gerbil/kbc/experiment?id=202301180129`,
   `http://w3id.org/gerbil/kbc/experiment?id=202301180056`,
   `http://w3id.org/gerbil/kbc/experiment?id=202301180123`, and
   `http://w3id.org/gerbil/kbc/experiment?id=202301180125`.
2. Using time-based negative examples:
   `http://w3id.org/gerbil/kbc/experiment?id=202305020014`,
   `http://w3id.org/gerbil/kbc/experiment?id=202305020015`,
   `http://w3id.org/gerbil/kbc/experiment?id=202305020012`, and
   `http://w3id.org/gerbil/kbc/experiment?id=202305020013`.

Table 3: Area under the curve (AUROC) score on DBpedia124k and Yago3k train sets; the abbreviations are: Txt/Text-based approaches, E./Example, Neg./negative, Path/Path-based approaches, H/Hybrid approaches, Gen./Generation, Avg./Average. Best performances are bold, second-best are underlined.

		Assertion-based Neg. E. Gen.			Time-based Neg. E. Gen.		
		DBpedia124k	Yago3k	Avg.	DBpedia124k	Yago3k	Avg.
Txt	FactCheck [51]	0.69	0.66	0.67	<u>0.55</u>	<u>0.53</u>	<u>0.54</u>
	Temporal-DeFacto [19]	0.67	0.64	0.65	0.53	0.51	0.52
KG-emb	TRANSE [5]	0.71	0.78	0.75	0.50	0.50	0.50
	CONEX [10]	0.80	0.81	0.80	0.50	0.50	0.50
	COMPLEX [54]	0.72	0.70	0.71	0.50	0.50	0.50
	QMULT [9]	0.73	0.77	0.75	0.50	0.50	0.50
H	HybridFC [42]	<u>0.92</u>	<u>0.94</u>	<u>0.93</u>	0.50	0.50	0.50
	TEMPORALFC	**0.93**	**0.97**	**0.95**	**0.68**	**0.65**	**0.66**
Path	KV-Rule [23]	0.58	0.61	0.59	0.50	0.50	0.50
	COPAAL [52]	0.65	0.67	0.66	0.50	0.50	0.50

This is expected since hybrid approaches combine aspects of different categories of approaches. Among both of the hybrid approaches, TEMPORALFC performs slightly better (Avg. $0.02 - 0.03$ on AUROC scores) than HybridFC on the DBpedia124k dataset, when using assertion-based negative examples in our dataset.[12] A potential reason for this small improvement could be that the additional temporal embeddings add more context to the data and, thus, the performance of the model increases.

When using time-based negative quadruples, TEMPORALFC outperforms other approaches by at least 0.12 AUROC on the train set and 0.13 AUROC on the test set. This difference is also due to the fact that most other approaches do not consider the temporal aspect of quadruples. Therefore, their classifiers do not consider temporal information during the training phase. *We use these negative examples to evaluate the potential performance that the fact-checking component would have if the fact-checking task would take time into account.* We modified the behavior of FactCheck and Defacto to include the temporal aspect as well, by including time-points in their search queries to the reference corpus. However, our results in Tables 3 and 4 show that the benefit of using the temporal aspect is not pertinent for FactCheck and Defacto.

During the training phase, we observe that our approach needs more epochs (e.g., 989 for DBpedia124k) than the comparable non-temporal approach HybridFC (479 epochs for DBpedia124k). This is most probably caused by the larger input vectors of the temporal embeddings.

The text-based approaches rely on the reference corpus as background knowledge. A look into the details of their output reveals that these approaches failed to find relevant evidence for 70% of the assertions, resulting in lower performance than hybrid or KG-based approaches for assertion-based negative sampling-based datasets. KG-embedding-based

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

Table 4: Area under the curve (AUROC) score on DBpedia124k and Yago3k test sets; the abbreviations are: Txt/Text-based approaches, E./Example, Neg./negative, Path/Path-based approaches, H/Hybrid approaches, Gen./Generation, Avg./Average. Best performances are bold, second-best are underlined.

		Assertion-based Neg. E. Gen.			Time-based Neg. E. Gen.		
		DBpedia124k	Yago3k	Avg.	DBpedia124k	Yago3k	Avg.
Txt	FactCheck [51]	0.69	0.66	0.67	<u>0.52</u>	<u>0.53</u>	<u>0.52</u>
	Temporal-DeFacto [19]	0.64	0.66	0.65	0.51	0.51	0.51
KG-emb	TRANSE [5]	0.74	<u>0.74</u>	0.74	0.50	0.50	0.50
	CONEX [10]	0.78	0.74	0.76	0.50	0.50	0.50
	COMPLEX [54]	0.74	0.70	0.72	0.50	0.50	0.50
	QMULT [9]	0.71	0.73	0.72	0.50	0.50	0.50
H	HybridFC [42]	<u>0.88</u>	**0.92**	<u>0.90</u>	0.50	0.50	0.50
	TEMPORALFC	**0.91**	**0.92**	**0.91**	**0.65**	**0.65**	**0.65**
Path	KV-Rule [23]	0.54	0.56	0.55	0.50	0.50	0.50
	COPAAL [52]	0.65	0.69	0.67	0.50	0.50	0.50

approaches achieve relatively better performance than text- and path-based approaches. In fact, CONEX is the third best performing system, followed by QMULT, TRANSE, and COMPLEX. Our results also show that KV-Rule performs worse among all competing approaches. This behavior of KV-Rule could be due to the fact that the pre-generated rule set is biased towards certain properties of the DBpedia dataset. Hence, for unknown properties the performance degrades. After the KV-Rule, COPAAL has the second-worst AUROC scores. It might be because COPAAL fails to find paths for the properties of our dataset and performs better on other properties [42]. These experimental findings imply that our strategy effectively utilizes the performance variability of the approaches it comprises. It appears to rely on KG-embedding-based approach's strong performance in particular. However, it also has the ability to supplement KG-embedding-based approach's predictions with those of other categories of approaches, in cases in which KG-embedding-based approach does not perform well.

6.2 Time-point prediction

Table 5 shows the results of the time-point prediction experiment. The results show that TEMPORALFC significantly outperforms all competing approaches in MRR by at least 0.27 and 0.25, Accuracy by at least 0.22 and 0.23, and Hit@1 by at least 0.48 and 0.54 on the DBpedia124k and Yago3k datasets, respectively. Temporal-DeFacto, TEMPORALFC's closest competitor, performs worst in terms of MRR and Accuracy as compared to all other systems. A closer look at the results reveals two main reasons for the low performance of Temporal-DeFacto. First, it fails to extract pieces of evidence for around 30 percent of quadruples ($43.5k/145k$ on DBpedia124k and $2k/7k$ on Yago3k). Second, its manual feature engineering seems to be optimized for the FactBench dataset [19] proposed by the authors. T-DYHE is the second best performing

Table 5: Results for the time-point prediction task on the DBpedia124k and Yago3k test datasets.

		MRR	Accuracy	Hits@1	Hits@3
DBpedia124k	Temporal-DeFacto [19]	0.16	0.10	0.17	0.19
	T-DYHE [36]	0.43	0.44	0.14	0.62
	T-TRANSE [30]	0.26	0.25	0.08	0.10
	T-COMPLEX [28]	0.24	0.21	0.13	0.17
	TEMPORALFC	**0.70**	**0.66**	**0.65**	**0.71**
Yago3k	Temporal-DeFacto [19]	0.11	0.05	0.01	0.12
	T-DYHE [36]	0.58	0.56	0.24	**0.90**
	T-TRANSE [30]	0.24	0.25	0.02	0.13
	T-COMPLEX [28]	0.21	0.20	0.17	0.19
	TEMPORALFC	**0.83**	**0.79**	**0.78**	0.86

embedding-based approach after TEMPORALFC, scoring MRR of 0.43 and 0.58 on DBpedia124k and Yago3k datasets respectively. On the Yago3k dataset, it performs better than TEMPORALFC on Hit@3 scores.

7 Conclusion

In this study, we propose TEMPORALFC—a hybrid method for temporal fact-checking for knowledge graphs. The goal of TEMPORALFC is to address two key issues: 1. most fact checking approaches do not take the volatility of some assertions into account and 2. those that do only achieve a low performance in the time-point prediction task. In both fact-checking and time-prediction tasks, we evaluate TEMPORALFC against the current state of the art. Our findings on two datasets imply that our TEMPORALFC approach can outperform competing approaches in both fact-checking and time-point prediction tasks. In particular, on the fact-checking task TEMPORALFC achieves an Area Under the Receiver Operating Characteristic curve that is 0.13 and 0.15 higher than the best competing approaches for volatile assertions, while it achieves the same or an even slightly superior performance as the current state-of-the-art approach HybridFC for non-volatile assertions. In the time-point prediction task, TEMPORALFC outperforms all other approaches in our evaluation by at least 0.25 to 0.27 Mean Reciprocal Rank.

In future work, we will enhance TEMPORALFC to support time-period-based assertions in $\mathcal{TG}$. In addition, we plan to extend the fact-checking component to include rule-based approaches.

Supplemental Material Statement

- The source code of TEMPORALFC, the scripts to recreate the full experimental setup, and the required libraries can be found on GitHub.[13]

[13] Source code: `https://github.com/dice-group/TemporalFC`

- For the fact-checking task, the datasets used in this paper and the output generated by text-based and path-based approaches on these datasets are available at Zenodo: `https://doi.org/10.5281/zenodo.7913193`.
- For the Time-point prediction task, the datasets are also available at Zenodo: `https://doi.org/10.5281/zenodo.7913222`.
- Pre-trained embeddings for these datasets are also available at Zenodo: `https://doi.org/10.5281/zenodo.7913193`.
- Prediction files and AUROC graphs are also available at Zenodo: `https://doi.org/10.5281/zenodo.7913193`.

Acknowledgments

This work is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 860801, the German Federal Ministry of Education and Research (BMBF) within the project NEBULA under the grant no 13N16364, the Ministry of Culture and Science of North Rhine-Westphalia (MKW NRW) within the project SAIL under the grant no NW21-059D, the German Research Foundation (DFG) within the project INGRID under the grant no NG 105/7-3, and the European Union's Horizon Europe research and innovation programme under grant agreement No 101070305. This work is also funded by the FWF Austrian Science Fund and the Internet Foundation Austria under the FWF Elise Richter and netidee SCIENCE programmes as project number V 759-N. This is the pre-print version of paper, which is accepted at ISWC 2023.

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 World Wide Web. p. 143–146. 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: The semantic web, pp. 722–735. Springer (2007)
3. Balazevic, I., Allen, C., Hospedales, T.: TuckER: Tensor factorization for knowledge graph completion. In: EMNLP-IJCNLP. pp. 5185–5194. Association for Computational Linguistics, Hong Kong, China (Nov 2019). https://doi.org/10.18653/v1/D19-1522, `https://aclanthology.org/D19-1522`
4. Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence **35**(8), 1798–1828 (2013)
5. Bordes, A., Usunier, N., Garcia-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: NIPS. p. 2787–2795. Curran Associates Inc., Red Hook, NY, USA (2013)
6. Chekol, M.W.: Tensor decomposition for link prediction in temporal knowledge graphs. In: Proceedings of the 11th on Knowledge Capture Conference. p. 253–256. ACM, New York, NY, USA (2021). https://doi.org/10.1145/3460210.3493558, `https://doi.org/10.1145/3460210.3493558`

7. Chen, Y., Goldberg, S., Wang, D.Z., Johri, S.S.: Ontological pathfinding: Mining first-order knowledge from large knowledge bases. In: ICMD. p. 835–846. ACM, New York, NY, USA (2016). https://doi.org/10.1145/2882903.2882954, https://doi.org/10.1145/2882903.2882954
8. Dasgupta, S.S., Ray, S.N., Talukdar, P.: HyTE: Hyperplane-based temporally aware knowledge graph embedding. In: EMNLP. pp. 2001–2011. Association for Computational Linguistics, Brussels, Belgium (2018). https://doi.org/10.18653/v1/D18-1225, https://aclanthology.org/D18-1225
9. 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)
10. Demir, C., Ngonga Ngomo, A.C.: Convolutional complex knowledge graph embeddings. In: European Semantic Web Conference. pp. 409–424. Springer (2021)
11. Dong, X.L., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., Zhang, W.: Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In: SIGKDD. pp. 601–610 (2014), http://www.cs.cmu.edu/~nlao/publication/2014.kdd.pdf
12. Drucker, P.F.: The age of discontinuity: Guidelines to our changing society. Transaction Publishers (2011)
13. Ermilov, I., Lehmann, J., Martin, M., Auer, S.: Lodstats: The data web census dataset. In: Groth, P., Simperl, E., Gray, A., Sabou, M., Krötzsch, M., Lecue, F., Flöck, F., Gil, Y. (eds.) The Semantic Web – ISWC 2016. pp. 38–46. Springer International Publishing, Cham (2016)
14. Galárraga, L., Teflioudi, C., Hose, K., Suchanek, F.M.: Fast rule mining in ontological knowledge bases with amie+. The VLDB Journal **24**(6), 707–730 (Dec 2015). https://doi.org/10.1007/s00778-015-0394-1, https://doi.org/10.1007/s00778-015-0394-1
15. Galárraga, L.A., Teflioudi, C., Hose, K., Suchanek, F.: Amie: Association rule mining under incomplete evidence in ontological knowledge bases. In: World Wide Web. p. 413–422. World Wide Web '13, ACM, New York, NY, USA (2013). https://doi.org/10.1145/2488388.2488425, https://doi.org/10.1145/2488388.2488425
16. García-Durán, A., Dumančić, S., Niepert, M.: Learning sequence encoders for temporal knowledge graph completion. In: EMNLP. pp. 4816–4821. Association for Computational Linguistics, Brussels, Belgium (2018). https://doi.org/10.18653/v1/D18-1516, https://aclanthology.org/D18-1516
17. Gardner, M., Mitchell, T.: Efficient and expressive knowledge base completion using subgraph feature extraction. In: EMNLP. pp. 1488–1498 (2015)
18. Gardner, M., Talukdar, P., Krishnamurthy, J., Mitchell, T.: Incorporating vector space similarity in random walk inference over knowledge bases. In: EMNLP. pp. 397–406. Association for Computational Linguistics, Doha, Qatar (Oct 2014). https://doi.org/10.3115/v1/D14-1044, https://www.aclweb.org/anthology/D14-1044
19. Gerber, D., Esteves, D., Lehmann, J., Bühmann, L., Usbeck, R., Ngonga Ngomo, A.C., Speck, R.: Defacto-temporal and multilingual deep fact validation. Web Semant. **35**(P2), 85–101 (Dec 2015). https://doi.org/10.1016/j.websem.2015.08.001, https://doi.org/10.1016/j.websem.2015.08.001
20. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: ICML. p. 448–456. JMLR.org (2015)
21. Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: Knowledge graph embedding via dynamic mapping matrix. In: IJCNLP. pp. 687–696. Association for Computational Linguistics, Beijing, China (Jul 2015). https://doi.org/10.3115/v1/P15-1067, https://www.aclweb.org/anthology/P15-1067

22. Jin, W., Zhang, C., Szekely, P.A., Ren, X.: Recurrent event network for reasoning over temporal knowledge graphs. CoRR **abs/1904.05530** (2019), http://arxiv.org/abs/1904.05530
23. Kim, J., Choi, K.s.: Unsupervised fact checking by counter-weighted positive and negative evidential paths in a knowledge graph. In: CICLing. pp. 1677–1686. International Committee on Computational Linguistics, Barcelona, Spain (Online) (Dec 2020). https://doi.org/10.18653/v1/2020.coling-main.147, https://www.aclweb.org/anthology/2020.coling-main.147
24. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR **abs/1412.6980** (2014)
25. Konstantinovskiy, L., Price, O., Babakar, M., Zubiaga, A.: Toward automated factchecking: Developing an annotation schema and benchmark for consistent automated claim detection. Digital Threats: Research and Practice **2**(2) (Apr 2021). https://doi.org/10.1145/3412869, https://doi.org/10.1145/3412869
26. Koster, A., Bazzan, A., Souza, M.d.: Liar liar, pants on fire; or how to use subjective logic and argumentation to evaluate information from untrustworthy sources. Artificial Intelligence Review **48** (08 2017). https://doi.org/10.1007/s10462-016-9499-1
27. Kotonya, N., Toni, F.: Explainable automated fact-checking for public health claims. arXiv preprint arXiv:2010.09926 (2020)
28. Lacroix, T., Obozinski, G., Usunier, N.: Tensor decompositions for temporal knowledge base completion (2020). https://doi.org/10.48550/ARXIV.2004.04926, https://arxiv.org/abs/2004.04926
29. Lajus, J., Galárraga, L., Suchanek, F.: Fast and exact rule mining with amie 3. In: The Semantic Web. pp. 36–52. Springer International Publishing, Cham (2020)
30. Leblay, J., Chekol, M.W.: Deriving validity time in knowledge graph. In: Companion Proceedings of the The Web Conference 2018. p. 1771–1776. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE (2018). https://doi.org/10.1145/3184558.3191639, https://doi.org/10.1145/3184558.3191639
31. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: AAAI. vol. 29 (2015)
32. Long, Y., Lu, Q., Xiang, R., Li, M., Huang, C.R.: Fake news detection through multi-perspective speaker profiles. In: ICNLP. pp. 252–256. Asian Federation of Natural Language Processing, Taipei, Taiwan (Nov 2017), https://aclanthology.org/I17-2043
33. Ma, Y., Tresp, V., Daxberger, E.A.: Embedding models for episodic knowledge graphs. Journal of Web Semantics **59**, 100490 (2019). https://doi.org/https://doi.org/10.1016/j.websem.2018.12.008, https://www.sciencedirect.com/science/article/pii/S1570826818300702
34. 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: International Semantic Web Conference. pp. 376–394. Springer International Publishing, Cham (2018)
35. 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: International Semantic Web Conference. pp. 376–394. Springer (2018)
36. Nayyeri, M., Vahdati, S., Khan, M.T., Alam, M.M., Wenige, L., Behrend, A., Lehmann, J.: Dihedron algebraic embeddings for spatio-temporal knowledge graph completion. In: ESWC. pp. 253–269. Springer International Publishing, Cham (2022)
37. 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/, last time accessed, March 30th 2022

38. Ortona, S., Meduri, V.V., Papotti, P.: Rudik: Rule discovery in knowledge bases. Proc. VLDB Endow. **11**(12), 1946–1949 (Aug 2018). https://doi.org/10.14778/3229863.3236231, https://doi.org/10.14778/3229863.3236231
39. Pasternack, J., Roth, D.: Knowing what to believe (when you already know something). In: CICLing. p. 877–885. Association for Computational Linguistics, USA (2010)
40. Pasternack, J., Roth, D.: Latent credibility analysis. In: World Wide Web. p. 1009–1020. World Wide Web '13, ACM, New York, NY, USA (2013). https://doi.org/10.1145/2488388.2488476, https://doi.org/10.1145/2488388.2488476
41. Paulheim, H., Ngonga Ngomo, A.C., Bennett, D.: Semantic web challenge 2018. Website (2018), http://iswc2018.semanticweb.org/semantic-web-challenge-2018/index.html, last time accessed, March 30th 2022
42. Qudus, U., Röder, M., Saleem, M., Ngomo, A.C.N.: Hybridfc: A hybrid fact-checking approach for knowledge graphs. In: International Semantic Web Conference. pp. 462–480. Springer International Publishing, Cham (2022). https://doi.org/10.1007/978-3-031-19433-7_27, https://papers.dice-research.org/2022/ISWC_HybridFC/public.pdf
43. Röder, M., Usbeck, R., Ngomo, A.N.: GERBIL - benchmarking named entity recognition and linking consistently. Semantic Web **9**(5), 605–625 (2018). https://doi.org/10.3233/SW-170286, http://www.semantic-web-journal.net/system/files/swj1671.pdf
44. Rula, A., Palmonari, M., Rubinacci, S., Ngomo, A.C.N., Lehmann, J., Maurino, A., Esteves, D.: Tisco: Temporal scoping of facts. Web Semant. **54**(C), 72–86 (jan 2019). https://doi.org/10.1016/j.websem.2018.09.002
45. Shi, B., Weninger, T.: Discriminative predicate path mining for fact checking in knowledge graphs. Know.-Based Syst. **104**(C), 123–133 (Jul 2016). https://doi.org/10.1016/j.knosys.2016.04.015, https://doi.org/10.1016/j.knosys.2016.04.015
46. Shiralkar, P., Flammini, A., Menczer, F., Ciampaglia, G.L.: Finding streams in knowledge graphs to support fact checking. In: ICDM. pp. 859–864 (2017). https://doi.org/10.1109/ICDM.2017.105
47. da Silva, A.A.M., Röder, M., Ngomo, A.C.N.: Using compositional embeddings for fact checking. In: International Semantic Web Conference. p. 270–286. Springer-Verlag, Berlin, Heidelberg (2021). https://doi.org/10.1007/978-3-030-88361-4_16
48. Strubell, E., Ganesh, A., McCallum, A.: Energy and policy considerations for deep learning in NLP. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 3645–3650. Association for Computational Linguistics, Florence, Italy (Jul 2019). https://doi.org/10.18653/v1/P19-1355, https://aclanthology.org/P19-1355
49. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: World Wide Web. pp. 697–706. ACM (2007)
50. Syed, Z.H., Röder, M., Ngomo, A.N.: Unsupervised discovery of corroborative paths for fact validation. In: International Semantic Web Conference. Lecture Notes in Computer Science, vol. 11778, pp. 630–646. Springer (2019). https://doi.org/10.1007/978-3-030-30793-6_36, https://doi.org/10.1007/978-3-030-30793-6_36
51. Syed, Z.H., Röder, M., Ngonga Ngomo, A.C.: Factcheck: Validating rdf triples using textual evidence. In: CIKM. p. 1599–1602. CIKM '18, ACM, New York, NY, USA (2018). https://doi.org/10.1145/3269206.3269308, https://doi.org/10.1145/3269206.3269308
52. Syed, Z.H., Srivastava, N., Röder, M., Ngomo, A.C.N.: Copaal - an interface for explaining facts using corroborative paths. In: International Semantic Web Conference (2019)

53. Syed, Z.H., Srivastava, N., Röder, M., Ngomo, A.N.: COPAAL - an interface for explaining facts using corroborative paths. In: International Semantic Web Conference. vol. 2456, pp. 201–204. CEUR-WS.org (2019), http://ceur-ws.org/Vol-2456/paper52.pdf
54. Trouillon, T., Welbl, J., Riedel, S., Gaussier, E., Bouchard, G.: Complex embeddings for simple link prediction. In: ICML. pp. 2071–2080 (2016)
55. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: A survey of approaches and applications. IEEE Transactions on Knowledge and Data Engineering **29**(12), 2724–2743 (2017). https://doi.org/10.1109/TKDE.2017.2754499
56. Watt, N., du Plessis, M.C.: Dropout algorithms for recurrent neural networks. In: SAICSIT. p. 72–78. ACM, New York, NY, USA (2018). https://doi.org/10.1145/3278681.3278691
57. Xu, C., Nayyeri, M., Alkhoury, F., Shariat Yazdi, H., Lehmann, J.: TeRo: A time-aware knowledge graph embedding via temporal rotation. In: CICLing. pp. 1583–1593. International Committee on Computational Linguistics, Barcelona, Spain (Online) (Dec 2020). https://doi.org/10.18653/v1/2020.coling-main.139, https://aclanthology.org/2020.coling-main.139
58. Yang, B., Yih, W., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: ICLR (2015), http://arxiv.org/abs/1412.6575