Fact Checking over Knowledge Graphs – A Survey

UMAIR QUDUS, MICHAEL RÖDER, MUHAMMAD SALEEM, and AXEL-CYRILLE NGONGA NGOMO, Data Science Research Group, Department of Computer Science, Paderborn University, Germany

Knowledge graphs are used by a growing number of applications to represent structured data. Hence, evaluating the veracity of assertions in knowledge graphs—dubbed fact checking—is currently a challenge of growing importance. However, manual fact checking is commonly impractical due to the sheer size of knowledge graphs. This paper is a systematic survey of recent works on automatic fact checking with a focus on knowledge graphs. We present recent fact-checking approaches, the varied sources they use as background knowledge, and the features they rely upon. Finally, we draw conclusions pertaining to possible future research directions in fact checking over knowledge graphs.

CCS Concepts: • **General and reference** → **Surveys and overviews**; • **Computing methodologies** → *Knowledge representation and reasoning*; Artificial intelligence; • **Information systems** → *Data cleaning*; Graph-based database models.

Additional Key Words and Phrases: fact checking, knowledge graphs, fact-checkers, check worthiness, evidence retrieval, trust, veracity.

ACM Reference Format:
Umair Qudus, Michael Röder, Muhammad Saleem, and Axel-Cyrille Ngonga Ngomo. 2023. Fact Checking over Knowledge Graphs – A Survey. 1, 1 (August 2023), 50 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Fact checking has been defined in a number of ways in the literature [14]. In general, fact checking can be understood as the task of computing the likelihood that a given assertion is true [14]. This discipline emerged in the area of journalism, where most of the fact checking was carried out manually by human fact checkers (e.g., expert journalists, bloggers, and analysts) [3] until recently. However, the proliferation of websites and bots spreading incorrect claims over the Web has led to the development of automated fact-checking approaches based on generalized sums [60], probabilistic logic [141], and constraint programming [98] amongst others. Corresponding services [128] (e.g., PolitiFact[1], FactCheck[2], and Snopes[3]) have also been developed to facilitate the use of (semi-)automatic fact-checking algorithms.

The need for the verification of assertions can also be observed on the Data Web and will be the focus of this paper. The participatory paradigm underlying the Data Web has led to more than 150 billion assertions on more than 3 billion things being published in more than 9,000 RDF knowledge graphs (KGs) [34]. Datasets such as Google Knowledge Graph [116], Freebase [15], DBpedia [8], NELL [20, 87], Yago [13, 53, 82, 117], and Wikidata [83] are known large-scale knowledge

[1] http://politifact.com
[2] http://factcheck.org
[3] http://snopes.com

Authors' address: Umair Qudus, umair.qudus@uni-paderborn.de; Michael Röder, michael.roeder@uni-paderborn.de; Muhammad Saleem, saleem@uni-paderborn.de; Axel-Cyrille Ngonga Ngomo, axel.ngonga@uni-paderborn.de, Data Science Research Group, Department of Computer Science, Paderborn University, Warburger Str. 100, Paderborn, NRW, Germany, 33098.

graphs with billions of assertions, which are updated regularly. The assertions found in these knowledge graphs are used in a growing number of applications (e.g., search engines [116], in-flight applications [83], community support systems [7] and even personal assistants such as Apple's Siri [83]). The correctness of knowledge graphs is hence mission-critical for these applications.

Manual fact-checking approaches are clearly impractical in the context of Web-scale KGs given the sheer size of KGs used in real-world applications. Moreover, fact-checking systems ideally need to operate in near real-time to match the rate at which these assertions are being made [114] as well as the needs of their end users [45, 52]. Hence, a large body of research has been devoted to developing automated fact-checking approaches for KGs (also called fact validation approaches) [35, 40, 66, 112, 114, 122]. This paper is a survey of this body of research and covers fact-checking approaches for knowledge graphs developed between 2014 and 2022. Our main contributions can be summarized as follows:

- We provide a **systematic survey** of works concerned with defining, understanding and representing the fact-checking task with a focus on KGs. This survey is the first of its kind and provides a comprehensive comparison of recent fact-checking approaches for KGs.
- We provide a **categorization** of different fact-checking approaches according to their methodologies used. Therewith, we aim to ease the path of novices into the research area.
- We provide a **comparison of the performance** of selected fact-checking algorithms using metrics such as runtime, average accuracy, F1-score, recall and precision scores.
- We provide a comprehensive comparison of fact-checking **datasets and benchmarks** proposed in literature.
- We provide a list of **open research questions** and **applications** based on our analysis of fact checking over KGs.

The rest of this paper is structured as follows: In Section 2, we introduce the notation required to understand the rest of the paper. In Section 3, we present our systematic survey methodology, which is divided into two parts: (1) related surveys on fact checking and (2) publications related to fact checking over knowledge graphs. This part is followed by a categorization of approaches in Section 4. Section 5 presents a detailed comparison of selected fact-checking approaches. Thereafter, related benchmarks and datasets are discussed in Section 6. We then discuss the results in Section 7. In Section 9, we provide an overview of research findings and open research questions. We also discuss applications in Section 10. Finally, we conclude and discuss potential future work in Section 11.

2 PRELIMINARIES

There are different definitions for fact checking and related concepts, which originate from various domains, e.g., journalism [62, 63, 80], natural language processing [98], and knowledge graphs [66, 112, 122]. In this section, we define the terminology and notation used throughout this survey. First, we define general terms, which are common in all approaches. Afterward, we define the terms related to specific categories of approaches. Table 1 lists the symbols used throughout this survey.

Definition 2.1 (Knowledge Graph)**.** Throughout this work, we use the definition provided by [54]. This definition states that a knowledge graph is a data graph, whose nodes represent entities of interest in the real world and whose edges represent relations between these entities. In general, a knowledge graph conforms to a graph-based data model, for example, the RDF model or the property graph model.

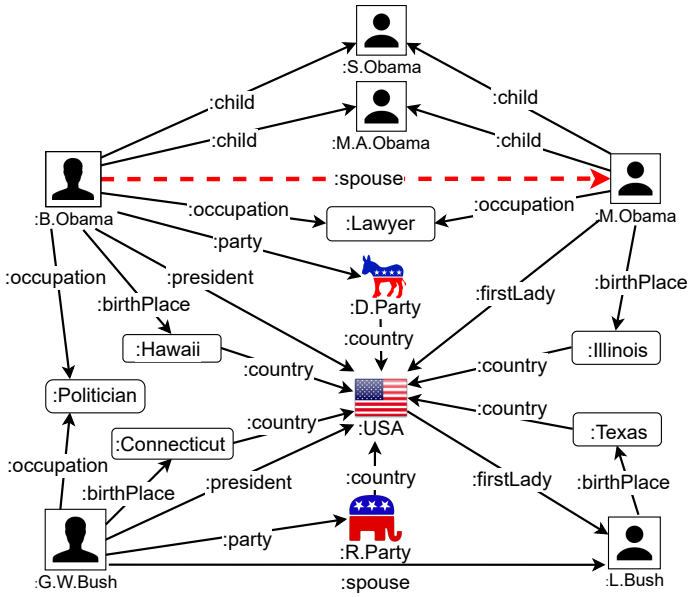


Fig. 1. A knowledge graph excerpt from a large knowledge graph. The dotted red line shows the given fact (to check the veracity). Filled black lines are edges (aka predicates) with labels, and the rest are nodes of different RDF classes, which include Person, State, County, Party, and Occupation.

There are different instantiations of knowledge graphs. For example, some works in the area of fact checking rely on heterogeneous information networks (HIN) [113, 121]. Still, most of the approaches surveyed herein use RDF knowledge graphs, which we define as follows:

Definition 2.2 (RDF Knowledge Graph)**.** An RDF knowledge graph $\mathcal{G}$ is a set of RDF triples, i.e.,

$$\mathcal{G} = \{s \xrightarrow{p} o \mid s \in \mathbb{E} \cup \mathbb{B}, p \in \mathbb{P}, o \in \mathbb{E} \cup \mathbb{B} \cup \mathbb{L}\}, \tag{1}$$

where $s \xrightarrow{p} o$ is a triple in $\mathcal{G}$ comprising a subject s, a predicate p and an object o. $\mathbb{E}$ is the set of all RDF resource IRIs, $\mathbb{B}$ is the set of all blank nodes, $\mathbb{P} \subseteq \mathbb{E}$ is the set of all RDF predicates and $\mathbb{L}$ represents the set of all literals [25]. [4]

Figure 1 depicts an excerpt of an RDF knowledge graph. We use this graph as a running example throughout this section.

$\mathcal{G}$ can contain terminological information, which can be exploited for fact checking. Current approaches exploit the following.

Definition 2.3 (Class of resource)**.** The predicate `rdf:type` is used to state that an RDF resource r is an instance of an RDF class c. This is done via the statement:

$$r \xrightarrow{\texttt{rdf:type}} c. \tag{2}$$

From this statement, one can further infer that c is an instance of `rdfs:Class` and $r \in \mathbb{E} \cup \mathbb{B}$ [18].

[4] From here on, we work with IRIs. The prefixes for these IRIs that we use are rdfs, rdf and, ":". The core RDF vocabulary is defined in a namespace informally called 'rdfs'. That namespace is identified by the URI-Reference http://www.w3.org/2000/01/rdf-schema# and is associated with the prefix 'rdfs'. This specification also uses the prefix 'rdf' to refer to the RDF namespace http://www.w3.org/1999/02/22-rdf-syntax-ns# [19]. Furthermore, we use ":" prefix for literals.

Table 1. List of symbols

Notation	Description	Notation	Description
$\mathcal{G}$	Knowledge graph	$\mathcal{O}$	Ontology
$C(\cdot)$	Function to return the type (rdf:type) of any RDF resource	$\mathbb{B}, \mathbb{C}, \mathbb{E}, \mathbb{L}, \mathbb{P}$	All blank nodes set, RDFS classes, RDF resources, Literals and RDF predicates, respectively
$D(p)$	Domain of the predicate p	$\mathcal{R} = \mathcal{B} \Rightarrow \mathcal{H}$	Horn rule
$R(p)$	Range of the predicate p	$\mathcal{S}$	Support
π_k	Meta paths of length k	$\mathcal{C}$	Confidence
$\mu_{(C(s),C(o),\pi_k)}$	An anchored predicate path w.r.t s, o and π_k	$\zeta(p)$	Functionality score of the predicate p
$\mathcal{A}_{(k,C(s),C(o))}$	Set of anchored predicate paths of length k between s and o	$\Pi^k_{(C(s),C(o))}$	Discriminative paths of length k between $C(s)$ and $C(o)$
$\Omega^k(p)$	Corroborative paths of length k between s and o	Γ^-	Negative (counter) examples for training
$p(\Gamma^+)$	Triples containing the positive examples with p as relation	Γ^+	Positive examples for training
$p(\Gamma^-)$	Triples containing the counterexamples with p as relation	$L(\cdot)$	Function to return the description of any resource
$s \xrightarrow{p} o$	Triple/Assertion ($s = subject \xrightarrow{p=predicate} o = object$)	$p(X, Y)$	Relation p between X, Y (constants or variables)
$\mathcal{B}$	Query of body atoms $B_1 \wedge \ldots \wedge B_n$	$\mathcal{H}$	Head atom
n_i	Intermediate nodes: $i \in 1 \ldots n$	R_i	Relation on i_{th} triple
$x_1 \ldots x_n, y_1 \ldots y_n$	Intermediate predicates on the path	$e \in E$	Edge in edges set
ω	Set of path type sequences (features)	$\mathcal{P}$	Prediction
$v \in V$	Vertex in vertices set	p^{-1}	Inverse property
φ	Graph fact checking rule (GFC)	$\aleph(s, o)$	Subgraph pattern
$\lvert . \rvert$	Absolute value	m	Depth of a subgraph

Table 2. Abbreviations and definitions

Notation	Description	Terms	Description
CWA	Closed world assumption	Claim	A statement or assertion
OWA	Open world assumption	A statement or assertion	A triple in $\mathcal{G}$
PCWA	Partial closed world assumption		
D-PCWA	Distant partial closed world assumption	A Fact	A statement/assertion that is known or proved to be true
E-PCWA	Extended partial closed world assumption	Evidence	Information that can be used to assess the truthfulness of an assertion or a claim
WPCA	Weak partial closed world assumption	Topic	Phrase describing a document
LCWA	Local Closed-World Assumption	Utterance	Expression of a claim in natural language
FS	Fact-spotting	Anchored nodes	Endpoint nodes (i.e., start or end) of a path

In this survey, we use $C(r)$ to denote the set of all classes of which r is an instance. In our running example, `:G.W.Bush` is a `:Person`, i.e., `:Person` $\in C($`:G.W.Bush`$)$.

Definition 2.4 (Domain and range)**.** The domain of p, denoted $D(p)$, is the set of classes such that $(s \xrightarrow{p} o) \rightarrow D(p) \subseteq C(s)$. The range of p, denoted $R(p)$, is the set of classes such that $(s \xrightarrow{p} o) \rightarrow R(p) \subseteq C(o)$.

Let the domain of the property `:party` be {`:Person`} in our running example. Then, all subjects of triples that include `:party` as predicate are instances of the class `:Person`. Analogously, `:R.Party` can be inferred to be an instance of the RDF class `:PoliticalParty` based on the range of the property `:party` being the singleton {`:PoliticalParty`}.

Two important classes are the datatype properties and object properties:

Definition 2.5 (Datatype property)**.** Given a triple $s \xrightarrow{p} o$, if p is a datatype property, then $o \in \mathbb{L}$ [11].

For example, `:B.Obama` $\xrightarrow{\texttt{:birthDate}}$ "`August 4, 1960`".

Definition 2.6 (Object property)**.** Given a triple $s \xrightarrow{p} o$, if p is an object property, then $o \in \mathbb{E} \cup \mathbb{B}$ [11].

For example, `:B.Obama` $\xrightarrow{\texttt{:spouse}}$ `:M.Obama`.

Definition 2.7 (Inverse property)**.** Given an object property p, the inverse of p —denoted p^{-1}— is the property such that $(s \xrightarrow{p} o) \rightarrow (o \xrightarrow{p^{-1}} s)$. Note that $D(p^{-1}) = R(p)$ and $R(p^{-1}) = D(p)$, and $\mathbb{P}^{-1}$ is the set of all inverse properties [11].

To exemplify, `:G.W.Bush` $\xleftarrow{\texttt{:spouse}}$ `:L.Bush`[5] is a direct consequence of the assertion `:G.W.Bush` $\xrightarrow{\texttt{:spouse}}$ `:L.Bush` in our knowledge graph.

Definition 2.8 (Functional property)**.** A functional property is a property that can have only one (unique) value object o for each subject s, i.e., if $s \xrightarrow{p} o_1$ and $s \xrightarrow{p} o_2$ hold, then $o_1 = o_2$ can be inferred [18].

If more than one object can be associated with a given subject by virtue of a property p, then p is either quasi-functional or not functional. The functionality score of a property is defined as follows.

Definition 2.9 (Functionality score)**.**

$$\zeta(p) = \frac{|\{s \mid \exists o : s \xrightarrow{p} o \in \mathcal{G}\}|}{|\{(s,o) \mid s \xrightarrow{p} o \in \mathcal{G}\}|}. \tag{3}$$

Note that $\zeta(p) \in [0, 1]$. It is exactly 1 for functional properties, e.g., `:birthDate`. It is close to 1 for quasi-functional properties, and it is smaller for non-functional properties, which can link a single subject to many objects, such as `:actedInMovie` [66].

2.1 Fact checking over knowledge graphs

For this survey, we build upon the definition of fact checking over knowledge graphs suggested in [123]:

Definition 2.10 (Fact Checking)**.** Given an assertion in the form of a triple $s \xrightarrow{p} o$, a reference knowledge graph $\mathcal{G}$, and/or a reference corpus, fact checking is the task of computing the likelihood that the given assertion is true or false.

In our running example, `:B.Obama` $\xrightarrow{\texttt{:spouse}}$ `:M.Obama` is the statement that is to be fact checked. To perform fact checking, supervised and rule-mining-based methods (see Section 5 for details) rely on the generation of counterexamples.

[5]The left arrow notation represents the inverse property

Definition 2.11 (Counterexamples)**.** A counterexample to a given triple $s \xrightarrow{p} o$ is a triple $x \xrightarrow{y} z$, which is similar (according to some measure of similarity) to $s \xrightarrow{p} o$ but has the opposite truth value. Commonly $x \xrightarrow{y} z$ is generated from $s \xrightarrow{p} o \in \mathcal{G}$ by altering a single member or several members of the triple in such a manner that the resulting triple $x \xrightarrow{y} z \notin \mathcal{G}$ [45].

From our running example, some counterexamples are: `:G.W.Bush` $\xrightarrow{\texttt{:birthPlace}}$ `:Canada` or `:B.Obama` $\xrightarrow{\texttt{:spouse}}$ `:L.Bush`. A set of counter- and true examples are denoted as Γ^- and Γ^+, respectively. In the following, we look at assumptions made for inferring supplementary triples from $\mathcal{G}$, these assumptions generally assume that $\mathcal{G}$ contains only true assertions.

Definition 2.12 (Open-world assumption (`OWA`))**.** Under the `OWA`, an assertion that is not contained in $\mathcal{G}$ and cannot be inferred from $\mathcal{G}$ is not necessarily false; its veracity (i.e., whether it is true or false) is just unknown [54].

This is a crucial difference to many standard database settings, which operate under the closed-world assumption (CWA).

Definition 2.13 (Closed World Assumption (`CWA`))**.** The closed-world assumption states that any statement that is neither in $\mathcal{G}$ nor can be inferred from $\mathcal{G}$ is false. Thus, each of these false statements can be in Γ^- under the closed world assumption [118].

For example, our running example does not mention whether `:G.W.Bush` and `:L.Bush` have children. Under the `CWA`, one can infer from the example that they do not have children. Similarly, the statement `:B.Obama` $\xrightarrow{\texttt{:spouse}}$ `:M.Obama` $\notin \mathcal{G}$, so it would be inferred to be false under the `CWA`, which is incorrect at the time of writing. However, under `OWA`, one can conclude from our example that `:G.W.Bush` and `:L.Bush` may or may not have children and that `:B.Obama` may or may not be married to `:M.Obama`. Hence, most fact-checking approaches refrain from assuming the `CWA`.

Definition 2.14 (Partial closed-world assumption (`PCWA`))**.** The idea behind `PCWA` is that given a statement $s \xrightarrow{p} o$ in the knowledge graph $\mathcal{G}$, and $\zeta(p) \geq \zeta\left(p^{-1}\right)$, we can assume that all $s \xrightarrow{p} o' \notin \mathcal{G}$ are valid counterexamples. Under the same assumption, we can derive that all $s' \xrightarrow{p} o \notin \mathcal{G}$ are valid counterexamples if $\zeta(p) < \zeta\left(p^{-1}\right)$ [66].

`PCWA` is alternatively denoted as Local Closed-World Assumption (`LCWA`) in the literature [58].

Definition 2.15 (Evidence for a statement)**.** A piece of information that can be used to assess the veracity of a given statement $s \xrightarrow{p} o$ is called evidence [14]. Evidence can be of different forms. Examples include a sentence in a document, a row in a web table, and an assertion in a knowledge graph [71].

For instance, "Barack Obama is the President of the United States, born in Hawaii..." from Wikipedia can be regarded as evidence for the statement `:B.Obama` $\xrightarrow{\texttt{:president}}$ `:USA`.

2.2 Meta-path-based approaches

Some fact-checking approaches rely on paths. The notation used in path-based approaches is as follows.

Definition 2.16 (Simple path)**.** A path of length k in a knowledge graph $\mathcal{G}$ is a cycle-free sequence of triples from $\mathcal{G}$ of the form $v_0 \xrightarrow{p_1} v_1 \xrightarrow{p_2} v_2 \ldots v_{k-1} \xrightarrow{p_k} v_k$, where v_i with $0 \leq i \leq k$ represents a subject or an object entity along the path [112].

For example, $\texttt{:B.Obama} \xrightarrow{\texttt{:party}} \texttt{:D.Party} \xrightarrow{\texttt{:country}} \texttt{:USA} \xrightarrow{\texttt{:firstLady}} \texttt{:M.Obama}$ is a path of length 3 in Figure 1. Several paths can exist between two nodes in the graph. A path can be a directed path as the example above, or an undirected path. Intermediate edges in a directed path are in a single direction (i.e., either subject to object or object to subject). On the other hand, edges in an undirected path are in both directions. An example of an undirected path is: $\texttt{:B.Obama} \xrightarrow{\texttt{:occupation}} \texttt{:Lawyer} \xleftarrow{\texttt{:occupation}} \texttt{:M.Obama}$.

Definition 2.17 (Specificity of a path)**.** The specificity $\mathcal{S}$ of a path $(v_0 \xrightarrow{p_1} v_1 \xrightarrow{p_2} v_2 \ldots v_{k-1} \xrightarrow{p_k} v_k)$ with $k+1$ nodes is inversely proportional to the sum of logarithms of the degrees of its intermediate nodes. It is formally defined as:

$$\mathcal{S} = \frac{1}{1 + \sum_{i=1}^{k-1} \log d\,(v_i)} \tag{4}$$

where $d\,(v_i)$ represents the degree of intermediate nodes in $\mathcal{G}$.

Definition 2.18 (Meta path)**.** Given a path of length k in a knowledge graph $\mathcal{G}$, a meta path π_k based on this path is defined as a directed, sequence of classes and predicates: $C_0 \xrightarrow{q_1} C_1 \xrightarrow{q_2} C_2 \ldots C_{k-1} \xrightarrow{q_k} C_k$ in $\mathcal{G}$, such that $C_i \in C\,(v_i)$, where $C\,(v_i)$ denotes the classes of vertex v_i, q_i represents the predicate of a directed edge that connects vertex V_i to V_{i+1}, such that $q_i \in \mathbb{P} \cup \mathbb{P}^{-1}$ [112].[6]

From our running example, the meta path $\pi^2_{ex} = \texttt{:Person} \xrightarrow{\texttt{:occupation}} \texttt{:Occupation} \xleftarrow{\texttt{:occupation}} \texttt{:Person}$ can be derived from the path : $B.Obama \xrightarrow{\texttt{:occupation}} \texttt{:Lawyer} \xleftarrow{\texttt{:occupation}} \texttt{:M.Obama}$.

Definition 2.19 (Anchored predicate path)**.** Given a meta path π_k, the corresponding anchored predicate path $\mu_{(C(s),C(o),\pi_k)}$ w.r.t. to a pair of entities (s, o) is the directed sequence of edges with classes of endpoints $\mu_{(C(s),C(o),\pi_k)} = C_0 \xrightarrow{\mathrm{p}_1} \xrightarrow{\mathrm{p}_2} \ldots \xrightarrow{\mathrm{p}_k} C_k$ with $C_0 \in C\,(s)$ and $C_k \in C\,(o)$ [112]. Furthermore, we denote a set of all anchored predicate paths of length k between two entities s and o as $\mathcal{A}_{(k,C(s),C(o))}$, and we write $\mathcal{A}_k$ as a short hand for $\mathcal{A}_{(k,C(s),C(o))}$.

For our running example, an anchored predicate path for π^2_{ex} between Barack Obama and Michelle Obama is $\mu_{(C(\texttt{:B.Obama}),C(\texttt{:M.Obama}),\pi^2_{ex})} = \texttt{:Person} \xrightarrow{\texttt{:occupation}} \xleftarrow{\texttt{:occupation}} \texttt{:Person}$. While this anchored path exists between the Obamas, it might not be helpful when determining the veracity of triples with the predicate `:spouse`, as having the same occupation may not be a viable predictor for the veracity of `:spouse` relation. Ergo, discriminative paths are defined.

Definition 2.20 (Discriminative path)**.** The discriminative paths $\Pi^k_{(C(s),C(o))} \subseteq \mathcal{A}_{(k,C(s),C(o))}$ are those anchored predicate paths of length k that encapsulate the relationship between $C\,(s)$ and $C\,(o)$ [112].

In our running example, one of the given assertions is $\texttt{:B.Obama} \xrightarrow{\texttt{:spouse}} \texttt{:M.Obama}$. Some discriminative paths $\Pi^2_{(\texttt{:Person},\texttt{:Person})}$ related to a given triple are: (1) $\texttt{:Person} \xrightarrow{\texttt{:child}} \xleftarrow{\texttt{:child}} \texttt{:Person}$, (2) $\texttt{:Person} \xrightarrow{\texttt{:president}} \xrightarrow{\texttt{:firstLady}} \texttt{:Person}$.

Corroborative paths of a property p are those anchored predicate paths that are used to corroborate the existence of a triple with the predicate p between two resources [122]. Given an RDF knowledge graph, approaches based on corroborative paths assume the existence of an RDFS class hierarchy for the said graph (defined via the `rdfs:subClassOf` predicate). Consequently, these approaches can derive the following important condition on $\mathcal{A}_k$ which are to corroborate

[6]The definition in [112] defines p_i as "the predicate" of the directed edge that connects two vertices. We changed this to "the predicate of a directed edge" since there might be several directed edges with different predicates connecting two vertices.

the correctness of $s \xrightarrow{p} o$: *Only anchored predicate paths in $\mathcal{A}_{(k,D(p),R(p))}$ can corroborate assertions with the given predicate p* [122]. Anchored predicate paths use only the class subsumption hierarchy (i.e., $C(s)$ and $C(o)$) to measure the strength of the association between paths and predicates. However, corroborative paths also exploit domain and range information, i.e., $D(p)$ and $R(p)$, in conjunction with RDFS axioms in the input knowledge graph $\mathcal{G}$ to determine whether a path π_k corroborates a predicate p or not.

Definition 2.21 (Corroborative path)**.** Given an RDF knowledge graph $\mathcal{G}$, *corroborative paths* of length k are paths that carry similar information as p [122]. We define the set of *corroborative paths* for a predicate p formally as:

$$\Omega^k(p) = \bigcup_{j=1}^{k} \mathcal{A}_{(j,D(p),R(p))} \,. \tag{5}$$

In our running example (shown in Figure 1), corroborative paths of the assertion `:B.Obama` $\xrightarrow{\texttt{:spouse}}$ `:M.Obama` are paths between persons, as `:Person` is both the domain and range of the predicate `:spouse`. For example, it will also include paths between `:G.W.Bush` and `:L.Bush`, and may also include paths between `:M.Obama` and `:L.Bush`, because they are also in the domain and range of the predicate (`:spouse`) [66].

2.3 Rule and graph-pattern -mining-based approaches

The following notation is used in rule-mining-based and graph-pattern-mining-based approaches for fact checking. Mining is the process of searching for patterns in the data. For example, if person A is the spouse of person B, then usually B is also the spouse of A (symmetry of marriage). We write such rules using the syntax of first-order logic [118]. For example, we would write the example rule as:[7]

$$(\texttt{?a} \xrightarrow{\texttt{:spouse}} \texttt{?b}) \Rightarrow (\texttt{?b} \xrightarrow{\texttt{:spouse}} \texttt{?a}). \tag{6}$$

Such rules have several applications: First, they can help to complete $\mathcal{G}$. For example, if we know that `:B.Obama` $\xrightarrow{\texttt{:spouse}}$ `:M.Obama`, then we can infer that it is likely that `:M.Obama` $\xrightarrow{\texttt{:spouse}}$ `:B.Obama` also holds. Second, those frequent rules give us insight about our data, biases in the data, or biases in the real world [118].

Recent work focuses on discovering such rules automatically in the data. The components of a rule are called atoms [118].

Definition 2.22 (Atom)**.** In rule-mining approaches, an atom is an expression of the form $s \xrightarrow{p} o$. An atom is instantiated if the subject or the object is bound, but not both subject and object are bound. Note that the predicate of an atom is always bound. Hence, `:B.Obama` $\xrightarrow{\texttt{:livesIn}}$ `?x`, `?z` $\xrightarrow{\texttt{:livesIn}}$ `:Berlin` and `?x` $\xrightarrow{\texttt{:spouse}}$ `?y` are atoms. Like in SPARQL, we denote unbound variables with small letters and a question mark as a prefix. If both subject and object are bound values, the atom is grounded, and it is an assertion [41]. For example, `:B.Obama` $\xrightarrow{\texttt{:spouse}}$ `:M.Obama` is a grounded atom.

A (conjunctive) query is a conjunction of atoms: $B_1 \wedge \cdots \wedge B_n$. A substitution function $\sigma(\cdot)$ is a partial mapping from unbound variables to bound values. Substitutions can be extended to atoms, conjunctions and rules. A result of a query $B_1 \wedge \ldots \wedge B_n$ on $\mathcal{G}$ is a substitution function $\sigma(\cdot)$ that (1) maps all unbound variables and (2) that entails $\forall i \in [1, n]\ \sigma(B_i) \in \mathcal{G}$ [41].

[7] We use the infix notation to be consistent with Section 2.2.

Definition 2.23 (Horn rule)**.** A (Horn) rule R is a formula of the form $\mathcal{B} \Rightarrow \mathcal{H}$, where $\mathcal{H}$ is the head atom and $\mathcal{B}$ is the (conjunctive) query of atoms $B_1, \ldots, B_n$. $\mathcal{B}$ is called body of the rule [41].

Approaches like `Tracy` [39] and `AMIE` [40, 41, 66] use Horn rules for fact checking over $\mathcal{G}$ using predictions.

Definition 2.24 (Prediction)**.** Given a rule $R = B_1 \wedge \ldots \wedge B_n \Rightarrow H$ and a substitution function $\sigma(\cdot)$, we call $\sigma(R)$ an instantiation of R. If $\forall i \in [1, n]\ \sigma(B_i) \in \mathcal{G}$, we call $\sigma(H)$ a prediction $\mathcal{P}$ of R from $\mathcal{G}$, and we write $\mathcal{G}\ \wedge R \models \mathcal{P}$. If $\sigma(H) \in \mathcal{G}$, we call $\mathcal{P}$ a true prediction [66].

To exemplify, we can predict that `:B.Obama` $\xrightarrow{\texttt{:spouse}}$ `:M.Obama` using the following rule:

$$(\texttt{?x} \xrightarrow{\texttt{:child}} \texttt{?z}) \wedge (\texttt{?y} \xrightarrow{\texttt{:child}} \texttt{?z}) \Rightarrow (\texttt{?x} \xrightarrow{\texttt{:spouse}} \texttt{?y}). \tag{7}$$

Most of the rule-mining-based approaches are supervised [40, 41, 66]. Therefore, these approaches need counterexamples for the training data. These counterexamples as well as assertions from $\mathcal{G}$ are used to calculate the support and confidence of the given assertion. The support and confidence of rules or patterns that assist a given assertion are used in all rule and pattern mining approaches.

Definition 2.25 (Support)**.** The support $\mathcal{S}(R)$ of a rule R in a knowledge graph $\mathcal{G}$ is the number of true predictions $\mathcal{P}$ that the rule makes [66]:

$$\mathcal{S}(\mathcal{R}) = |\{\mathcal{P} : (\mathcal{G}\ \wedge \mathcal{R} \models \mathcal{P}) \wedge \mathcal{P} \in \mathcal{G}\}|. \tag{8}$$

Definition 2.26 (Confidence)**.** The confidence $C(R)$ of a rule R in a knowledge graph $\mathcal{G}$ is the proportion of true predictions out of the total number of predictions:

$$C(\mathcal{R}) = \frac{\mathcal{S}(\mathcal{R})}{\mathcal{S}(\mathcal{R}) + |\{\mathcal{P} : (\mathcal{G}\ \wedge R \models \mathcal{P}) \wedge \mathcal{P} \in \Gamma^{-}(\mathcal{R})\}|}. \tag{9}$$

Here in equation 9, $\Gamma^{-}(\mathcal{R})$ denotes the set of counterexamples of R. If the counterexamples are chosen under `PCWA`, we refer to $C(R)$ as the `PCWA` confidence and denote it by $C_{\texttt{PCWA}}(R)$ (analogously for the `CWA` or other approaches) [66].

These counterexamples are generated in most of the rule-mining-based or supervised mining methods under `PCWA`. A special form of `PCWA` is weak `PCWA` (`WPCA`). The description of resources and the similarity between resources are used in `WPCA`, which are defined as follows.

Definition 2.27 (Description of resource)**.** Let L be a function, which encodes the content of entities v_i (respectively edges e_i), such as types, properties, or names (respectively, relations or actions) as found in $\mathcal{G}$ [78].

In our running example, we may define the description of node $L($`:R.Party`$)$ as "Republican Party". These descriptions are used for approximate graph pattern matching. $L_{\aleph}(\cdot)$ is used to encode the graph pattern's resources, while $L(\cdot)$ is used to encode $\mathcal{G}$'s resources.

Definition 2.28 (Graph pattern)**.** A graph pattern $\aleph(v_s, v_o)$ contains a set of pattern nodes $V_{\aleph}$ and pattern edges $E_{\aleph}$[8]. From the variables in the graph patterns, two designated nodes are called anchored nodes v_s and v_o. These variables have the given labels $L_{\aleph}(v_s)$ and $L_{\aleph}(v_o)$, respectively. For simplicity, we will refer to $\aleph(v_s, v_o)$ as $\aleph(s, o)$ [78].

Specifically, a triple pattern $v_s \xrightarrow{p} v_o$ is a graph pattern that contains a single pattern edge with two pattern nodes with labels $L_{\aleph}(v_s)$ and $L_{\aleph}(v_o)$, respectively. The similarity between two resource labels in $\mathcal{G}$ and $\aleph(s, o)$, respectively, is defined as follows.

[8] "A graph pattern $\aleph(v_s, v_o)$ is a set of basic graph patterns. A Basic Graph Pattern is a set of Triple Patterns" [104]

Definition 2.29 (Similarity function)**.** Given two descriptions L and L', we make use of a description similarity predicate to verify if L and L' are similar for a similarity threshold α, denoted as $L \sim_\alpha L'$.

Definition 2.30 (Graph pattern matching)**.** Let V and E represent nodes and edges of $\mathcal{G}$, respectively. $V_{\aleph}$ and $E_{\aleph}$ represent the nodes and edges in $\aleph(s, o)$, respectively. Given a description similarity predicate, it extends the approximate pattern matching [81] as a matching relation $R \subseteq V_{\aleph} \times V$ between pattern $\aleph(s, o)$ and $\mathcal{G}$, under similarity threshold α. For each node pair, $(u, v) \in R$, the following three requirements should hold:

(1) $L(u) \sim_\alpha L(v)$,
(2) for every edge $e = (u, u') \in E_{\aleph}$, there exists an edge $e' = (v, v') \in E$ such that $L(u') \sim_\alpha L(v')$, and $L(e) \sim_\alpha L(e')$, and
(3) for every edge $e = (u'', u) \in E_{\aleph}$, there exists an edge $e'' = (v'', v) \in E$ such that $L(u'') \sim_\alpha L(v'')$, and $L(e) \sim_\alpha L(e'')$ [78].

Definition 2.31 (Weak partial closed world assumption (WPCA))**.** Given an assertion $s \xrightarrow{p} o \in \Gamma^+_p$, where Γ^+_p are those assertions in Γ^+ that contain the predicate p. We say that a statement $s \xrightarrow{p} o'$ is false under WPCA witnessed by a statement $s \xrightarrow{p} o$, iff $L(o) \nsim_\alpha L(o')$. We remark that PCWA is a special case of WPCA; however, if $L(o) \neq L(o')$ is enforced, WPCA is equivalent to PCWA [78].

To exemplify, consider a statement $t_1 =$:G.W.Bush $\xrightarrow{\text{:birthPlace}}$:Connecticut where $L($:Connecticut$)$ is "state". Let $t_2 =$:G.W.Bush $\xrightarrow{\text{:birthPlace}}$:NewHaven $\notin \mathcal{G}$ be another statement and let $L($:NewHaven$)$ be "city" instead of "state". Here, the missing statement t_2 is considered to be false under PCWA (due to a witness t_1). However, WPCA excludes it from Γ^-, given that "city" and "state" are close resources (i.e., $\leq \alpha$).

Definition 2.32 (Graph fact checking rule)**.** A graph fact checking rule (denoted as GFC) is in the form of $\varphi : \aleph(L(s), L(o)) \Rightarrow q_s \xrightarrow{p} q_o$, where (1) $\aleph(L(s), L(o))$ is a graph pattern and (2) $q_s \xrightarrow{p} q_o$ is a triple pattern (not in $\aleph(s, o)$) carrying the same pair of anchored nodes as $\aleph(L(s), L(o))$ [78].

Approaches such as graph fact checking rules (GFC) [76] and ontological graph fact checking rules (OGFC) [77] mine the subgraph patterns from $\mathcal{G}$ and use them for fact checking.

A GFC φ states that a pattern $\aleph(s, o)$ covers an assertion $s \xrightarrow{p} o$ in $\mathcal{G}$ if s and o match its anchored nodes $L(s)$ and $L(o)$, respectively. In other words, given a new assertion $t = s_1 \xrightarrow{p_1} o_1$ not in $\mathcal{G}$, φ states that t should exist in $\mathcal{G}$ if it is an instance (match) of $q_{s_1} \xrightarrow{p_1} q_{o_1}$, and s_1 and o_1 are also matches of $\aleph(s_1, o_1)$ [78]. The GFC-based approach uses the graph patterns to validate assertions.

3 SURVEY METHODOLOGY

We mainly rely on the approach to systematic surveys proposed in [59, 89]. We also include a description of related surveys for the sake of completeness.

3.1 Related Surveys

In the literature, researchers focus on the automatic fact-checking task stemming from natural language processing tasks [49, 143, 145] (i.e., claim detection, evidence retrieval, verdict prediction, and justification production). Hence, these surveys are disjoint from our work, as we target on summarizing existing fact-checking methods over knowledge graphs.

To the best of our knowledge, until date, there exists no comprehensive peer-reviewed survey on fact checking over knowledge graphs. Neema et al. [64] present a survey of automated fact-checking approaches that provide explanations, i.e., reasons for their predictions. Table 3 shows that their survey has a limited coverage compared to our work. We aim to fill this gap. Our survey covers 27 approaches and 35 benchmarks.

Table 3. Overview of surveys on fact checking; the abbreviations are: TB/Text-based, PB/Path-based, RMB/Rule-mining-based, EB/Embedding-based, App/Approaches, and Bench/Benchmarks.

Study	Year	Techniques					Datasets	App	Bench
		TB	PB	RMB	EB.	Hybrid			
Neema et al. [64]	2020	✗	✓	✗	✗	✗	✗	5	5
Qudus et al.(ours)	2023	✓	✓	✓	✓	✓	✓	27	35

3.2 Methodology

In this section, we present our methodology for the literature review on fact checking over knowledge graphs [59, 89].

3.2.1 Research questions. We aim to answer the following research questions:

RQ_1 : Which categories of fact-checking approaches exist (e.g., based on common implementation templates) and what are their characteristics?
RQ_2 : Which challenges are associated with fact checking?
RQ_3 : Which trends can be observed across fact-checking approaches over the time frame of the survey?
RQ_4 : What is the (relative) performance of representative fact-checking approaches?

3.2.2 Search Strategy. The search strategy for this systematic survey was iterative and was run separately by the first two authors to avoid a bias and ensure an appropriate coverage of related articles. Based on the research questions above, each author identified terms that seemed most appropriate for this systematic review. These include "fact checking", "fact-checking", "truth finding", "fact validation", "fact confidence", and "fact verification". We used those keywords to derive the following queries:

- `intitle:`(`fact checking` OR `fact-checking` OR `fact verification` OR `fact validation` OR `fact confidence` OR `truth finding`) AND (*using* OR *over*) AND (*knowledge graphs* OR *corpora* OR *both: corpora and knowledge graphs*)
- (*fact checking* OR *fact-checking* OR *fact verification* OR *fact validation* OR *fact confidence*) AND (*using* OR *over*) AND (*knowledge graphs* OR *corpora* OR *both: corpora and knowledge graphs*)

The `intitle` tag was used to specify that the selected keywords should be present in the title of the paper, where applicable. As suggested by Moher et al. [59, 89], searching in the title alone does not always provide us with all relevant publications. Ergo, we do not restrict ourselves to searching the keywords in the title only. In the second query we include the abstract and the remaining text of the publication in the search space.

3.2.3 Search databases. With the keywords defined above, we searched for publications in the following list of search engines, digital libraries, journals, conferences, and their respective workshops:

- Google Scholar [9]
- IEEE Xplore Digital Library (IEEE Xplore) [10]
- ACM DL [11]
- Science Direct [12]
- Semantic Web Journal (SWJ) [13]
- Journal of Web Semantics (JWS) [14]
- Journal of Data and Knowledge Engineering (JDWE) [15]
- International Journal on Semantic Web and Information Systems (IJSWIS) [16]

3.2.4 Inclusion criteria. Our inclusion criteria were as follows:

- Publications in English between 2014 and the first half of 2023.
- Algorithms and frameworks that focus on fact checking over knowledge graphs using (1) knowledge graphs, (2) corpora, or (3) both corpora and knowledge graphs as background knowledge.

3.2.5 Exclusion criteria. We did not include publications in the current survey if they fulfilled one of the following criteria:

- Publications that were not peer-reviewed.
- Assessment methodologies that were published as a poster or abstract.
- Fake news identification approaches that are related but aim at a different goal.
- Fact checking for unstructured web data and structured relational data.
- Studies that did not propose any methodology or framework for fact checking.

3.2.6 Search methodology steps. To conduct our systematic literature search for fact-checking approaches, we applied a seven-step search methodology:

(1) Apply keywords to the search engine using the time frame 2014–2023.
(2) Scan article titles based on our criteria.
(3) Import the output from step 2 to a reference manager software to remove duplicates[17].
(4) Review abstracts based on our criteria.
(5) Analyze the selected articles based on our criteria and the research questions.
(6) Retrieve new papers from the list of references cited by the papers of step 5.
(7) Scan the references from the survey papers that passed steps 5 and 6 and retrieve new papers that fulfill the criteria.

Additionally, we added any new survey that we discovered during our search methodology steps to the results of the related surveys in Section 3.1. Table 4 provides an overview of the number of papers retrieved in each phase of the search methodology. Due to the large amount of results obtained from Google Scholar, ACM DL, and Science Direct, step 2 was applied to the top-1000 relevant results.

[9] http://scholar.google.com/
[10] https://ieeexplore.ieee.org/Xplore/home.jsp
[11] http://dl.acm.org/
[12] http://www.sciencedirect.com/
[13] http://www.semantic-web-journal.net/
[14] http://www.websemanticsjournal.org/
[15] http://www.journals.elsevier.com/data-and-knowledge-engineering/
[16] http://www.ijswis.org/
[17] Here, we used Mendeley 1.19.3 http://www.mendeley.com/ as it is free and has the required deduplication functionality.

Table 4. Number of retrieved related papers for each phase of the search methodology

Search Engines	Steps 1	2	3	4	5	6	7
Google Scholar	35,900	293	280	23	13	14	18
IEEE Xplore	6	1	0	0	0	0	0
ACM DL	1762	156	15	2	2	2	3
Science Direct	35,396	110	17	5	5	6	6
SWJ	50	1	0	0	0	0	0
JWS	3	1	0	0	0	0	0
JDWE	0	0	0	0	0	0	0
IJWIS	0	0	0	0	0	0	0
Total	45,297	562	312	33	20	22	27

4 CATEGORIZATION

To answer the first research question, we divided the fact-checking approaches into categories. We applied two different category schemes: hierarchy and coloring (see Figure 2). First, fact-checking approaches can be divided into two major categories based on the reference knowledge the approach relies on. Some approaches use structured knowledge [43, 44, 122], while others use textual knowledge [45, 123] to check the veracity of assertions. Figure 2 gives an overview of these two major categories, their subcategories and publications within the single categories. Second, the different approaches can be either supervised or unsupervised. We used colors to mark the approaches of these two categories in Figure 2.

4.1 Fact checking using textual reference knowledge

Approaches in this category validate a given assertion by searching for evidence in the textual references. There are mainly two types of textual reference knowledge sources that are used—web text corpora and local text corpora.

4.1.1 Fact checking using a web corpus. Fact checking using a web corpus requires access to a web search engine. Search engines like Bing[18] or Google[19] provide this access through their web application programming interface (API). The most popular API among fact-checking approaches is the Bing search API[20]. Different approaches have been proposed for validating an input assertion using the web corpus, such as T-verifier[21] and DeFacto [45, 74]. In contrast to other approaches, DeFacto (Deep Fact Validation) uses a combination of textual evidence and machine learning to evaluate a given assertion. It relies on word proximity analysis [100] combined with a string similarity search in the web corpus to determine the veracity of a given assertion [45].

4.1.2 Fact checking using a local corpus. Some fact-checking approaches use a local corpus as reference knowledge, e.g., FactCheck [123] uses Elasticsearch[22] to index all articles in the English Wikipedia and search for textual evidence. Moreover, FactCheck calculates the confidence of a given assertion and the document's trustworthiness to determine whether the given assertion is true or false [123]. To address Defacto's drawback (i.e., string matching and string similarity search [45]), FactCheck takes into account the sentence coherence features.

[18] https://www.bing.com/
[19] https://www.google.com/
[20] https://azure.microsoft.com/en-us/services/cognitive-services/bing-web-search-api/
[21] T-verifier is excluded because it is older than the timeframe of this survey.
[22] https://www.elastic.co/

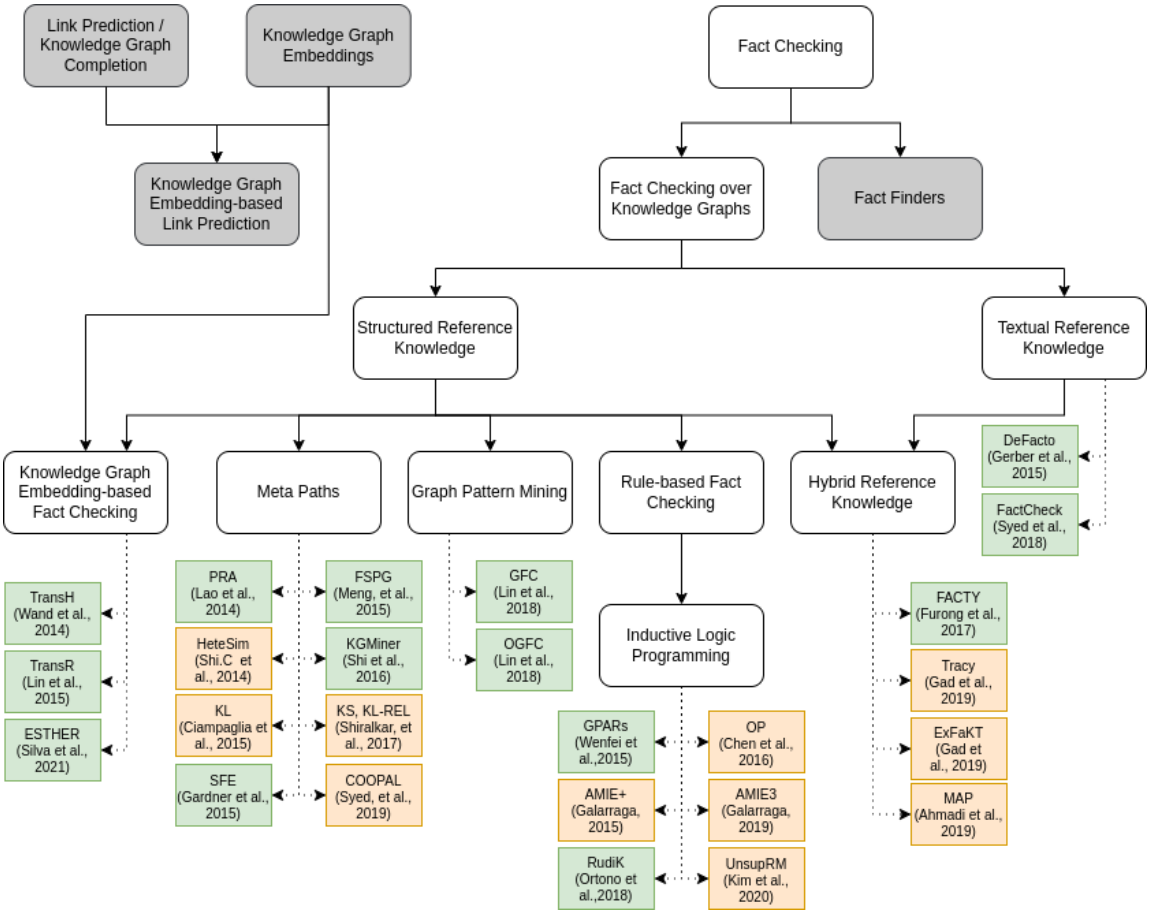


Fig. 2. Categorization of fact-checking approaches. Supervised and unsupervised approaches are marked orange (●) and green (●), respectively. Categories that are related but excluded from the survey are colored gray (●).

4.2 Fact checking using structured reference knowledge

Approaches in this category validate a given assertion using a knowledge graph as reference knowledge. Some of the approaches, i.e., link prediction and knowledge graph completion, listed in this category serve a different but related goal and can also be used to perform fact checking [43, 44, 137]. We further categorize these approaches based on their methodology (see Figure 2).

4.2.1 Path-based approaches. Path-based approaches are generally used to validate the input assertion by traversing one or several paths from the subject to the object node of the given assertion. These approaches validate the input assertion by exploiting the related knowledge represented in the knowledge graph. We further categorize them based on the type of path they use.

- Shortest specific path-based approach: Ciampaglia et al. [24] propose Knowledge Linker (`KL`), an approach that relies on the shortest and most specific path between s and o. The veracity of the given assertion is calculated based

on the length and the specificity of the respective path (see Figure 5a). Shiralkar et al. [114] propose an improved version of KL by introducing semantic biases in the search for the shortest specific path. The approach is called Relational Knowledge Linker (KL-REL).

- Path enumeration-based approaches: Approaches within this category gather all paths π_k of length k between s and o. The fact-checking task is performed by calculating a confidence score for each path. After computing the confidence score of each path, a final veracity score is computed by taking into account the most relevant meta paths based on some measure, e.g., similarity. However, approaches such as PathSim [121], PC [120], HeteSim [113], PRA-basic [67], and PCRW [68], do not only need human-annotated paths. They also have strict constraints on input paths, e.g., paths with specific types of relations only. Some of these approaches (such as PCRW [68]) require exhaustive enumeration of all paths and are very sensitive to the length of paths [67]. In addition, approaches of this category are impractical on large-scale knowledge graphs due to the exhaustive number of paths. Within this survey, we exclude PC, PRA-basic, PathSim, and PCRW from our further discussion because they are older than our selected time frame.
- Anchored predicate path mining-based approaches: In simple meta paths, the presence of entity class information may sometimes be redundant because it can be inferred from the predicates on the path [112]. KGMiner [112] and FSPG [85] use anchored predicate paths A^k, which only contain class information of anchored nodes and the properties of the path. In KGMiner, the most discriminative paths ($\Pi^k_{(C(v_s),C(v_o))}$) are selected from all possible A^k as input to a logistic regression model [112]. The FSPG approach selects only those anchored predicate paths A^k that have the highest similarity between the given node pair (s and o) and A^k using a similarity scoring function. With an increasing size of the knowledge graph, the number of meta paths increases exponentially [68]. Identifying and selecting all meta paths is a challenging task [5, 55]. Consequently, the following approaches are built to select the most appropriate meta paths between the subject and object of a given triple.
- Corroborative path-based approach: Anchored predicate path-based approaches make use of class subsumption information. COPAAL [122, 125] exploits further RDFS semantics of the reference knowledge graph ($\mathcal{G}$) using domain and range information of the given assertion's predicate. It uses corroborative paths, which are further restricted to start and end with entities that fulfill the predicate's domain and range restrictions, respectively.
- Path ranking-based approaches: Approaches of this category use meta paths between s and o as features in a statistical model. Based on this model, they rank meta paths and determine a final veracity score. A basic example of such an approach is PRA [44]. It generates a training set by performing two-sided-random walks starting from the source and target nodes without any constraints. The best paths are selected as features and stored in a feature matrix. For a given assertion $s \xrightarrow{p} o$, any value in the matrix is calculated by finding the probability of reaching the target node after a random walk from the source node on one of the selected paths. PRA approximates the probability using rejection sampling to reduce the computational complexity [56]. The input assertion is then validated using the feature matrix. SFE extends PRA by extracting features from subgraphs of the knowledge graph [43]. Given a parameter m, the subgraph of depth m for each node n_i is the result of m breadth-first search steps starting at n_i. A meta path that connects a source node to a target node is obtained by the intersection of the subgraphs of the source and target at a common node. SFE uses such meta paths to identify binary features. These features are then used in a classifier trained on positive and negative examples [43].
- Network-flow-based approach: Shiralkar et al. [114] propose a network-flow-based unsupervised approach called Knowledge Stream (KS). This approach transforms the fact-checking problem into a minimum cost maximum

flow problem. Each edge of the graph is associated with two quantities: a cost of usage and a capacity to carry knowledge related to $s \xrightarrow{p} o$ across its two endpoints [114]. The final veracity score is based on the scores of the best paths between s and o.

4.2.2 Rule-mining-based approaches. Rule-mining-based approaches extract association rules to perform the fact-checking or fact-prediction task on the knowledge graph [40, 41, 66]. We categorize these approaches based on the type of methodology they adapt to perform the fact-checking task.

- Inductive logic programming (ILP)-based rule-mining approaches: ILP-based approaches make use of training examples. Training examples consist of positive examples and counterexamples. The task of learning rules from these examples is called ILP. ILP-based approaches are used in database literature [47, 91, 106] and are also known as the first generation of rule-mining approaches. However, these approaches are not suitable for large-scale knowledge graphs because of the `OWA`. The `OWA` implies that the knowledge graphs do not store the counterexamples (see Definition 2.12), which are required in ILP-based approaches. WARMeR generates the counterexamples by assuming the knowledge graph is complete [47]. In contrast, Meng et al. [91] propose a method to generate the counterexamples from random facts using a positive-only learning function. This approach is used for ILP on knowledge graphs in a more systematic way by Suchanek et al. [118]. Galárraga et al. elaborate that these ILP-based approaches are not designed for knowledge graphs and present the `PCWA` [41]. `AMIE` and `AMIE+` are the first approaches in rule mining, which explicitly target large-scale knowledge graphs [40, 41, 119]. These approaches mine association rules from the knowledge graph in the form of conjunctive horn clauses. In rule-mining-based models, rules state that the instance of an assertion exists if it satisfies the description of the pattern. Graph-pattern association rules (`GPARs`) on the other hand discover association rules in the form of $\varphi \Rightarrow s \xrightarrow{p} o$, with a subgraph pattern φ and a single assertion $s \xrightarrow{p} o$ not in φ [36]. `GPARs` predict the edges via frequent subgraph pattern mining. These methods are faster than the previous ILP-based approaches, but they can still take hours, or even days, on large-scale knowledge graphs (see Table 14). In the following, we discuss the state-of-the-art optimization techniques to further speed up the rule-mining process.
  - Parallelism-based rule mining: The ontological path-finding approach (OP) uses a cluster computing framework (i.e., SPARK[23]) to compute the confidence (C) and support ($\mathcal{S}$) of atoms to speed up the rule mining process [22, 23]. To scale to large knowledge graphs, Chen et al. [22] parallelize the mining algorithm by dividing the input knowledge graph into smaller subgraphs. Afterward, the efficient rule mining algorithm extracts inference rules from subgraphs using parallel in-memory join queries.
  - Non-exhaustive rule mining: `RuDiK` is another rule-mining-based approach that uses the partial closed world assumption to generate explicit semantically related counterexamples [95]. `RuDiK`'s strategy, to find the best rules that are necessary to predict the true training assertions, is based on a greedy approach that at each step adds the most suitable rule to the output set. This strategy is different from other exhaustive rule mining approaches [36, 40, 41, 66]. The goal of the approach is to find the best possible rules for better predictions, not all rules above a given threshold value. This non-exhaustive nature leads to a better runtime than `AMIE` [41] or `AMIE+` [40]. But `AMIE-3.0` [66] outperforms `RuDiK` in terms of runtime due to its optimization and pruning strategies, such as lazy evaluation, parallel computation and others (see Section 5 for details).
- Unsupervised rule-mining-based approach: The majority of the state-of-the-art rule-mining-based approaches rely either on positive or negative rules for link-prediction or fact-checking tasks. However, Kim et al. [58]

[23] https://spark.apache.org

propose KV−Rule, an unsupervised approach that considers both negative and positive rules to predict the missing assertions in a knowledge graph. They also propose a distant partial closed world assumption (D−PCWA) for the generation of counterexamples [58], as explained in Section 6.3.

4.2.3 Graph pattern-mining-based approaches. In this category of approaches, models and pattern discovery methods are proposed that explicitly incorporate discriminant graph patterns to support fact checking over knowledge graphs [76]. These patterns are used to construct classifiers based on rules or latent features. While rule-mining-based models are easy to interpret, they come with several drawbacks. For example, rule-mining-based models lack the necessary expressiveness to explicitly describe subgraphs as discriminant features [76]. Rules cover only a subset of useful patterns [93], and they are expensive to find using subgraph isomorphism. To overcome these drawbacks, Lin et al. [76] propose the following two approaches.

- Graph Fact Checking rules (GFC): In a GFC-based approach, a set of approximate patterns GFCs are mined from the knowledge graph. Afterward, it dynamically selects the useful GFCs from the GFCs set to perform the task of fact checking [76, 78]. For example, a given assertion is true if it exists in the selected GFC φ from the knowledge graph. The GFC approach is a supervised fact-checking approach that uses the approximate pattern mining approach instead of subgraph isomorphism. Subgraph isomorphism-based approaches (such as GPARs [36]) produce redundant patterns that are hard to compute compared to the GFC approach. The GFC can also be used as a standalone rule model like in AMIE, AMIE+ or AMIE−3.0 [40, 41, 66], but not vice versa. The GFC approach generates counterexamples by considering the PCWA.
- Ontological Graph Fact Checking rules (OGFC): Ontology-based pattern matching was introduced to replace exact label matching [113] with an ontology-based similarity function [72, 140]. In the ontology-based similarity function, OGFC's authors tweak the similarity function defined in Definition 2.29 to find semantically close labels within the ontology. The OGFC approach [77] generalizes the GFC approach [76] by incorporating both topological constraints and ontological closeness to better distinguish between true and false assertions. Unlike GFC, this approach proposes a unified model for multiple types of assertions with semantic closeness. Similar to GFC, OGFC also generates counterexamples by considering the PCWA.

4.2.4 Embedding-based approaches. Embeddings encode entities and relations in the knowledge graph into a low-dimensional vector space while preserving certain information of the graph and minimizing a margin-based ranking loss [16, 57, 79, 137]. Multiple variants of embedding-based algorithms exist. These approaches use embedding features for tasks, such as knowledge graph completion or link prediction [16, 30, 57, 130]. However, only three approaches are included in our survey. TransH [137] and TransR [79] use their embedding models to tackle fact checking. ESTHER [27]) makes use of the continuous representation of $\mathcal{G}$ provided by an embedding model to search for paths. These paths can be used in combination with one of the aforementioned path-based approaches (e.g., COPAAL [124]) to improve the performance of the fact-checking task [27].

4.3 Hybrid fact-checking approaches

FACTY, ExFaKT and Tracy are hybrid approaches that exploit structured and textual reference knowledge to find the veracity of a given fact [38, 39, 71]. ExFaKT and Tracy use rules mined from the knowledge graph and search for evidence on the web and knowledge graph using those rules. FACTY leverages textual reference and meta-path-based techniques to find supporting evidence for each triple, and subsequently predicts the correctness of each triple based on

the evidence [71]. Ahmadi et al. also propose a hybrid approach (dubbed as MAP) to discover assertions using answer set programming (ASP) [1]. Rules are automatically discovered from the knowledge graph using an existing rule-mining approach, e.g., [95]. Then, those rules are used for web-mining to support the task of fact checking. The set of rules is turned into a logical program, and the fact-checking task is modeled as an inference problem in a probabilistic extension of ASP. In ASP, search problems are reduced to computing stable models (aka. answer-sets), which are a set of beliefs described by the program. In a horn program, the stable models are the same as the minimum models. However, they can differ when the program allows for more expressive language structures like defaults, negations, and aggregates [1]. Most of the approaches above make use of at most two categories of fact-checking approaches. HybridFC is the only approach that makes use of three categories of approaches, i.e., text-based, embeddings-based, and meta-path-based, in an ensemble learning setting [105].

Table 5. Path-based approaches. The abbreviations are Biased Path Constrained Random Walk (BPCRW), Forward Selection Algorithm (FSA), Breadth-first search (BFS), not applicable (NA), term frequency–inverse document frequency (TF-IDF), logistic regression (LR), shortest specific path (SSP), Biased SSP (BSSP), Machine learning method (ML) [61].

Approach	Path-type	Length	Path generation	Path filtering	ML
HeteSim [113]	pairwise random walk	variable	manual	similarity relevance	NA
PRA [44]	random walk	bound	automatic	rejection sampling	LR
FSPG [85]	BPCRW	variable	semi-automatic	FSA (GreedyTree)	NA
SFE [43]	path sampling	bound	automatic	subgraph-based BFS	LR
KL [24]	SSP	variable	automatic	specificity	NA
KL-REL [114]	BSSP	variable	automatic	contextual biases	NA
KGMiner [112]	discriminative paths	bound	automatic	Jaccard coefficient	LR
KS [114]	network-flow	variable	automatic	TF-IDF	LR
COPAAL [122]	corroborative paths	bound	automatic	$D(p)$ and $R(p)$	NA

Table 6. Path-based approaches. The abbreviations are Forward Stage-wise Path Generation algorithm (FSPG), not applicable (NA), open source not available (ONA), Path Constrained Random Walk (PCRW), Normalized pointwise mutual information (NPMI), Neighborhood scoring (NS), Path counting (PC), shortest specific path (SSP), Biased SSP (BSSP), ontology features (OF), unsupervised (unsup.), supervised (sup.).

Approach	Knowledge Graph(s)	OF	Learning	Formulation	Veracity calculation	implementation	Semantics
HeteSim [113]	ACM and DBLP	NA	sup.	$\mathcal{G}$-completion	pairwise random walk	ONA	no
PRA [44]	PubMed	C	sup.	$\mathcal{G}$-completion	Probability based PCRW	JAVA+Scala	no
FSPG [85]	DBLP, Yago	C	sup.	fact checking	NS+PC	ONA	no
SFE [43]	NELL	C,D,R	sup.	$\mathcal{G}$-completion	subgraphs feature set	JAVA	no
KL [24]	DBpedia	NA	unsup.	fact checking	SSP	ONA	no
KL-REL [114]	DBpedia	NA	unsup.	fact checking	BSSP	Python	yes
KGMiner [112]	Wikipedia, PubMedDB	C	sup.	fact checking	information gain	C++	yes
KS [114]	DBpedia	C	unsup.	fact checking	Network Flow	Python	yes
COPAAL [122]	DBpedia	C,D,R	unsup.	fact checking	NPMI [17]	JAVA	yes

5 DETAILED SYSTEM COMPARISON

Six main families of approaches have been devised to address the problem of fact checking over knowledge graphs. In this section, we describe in detail these families that are the outcome of our methodology section.

Table 7. Rule and graph-pattern -mining-based approaches. The abbreviations are indicative logic programming (ILP), graph fact checking (GFC), open-source not available (ONA), ontology aware approximate pattern matching (OAAPM), approximate pattern matching (APM), answer set programming (ASP), negative sampling (NS), Pattern or rules generation (Pattern/rule gen.), DBpedia (DBP), Wikidata (WD), unsupervised (unsup.), supervised (sup.), implementation language (Imp.), Rule-mining (RM), Web-mining (WM), Pattern-mining (PM).

Approach	Formulation	Pattern/rule gen.	Dataset(s)	Learning	Distinct feature	Key features	Imp.
`AMIE` [41]	RM	automatic	Yago,DBP	unsup.	Horn rules	ILP	JAVA
`AMIE+` [40]	RM	automatic	Yago,DBP,WD	unsup.	Horn rules	ILP	JAVA
`GPARs` [36]	Association RM	manual	DBP	sup.	Association rules	$\mathcal{S}$ and C	JAVA
`GFC` [76]	Graph PM	automatic	DBP,Yago,WD,MAG	sup.	APM	top-f φ	JAVA
`OGFC` [77]	Graph PM	automatic	DBP,Yago,WD,MAG	sup.	OAAPM	top-f φ	JAVA
`RuDiK` [95]	Incremental RM	automatic	DBP,Yago,WD	unsup.	-ve rules	NS	JAVA
`MAP` [1]	RM, WM	automatic	DBP	sup.	hybrid approach	ASP	Python
`ExFaKT` [38]	RM, WM	automatic	Yago,DBP,wikipedia	unsup.	Explanations	Hybrid	JAVA
`Tracy` [39]	RM, WM	automatic	Yago,wikipedia	unsup.	Explanations	Hybrid	ONA
`AMIE-3.0` [66]	RM	automatic	Yago,DBP,WD	unsup.	Horn rules	ILP	JAVA
`KV-Rule` [58]	RM	automatic	DBP	unsup.	+ve/-ve rules	NS	JAVA

5.1 Text-based approaches

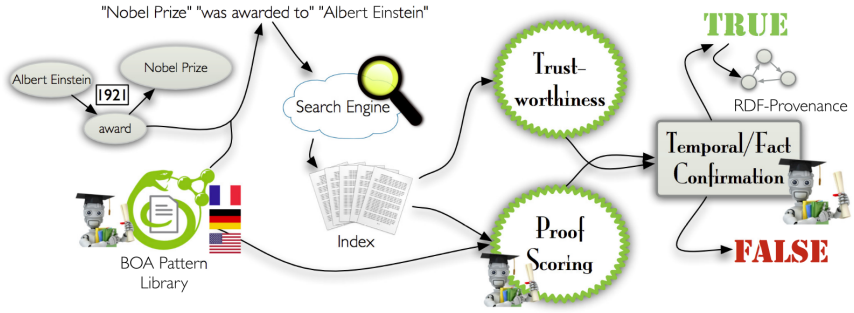
Text-based approaches encompass solutions that verbalize the input assertion and use textual evidence to find statements that support or refute the input assertion. There are mainly two types of textual evidence sources that are used-—web text corpora and local text corpora.

DeFacto [45] is an algorithm that validates statements by finding confirmative sources on the web. These statements can be in the form of an assertion (such as `:A.Einstein` $\xrightarrow{\texttt{:award}}$ `:NoblelPrize`) or a natural language sentence (such as "Nobel Prize was awarded to Albert Einstein"). In the case of natural language statements, an assertion is extracted. The assertion is transformed into several search engine queries using the BOA pattern library [46]. These search engine queries of the assertion are used to search through a web corpus and gather documents containing sentences similar to the respective queries. The search results are analyzed, and snippets of text are extracted that could be used as evidence. From each extracted snippet, several features are gathered. These features are used to assign a score to the snippet using a logistic regression classifier. In the final step, the scores of the pieces of evidence, along with other features, are used as inputs to a classifier that assigns a veracity score to the input statement. Veracity scores and evidence pieces are included in the result of DeFacto [35].

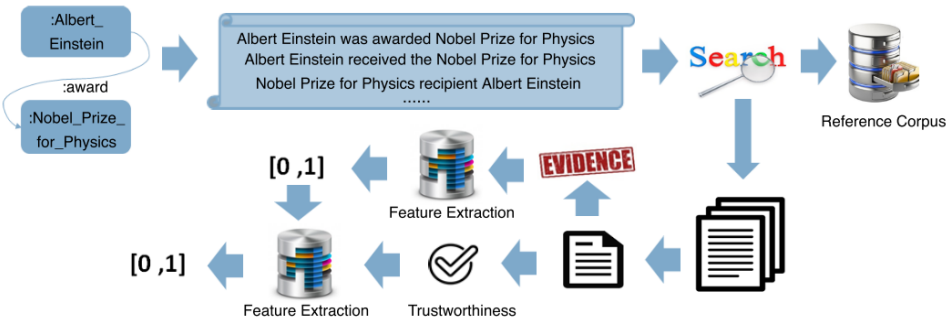
DeFacto relies on string matching and string similarities and does not consider the sentence structure into account. FactCheck [123], an extension of DeFacto, on the other hand, employs sentence coherence features gathered from trustworthy source documents. It combines deep sentence parsing with topic coherence on a local reference corpus to determine how likely an assertion is to be true [123]. Additionally, FactCheck calculates the sentence similarity by constructing a parsing tree and checking whether the subject and the object are connected by a predicate or not. The architectures of FactCheck and DeFacto are shown in Figure 3.

5.2 Path-based approaches

Path-based approaches use a knowledge graph as background knowledge and use the paths contained therein to evaluate the likelihood that a given assertion $s \xrightarrow{p} o$ is true or false. In this section, we provide a systematic comparison of these approaches. Out of 9 recent approaches, 4 are unsupervised and 5 are supervised. Tables 5 and 6 depict different



a **Temporal-DeFacto** Architecture [45]



b **FactCheck** Architecture [123]

Fig. 3. Text-based approaches architectures.

characteristics of these approaches in detail. Other approaches are left intentionally because they are older than the time frame of this survey (e.g., path counting-based [120] and path-constrained random walk-based [68] approaches).

As defined in the previous section, the discriminate predicate path mining approach `KGMiner`, alternatively dubbed Predpath, is a supervised discriminative path-enumeration-based approach [112]. Given an assertion $s \xrightarrow{p} o$, the first step to calculate its veracity is to collect all anchored predicate paths ($\mathcal{A}_{(k,C(s),C(o))}$) between all instances of the subject's class $C(s)$ and instances of the object's class $C(o)$. It extracts the discriminative paths ($\Pi^{k}_{(C(v_s),C(v_o))}$) from $\mathcal{G}$and uses the discovered model to validate the input assertion. Each subject object pair (s_i, o_i), from the set of subjects and objects, represents an alternate anchored predicate path and is explored using depth-first search traversal on the graph up to length m. In the second step, `KGMiner` selects the most discriminative paths from the set of anchored predicate paths. The discriminative paths are selected because some $\mathcal{A}_{(k,C(s),C(o))}$ are not discriminative enough to uniquely encapsulate the predicate p. For example, the path $\texttt{:Person} \xrightarrow{\texttt{:occupation}} \xleftarrow{\texttt{:occupation}} \texttt{:Person}$ is not discriminative enough to uniquely encapsulate the predicate `:spouse`. The resulting discriminative paths are the input to a logistic regression model that computes the veracity of the given assertion.

The disadvantage of the `KGMiner` approach is that it extracts features, searching exhaustively over $\mathcal{G}$. To reduce the complexity of this exhaustive search in large-scale knowledge graphs, a method called Path Ranking Algorithm (`PRA`)

was introduced [44]. PRA learns to rank paths relative to the given assertion $s \xrightarrow{p} o$. It begins by enumerating a large set of anchored predicate paths $\mathcal{A}_{(k,C(s),C(o))}$ using random walks starting from an instance of $C(s)$ as s_i and ending at an instance of $C(o)$ as o_i, or vice versa. In Figures 4 and 5, U and V represent the set of instances of $C(s)$ and $C(o)$, respectively. The intermediate nodes $(n_1 \dots n_k)$ are the nodes between U and V. PRA uses $\mathcal{A}_{(k,C(s),C(o))}$ excluding the endpoint nodes from the path. These $\mathcal{A}_{(k,C(s),C(o))}$ without the endpoint nodes are referred to as path types for simplicity. PRA calculates the probability of arriving at the target node given that a random walk begins at the source node and follows a path type and vice versa. Formally, it is represented as $P(o_i|s_i, \omega_i)$, where ω represents the path types (also called features), which alternatively define the corresponding node pair (s_i, o_i) [44]. As shown in Figure 4a, the PRA algorithm creates a feature matrix where rows correspond to node pairs and columns correspond to the path types. It changes the cell values from representing the presence of an edge (same as in KGMiner) to representing the specificity of the connection that the path type makes between the node pair [44]. Since the feature space considered by PRA is very large (i.e., the set of all possible edge label sequences), the first step PRA performs is a feature selection, which is done by finding potential path types between node pairs. The second step of PRA is a feature computation. In this step, each cell in the feature matrix is computed based on constrained random walk probabilities that are associated with each path type and node pair [44]. Then, the feature matrix is used with a classifier (trained on positive examples and counterexamples) to validate the input assertion. The disadvantage of PRA is that it spends significant computational resources on the feature selection and computation steps. An advantage of PRA is that it offers some interpretability due to the features it learns.

Computing the random walk probabilities in PRA is an expensive operation. The subgraph feature extraction approach (SFE) [43] proposes an improvement of PRA by utilizing a new way of generating feature matrices over node pairs in a graph that aims to improve the efficiency and expressiveness of the model [43]. SFE uses features, which are extracted from subgraphs built from nodes in $\mathcal{G}$, to classify the given input assertion. In Figure 4b, blue rectangles depict the subgraphs. For each instance of $C(s)$ as s_i and $C(o)$ as o_i, a subgraph of depth m is generated using k random walks. Each random walk, that leaves the source node s_i or the target node o_i, follows some path type ω_i. After m steps, it ends either at an intermediate node n_i or at the endpoint node o_i or s_i. SFE keeps all of these (ω_i, n_i) pairs as characterization of subgraphs around the source and target nodes. These (ω_i, n_i) pairs are used to create 2 types of features: (1) two-sided random walk and (2) one-sided random walk. Two-sided random walk features are extracted by connecting the source node with the target node by the intersection of their respective subgraphs at a common node n_i. Hence, SFE updates the (ω_i, n_i) pair. The path type ω_i now becomes equivalent to PRA's path type and n_i becomes a target o_i or source s_i node (i.e., ω_i starts at s_i and ends at o_i or vice versa). To this end, these two-sided random walk features are dubbed as PRA-style features (PSF) [43]. The difference between feature matrices of PRA and SFE is that, instead of containing the associated probabilities, SFE's PSF contains only binary values (1 or 0), which denote the existence of a path (or paths) between the source and the target nodes. The reason behind this behavior is that it generates binary features, unlike the enumeration of KGMiner and random walk probabilities of PRA. However, because SFE does not have to compute random walk probabilities associated with each path type in the feature matrix, it can extract much more expressive features, including features that cannot be represented as paths in the graph (e.g., embedding-based features). In contrast to PRA, SFE does not limit itself to a single feature type. As shown in Figure 4b, SFE also extracts one-sided random-walk features (OSF) in the form of (ω_i, n_i) pairs, where n_i is not necessarily a target or source node. These one-sided features are intended to better model which sources and targets are good candidates for participation in a particular predicate. For example, in spite of the predicate's domain being all cities, not all cities participate in the predicate :cityCapitalOfCountry. A city with many government institutions is more likely to be a capital city. It would be easy to capture this kind of

information using these one-sided features. For obtaining feature matrices over node pairs in a graph, SFE is faster than PRA since it does not calculate random walk probabilities [43].

There are some other supervised meta-path-based approaches for fact checking over knowledge graphs that include: HeteSim [113] and forward stagewise path generation FSPG [85]. Both of these approaches require manually created meta paths, therefore we excluded them from Figure 4 and 5. However, we provide their comparison in Table 6 and 5. The HeteSim method uses path-constrained measures to analyze the semantic similarity between heterogeneous entities in $\mathcal{G}$. Path-constrained measures connect two nodes by following a strict sequence of node classes. For example, from Figure 1, the path $\texttt{:President} \xrightarrow{\texttt{:child}} \texttt{:Child} \xrightarrow{\texttt{:hasParent}} \texttt{:FirstLady} \xrightarrow{\texttt{:child}} \texttt{:Child} \xrightarrow{\texttt{:hasParent}} \texttt{:President}$ represents the :spouse relationship between :President and the :FirstLady, because both have the common relations :child and :hasParent connecting the same node :Child. HeteSim does not directly perform experiments for fact checking. However, this method can be used for fact-checking applications [56]. FSPG not only uses manually created meta paths but generates the meta paths under a given regression model as well. These meta paths are then used for the fact-checking task. In our survey, we have not included earlier approaches requiring manually annotated meta paths, such as PC [120], PCRW [68], and PathSim [121], since their publication dates do not fall within our selected time frame.

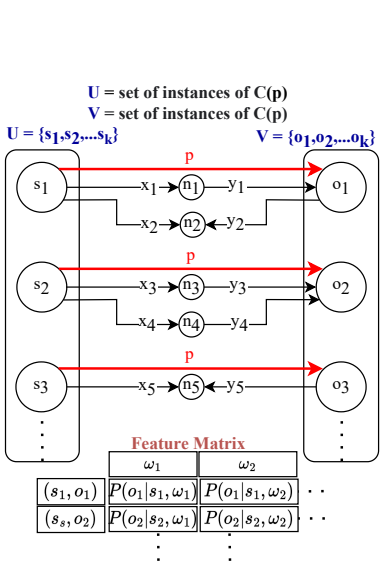
We also discuss 4 recent unsupervised path-based approaches that are part of this survey. In general, all these approaches are based on searching and scoring paths between the given subject and object.

Knowledge Linker (KL) [24] uses the shortest specific path to describe the truthfulness of the given claim. The shortest specific path is one that involves the fewest intermediate nodes with a minimum degree between the source and the target nodes. The specificity $\mathcal{S}$ of a path is defined in Equation 2.17. KL is based on a weighted adjacency matrix, with edge weights computed as the in-degree of each node in $\mathcal{G}$. These weights are then transformed into similarity scores. This method computes the veracity of the given claim by computing the proximity between subject and object. For proximity, KL introduces two distance closure methods: Metric and Ultra metric. Both methods consider the generality of a node, which is defined as its frequency in $\mathcal{G}$. In the first method, each path between a given subject and an object is mapped to a score generated based on the generality[24] of the intermediate nodes, i.e., a node with a higher degree conveys less information. In the second method, the score for each potential path is computed using only the node with the highest generality. The maximum score in both methods is equal to the shortest path between the subject and the object. In practice, the metric closure method works better as compared to the ultra-metric closure [56]. KL classifies the assertion in Figure 5a (i.e., B.Obama is related to Islam) as a false assertion, because the degrees of intermediate nodes, such as Canada or Columbia University, are very high. Hence, the path is very general and less informative.

For a given assertion, the Knowledge Stream (KS) [114] approach views knowledge as a certain amount of an abstract commodity that needs to be moved from s to o in $\mathcal{G}$. It considers that each edge is associated with two quantities: (1) a capacity of knowledge-flow related to $s \xrightarrow{p} o$ across its two endpoints, and (2) a cost of usage. Knowledge is viewed as a fluid, and $\mathcal{G}$ is viewed as a flow network. The width of an edge is roughly proportional to the flow of knowledge through it [114].[25] Let's assume an edge in a path is labelled with a predicate $p\prime$, which may differ from the target predicate p. Intuitively, the higher $p\prime$'s relevance to p is, the greater the edge's capacity. A data-driven approach is used to define the similarity between predicates using the structure of $\mathcal{G}$. To this end, the authors use the graph-theoretic concept of a line

[24] A node's generality is measured by its degree.

[25] There are three motivations for this idea: (1) the semantic context provided by multiple paths may be greater than that provided by a single path; (2) due to the non-disjoint nature of the paths, the method reuses edges participating in overlapping chains of relationships by sending additional flows; and (3) edges' limited capacities limit how many paths they can participate in, thus constricting the search path space [114].



a **Path ranking algorithm (PRA)**: Path-constrained random walk from s_i to o_i. Each path type is an alternate path from source to target node. Each cell value in the matrix is the probability of arriving at o_i starting from s_i, or vice versa

b **Subgraph Feature Extraction approach (SFE)**: Subgraph feature extraction approach (starting from s_i or o_i and find the intersection of subgraphs). Blue rectangles denote subgraphs and 0/1 denotes a binary value in the feature matrix; the abbreviations are:PSF/PRA-style features, OSF/One-sided random walk features, BFS/Breadth first search.

Fig. 4. Supervised path-based approaches for fact checking: $\{U\} \xrightarrow{x_i} \{n_i\} \xrightarrow{y_i} \{V\}$, $\{U\} \xleftarrow{x_i} \{n_i\} \xleftarrow{y_i} \{V\}$, $\{U\} \xleftarrow{x_i} \{n_i\} \xrightarrow{y_i} \{V\}$, and $\{U\} \xrightarrow{x_i} \{n_i\} \xleftarrow{y_i} \{V\}$ are the meta paths of $\{s\} \xrightarrow{p} \{o\}$, where x_i and y_i denote predicates and n_i are nodes on the path.

graph [114]. A path's bottleneck occurs at the edge at which the minimum similarity between p and $p\prime$ is found (i.e., the least relevant assertion along the path) [2]. The total knowledge that can flow through a set of paths between s and o is bound by the sum of their bottlenecks. The final veracity score of the input assertion is computed by identifying the set of paths responsible for the maximum flow of knowledge between s and o at the minimum cost, which is achieved by multiplying the specificity of paths (from KL approach) and the total knowledge that can flow through it. The KS approach is better when compared to the KL approach in terms of generating more meaningful explanations of its predictions. It does so by automatically discovering contextual assertions from the knowledge graph in the form of paths [114]. For example, Figure 5c depicts that the geography is less relevant than assertions about ancestry or family history to confirm the assertion `:B.Obama` $\xrightarrow{\texttt{:spouse}}$ `:M.Obama`. However, the same authors report that in terms of performance, KS lags behind KL [114]. This is possibly due to the fact that the extra signal provided by the additional paths found by KS may not always be beneficial.

Relational knowledge linker (KL−REL) is an extension of KL [24]. The KL approach fails to account for the semantics of the target predicate p [114]. In the KL−REL approach, the authors use the concept of semantic relevance from the KS approach. To achieve semantic relevance of the most specific paths, KL−REL adds bias in the search for specific paths

by favoring those edges semantically related to p. Ergo, KL−REL uses semantic relevance as the third criterion in addition to path specificity and path length used in the KL approach. The final veracity score is generated by taking the maximum of all these paths' specificity.

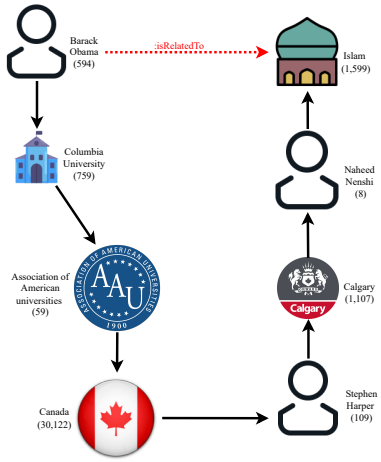
Using the domain, range, and RDFS class subsumption hierarchy of $\mathcal{G}$, COPAAL [124] filters meta paths π_k of length k that verify the predicate p and validate the given assertion $s \xrightarrow{p} o$. To this end, COPAAL excludes paths starting with a subject s outside the domain of the predicate p or ending with an object o outside the range of the predicate p (see Figure 5b). The resultant paths are dubbed as corroborative paths $\Omega^k(p)$ (see Definition 2.21). The corroborative paths $\Omega^k(p)$ and the given assertion's predicate p are then used to calculate the final veracity of the given assertion using the normalized pointwise mutual information (NPMI) [124]. In Figure 5b, $U\prime \in D(p)$ represents the set of subjects in the domain of the predicate p and $V\prime \in R(p)$ represents the set of objects in the range of the predicate p. COPAAL is the only path-based approach that make use of the combination of domain, range and subsumption hierarchy information expressed in the schema of most knowledge graphs. On the contrary, when this information is missing in the $\mathcal{G}$, then COPAAL fails to produce results [105].

5.3 Rule-mining-based approaches

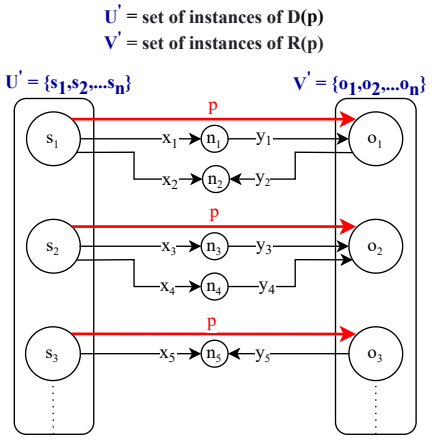
The identification and exploitation of constraints on databases have been widely discussed in the literature [47, 91, 106]. ILP-based systems (aka first generation rule mining systems), such as WARMeR [47] and Sherlock [106], are designed to work under CWA and require the definition of positive and negative error-free examples. It is difficult to apply these approaches directly to knowledge graphs because of their schema-less nature and OWA. Therefore, in order to mine rules from large knowledge graphs, several rule-mining-based approaches have been proposed. In order to perform the fact-checking or fact-prediction task, rule-mining-based approaches extract association rules from the knowledge graph [40, 41, 66]. In our survey, we compare the six most recent rule-mining-based approaches for knowledge graphs (see Table 7).

AMIE [41] is the first rule-mining-based approach explicitly designed for large-scale knowledge graphs. AMIE mines horn rules described in Definition 2.23 of the form $B_1 \wedge \ldots \wedge B_n \rightarrow H$, on large knowledge graphs. H is the head atom and $B_1 \wedge \ldots \wedge B_n$ are the body atoms. In contrast to the previous ILP-based methods, AMIE mines these rules from the knowledge graph without using explicit counterexamples. For implicit counterexample generation, AMIE uses PCWA(see Definition 2.14). The rule is instantiated by substituting constants for all its body atoms. Following the substitution of constants for variables in the body atoms, the head atom is predicted (see Definition 2.24). AMIE requires a knowledge graph $\mathcal{G}$, a threshold l for the maximal number of atoms per rule, *minHC* for the minimum head coverage (number of correct predictions), and *minC* for the minimum PCWA confidence as input [66]. AMIE uses breadth-first search. To this end, it initializes a queue with all possible rules of size 1, i.e., rules with an empty body [66]. The search strategy then dequeues a rule R at a time and adds it to the output list if certain conditions are met, specifically, (1) the rule is closed[26], (2) its PCWA confidence $\mathcal{C}_{\text{PCWA}}(R)$ is higher than *minC*, and (3) its PCWA confidence is greater than the confidence of all previously mined rules with the same head atom as R and a subset of its body atoms. AMIE refines R if it has fewer than l atoms and its confidence can still be enhanced. The refinement operator generates new rules from R by considering all possible atoms that can be added to the rule body. It constructs one new rule for each of them. AMIE iterates over all the non-duplicate refinements of rule R and adds those that have sufficient head coverage. When the rules queue is depleted, the routine ends [66]. One problem with AMIE is that the search space is exhaustive, i.e., every relation can potentially be

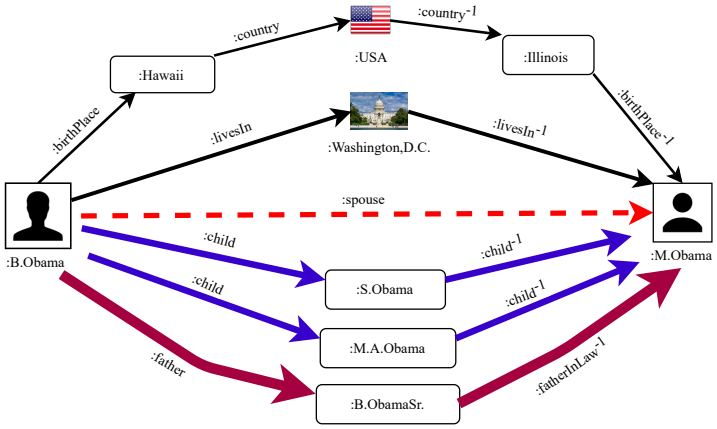
[26]A variable in a rule is closed if it appears at least twice in the rule. A rule is closed if all its variables are closed [41].



a **Knowledge Linkers (KL)**: Numbers represent the degree of nodes. High degree of nodes means general concept and low truth value (based on [24]).



b **COPAAL**: D(p) and R(p) represents domain and range of the predicate p, respectively.



c **Knowledge Stream (KS)**: The width of an edge is roughly proportional to the flow of knowledge through it. Paths representing the assertion $(\texttt{:B.Obama} \xrightarrow{\texttt{:spouse}} \texttt{:M.Obama})$. Arrows are in the direction of the stream, not in the direction of the predicates (based on [114]).

Fig. 5. Unsupervised path-based approaches for fact checking

combined with every other relation in a rule. Various optimization strategies have been proposed in the literature to deal with this drawback, such as approximate confidence calculation or sampling. The fewer samples the approach collects, the faster it becomes.

A similar optimization strategy is used by AMIE+ [40]—an extension of the AMIE project. AMIE+ provides pruning and approximation strategies to speed up the search by focusing on valid rules. Invalid rules are pruned beforehand. An example of an invalid rule is as follows:

$$\texttt{?x} \xrightarrow{\texttt{:livesIn}} \texttt{?z} \Rightarrow \texttt{?z} \xrightarrow{\texttt{:birthPlace}} \texttt{?y}. \tag{10}$$

As assertion ?x $\xrightarrow{\texttt{:livesIn}}$?z indicates that ?x should represent a person and ?z should represent a place. The right-hand side of the rule, i.e., ?z $\xrightarrow{\texttt{:birthPlace}}$?y, conflicts with the left-hand side, as ?z should represent a person according to this side of the rule. AMIE+ algorithm approximates confidence scores of such rules quickly. This is because it has pre-computed and stored the functionality score ζ of each relation and the overlaps between each relation pair's domains and ranges when $\mathcal{G}$ is loaded into the in-memory database. Consequently, the low-confidence rules are pruned at the beginning. AMIE+ provides further improvement by iteratively extending rules using a set of mining operators (e.g., the refinement operator of AMIE). However, this can only be done if the rules can be closed before the length constraint is exceeded. In addition, AMIE+ tunes the following parameters to reduce the runtime: rule length, minimum rule confidence, and head coverage. However, this parameter tuning comes with a trade-off between the number of rules discovered and the runtime. For example, if we reduce the rule length, the number of rules discovered will be reduce as well while the algorithm will terminate faster. If the rule length is increased, more rules can be discovered but the algorithm will be slower. However, since most users are typically interested in many rules with low scores, rules with low head coverage, low confidence, and lengths greater than 3 are discarded by AMIE+ by default.

AMIE-3.0 [66] is the successor to AMIE+. Approximation and sampling make rule-mining faster but less accurate. On the contrary, AMIE-3.0 focuses on accuracy and completeness and employs further pruning strategies, parallelization, and lazy confidence scores computation to speed up the rule-mining process. It does not calculate the support and confidence of "bad" rules, because those rules are pruned anyway. Its optimization is based on the principle, "if you know something is bad, don't spend time figuring out how bad it is" [66]. AMIE-3.0 also uses in-memory databases to store the entire compressed knowledge graph in-memory for faster computations. All strings are compressed into byteStrings where each character is only 8 bits. Equivalent byteStrings point to the same physical object (i.e., each string exists only once), which saves a considerable amount of memory in large knowledge graphs.

Ontological Path-finding (OP) [22, 23] is a rule-mining-based approach that focuses on large-scale knowledge graphs and provides parallelization-based optimizations. The algorithm applies inference rules in batches, parallelizing the join queries in the rule-mining process, partitioning the rule-mining process into sub-tasks, and pruning the invalid rules before applying them. Like AMIE-3.0 they also use in-memory joins and execute them in parallel by dividing the large knowledge graphs into smaller groups in order to perform rule-mining. To maximize resource utilization, they formalize the rule-mining algorithm using a state-of-the-art cluster computing framework, such as SPARK.

The aforementioned rule-based discover only positive rules. However, positive rules alone may not be able to eliminate all inconsistencies [95]. Positive rules identify relationships between entities, e.g., "if two persons have the same parent, they are siblings"; negative rules identify data contradictions, e.g., "if two persons are married, one cannot be the child of the other". In the following, we will describe approaches that employ both positive and negative rules.

RuDiK [95] discovers positive and negative rules mined from noisy and incomplete large-scale knowledge graphs. In this approach, both positive and negative examples for all the predicates are provided as input to their rule-discovery module.[27]. To obtain all rules for a $\mathcal{G}$, RuDiK computes rules for every predicate in it. It starts with a small set of approximate horn rules, i.e., rules that at least cover one positive example pair. In order to solve the mining problem exactly, all pairs in the positive example set must be covered and none in the negative example set. RuDiK minimizes the number of rules in the output to avoid overfitting rules covering only one pair, as such rules have no impact when applied

[27]These examples can be hand-crafted by domain experts or automatically generated by their proposed negative sampling method called extended partial closed world assumption (E-PCWA). E-PCWA is described in detail in Section6.3

to the $\mathcal{G}$. As a result, only promising rules[28] are generated. The negative rules are generated by the same method as the positive rules. The only difference is that it replaces the positive examples with the negative ones. The RuDiK algorithm assigns weights to both negative and positive rules. They use positive rules to predict new assertions and negative rules to check for inconsistencies in the knowledge graph. For example, a positive rule could state that if a person X lives in Y and Z also lives in Y, they are a couple (i.e., :spouse). By stating that the children live with their parents in the same house, the negative rule contradicts the positive rule. Using disk-based algorithms, RuDiK increases its search space and discovers rules with richer language and greater expressive power compared to the aforementioned in-memory database-based algorithms. Furthermore, it does not load the entire graph into memory like AMIE-3.0 does, which makes it a scalable approach. On the contrary, disk-based algorithms make RuDiK slower in terms of speed compared to in-memory database-based algorithms.

KV-Rule [58] is a rule-mining-based approach that mines positive and negative rules in $\mathcal{G}$ for a given assertion. The KV-Rule workflow consists of 3 steps. It (1) generates positive and negative examples, (2) generates positive and negative rules and weights them according to their logical validity, and (3) scores the given assertion by an ensemble of the most valid positive and negative rules that cover the given assertion. In the first step, positive examples are sampled from the $\mathcal{G}$ itself. For the generation of negative examples, KV-Rule's authors propose a distant partial closed world assumption (D-PCWA). The D-PCWA approach does not replace the entities (s or o) with neighboring entities in $\mathcal{G}$, but rather with distant entities.[29] In the second step, positive and negative rules are generated and weighed using the generated examples. KV-Rule generates positive rules using positive assertions. In contrast, negative rules are generated using negative assertions. In order to weight a positive rule according to its logical validity, positive assertions are used as correct examples and negative assertions as counterexamples. In contrast, to weight a negative rule according to its logical validity, negative assertions are used as correct examples, and positive assertions are used as counterexamples. In the last step, for calculating a truth score for a given assertion, KV-Rule finds the most valid positive and negative rules that cover the given assertion and calculates a final truth score [58].

5.4 Graph pattern-mining-based approaches

Graph pattern-mining-based approaches rely on an approximate pattern-matching-based approach (see Definition 2.30) instead of subgraph isomorphism. These approaches are different from rule-mining-based and path-based approaches, because rule-mining-based approaches usually cover only a subset of useful patterns and are easy to interpret, and path-based approaches provide explainable models but lack the expressiveness to explicitly describe subgraphs as discriminant features [76]. The focus of these approaches is on identifying "soft rules" to infer new facts for the knowledge graph completion task, instead of detecting errors using hard constraints. GFC-based approaches overcome these limitations. There are two GFC-based approaches proposed in the literature: GFC [76] and OGFC [77].

The GFC approach [76] is an approximate pattern mining-based approach. It adopts approximate pattern mining instead of subgraph isomorphism. It discovers a set of top-f GFCs that distinguish true and false assertions best using quality measures, which are used for characterization. These patterns are then used to construct classifiers based on either rules or latent features. To this end, they propose two methods: $GFact_R$ and $GFact$. $GFact_R$ is a rule-mining-based model, which uses the top-f GFCs as fact-checking rules. It follows the "hit-and-miss" convention and checks whether there exists a GFC φ within the top-f GFCs that covers the given assertion t. If so, it accepts the given assertion, otherwise, it rejects it. On the other hand, $GFact$ extracts instance-level features from GFC patterns to train the classifier. It constructs a feature

[28] Rules which cover maximum positive example pairs.
[29] A comprehensive comparison of PCWA, E-PCWA, and D-PCWA is provided in Section 6.3.

vector of size k, where each entry encodes the presence of an i-th GFC_i in the set of top-f GFCs. The class label is true (resp. false) if the assertion belongs to a positive (resp. negative) example assertion set. In the training data, the positive example assertion set is sampled from the $\mathcal{G}$, while the negative assertion set is generated using `WPCA` (see Definition 2.31). *GFact* adopts logistic regression, and it achieves better performance than $GFact_R$ over real-world knowledge graphs [76]. For example, from our excerpt of the knowledge graph in Figure 1, it can be observed that a pattern between `:B.Obama` and his wife `:M.Obama` exists. Both have common children and both are related to the `:USA` as `:president` and `:firstLady` relationship. So a graph pattern could be: `:President` associated with the `:Child` using `:child` relationship. `:FirstLady` associated with the same `:Child` with `:child` relationship. Both are associated with the `:Country` entity with `:president` and `:firstLady` relationship. Ergo, we could assume that this is one of the possible top-f graph-patterns which corroborates the veracity of the given assertion `:B.Obama` $\xrightarrow{\texttt{:spouse}}$ `:M.Obama`.

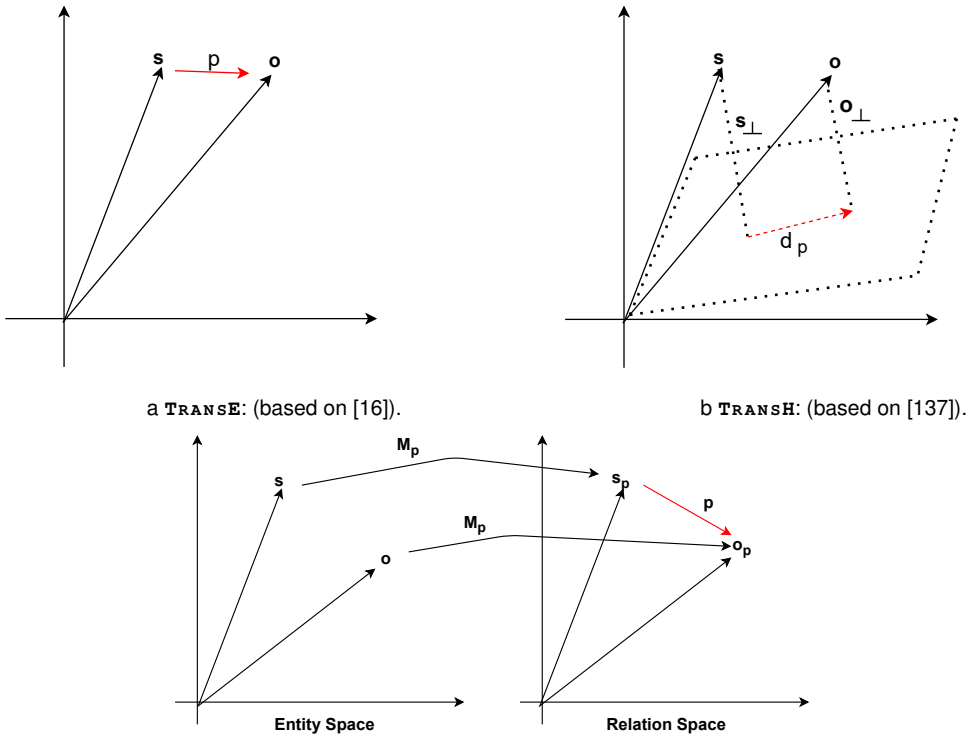
`OGFC` [78] uses ontological features to aid pattern-mining in the `GFC` approach. It extracts more expressive features from knowledge graphs due to the usage of ontological information. To this end, `OGFC` replaces the semantic similarity defined in Definition 2.29 with the ontological closeness of concept nodes. The ontological closeness uses three features of the ontology for the given description of an edge (v_x, v_y) as $(L(v_x), L(v_y))$ (from Definition 2.27). These three features are: (1) *equivalence* states that $L(v_x)$ and $L(v_y)$ are semantically equivalent, thereby representing "refersTo" or "knownAs"; (2) *hyponymy* states that $L(v_x)$ is a kind of $L(v_y)$, such as "isA" or "subclassOf" that enforces a preorder over all ontology concepts; and (3) *descriptiveness* states that $L(v_x)$ is described by another $L(v_y)$ in terms of, for example, "association", "partOf" or "similarTo" [77]. These features help in finding more expressive graph patterns φ. This approach discovers discriminant top-f graph patterns using ontologies that best distinguish between true and false assertions in knowledge graphs. In the `OGFC` approach, semantically related nodes having ontologically similar descriptions are considered the same objects. For example, a "president" and a "head of the state" could be different nodes in $\mathcal{G}$ but are considered the same when discovering a graph pattern φ, due to their ontological closeness. For the fact-checking task, the rest of the approach is similar to the `GFC` approach, except they use the names $OFact_R$ and *OFact* for the two different methods instead of $GFact_R$ and *GFact* as in the `GFC` approach.

5.5 Embedding-based approaches

Existing methods for reasoning over knowledge graphs are neither tractable nor practical when dealing with long-range reasoning over large-scale knowledge graphs [137]. Generally, a knowledge graph embedding model represents an entity as a k-dimensional vector ($\mathbf{s}$ or $\mathbf{o}$). [30] In addition, an embedding model has a scoring function $f(\mathbf{s}, \mathbf{o})$ that calculates the validity of an assertion $s \xrightarrow{p} o$ in the model's embedding space. For example, in `TransE` the scoring function $f_p(\mathbf{s}, \mathbf{o}) \triangleq \|\mathbf{s} + \mathbf{p} - \mathbf{o}\|_{\ell_{1/2}}$ implies a transformation vector $\mathbf{p}$ on the subject entity (i.e., s). This implies that $\mathbf{o}$ is the close neighbor of $\mathbf{s} + \mathbf{p}$ (see Figure 6a). Hence, if the assertion is true, the scoring function produces a low value. Otherwise the value should be high. Different methods, such as `TransE` [16], `TransH` [137], and `TransR` [79], propose different scoring functions. All these methods are supervised as they need training data to create the embedding vectors. It is important to note that most of these methods aim to achieve a different but related goal, i.e., link prediction. However, the authors of `TransH` and `TransR` also performed experiments for fact checking over knowledge graphs. Therefore, we include them in our survey.

As discussed earlier, `TransE` is the embedding-based method proposed in the literature, which is efficient while achieving state-of-the-art predictive performance [137]. However, it lacks some important properties ,such as reflexive,

[30] bold face characters represent the embedding vector representations of entities and relations



a TransE: (based on [16]). b TransH: (based on [137]).

c TransR: (based on [79]).

Fig. 6. Embedding-based approaches for fact checking

one-to-many, many-to-one, and many-to-many. TransH fills this gap by modeling a relation as a hyperplane ($\mathbf{w}_p$ as a norm vector) together with the translation operation $(\mathbf{d}_p)$ on it. TransE claim that by using this method they have preserved the above mapping properties without increasing complexity. In TransH, each relation is characterized by two vectors, the norm vector $(\mathbf{w}_p)$ of the hyperplane, and the translation vector $(\mathbf{d}_p)$ on the hyperplane. In order for a given assertion $s \xrightarrow{p} o$ to be correct , the projections of $\mathbf{s}$ and $\mathbf{o}$ on the hyperplane ($\mathbf{s}_\perp$ and $\mathbf{o}_\perp$) are expected to be connected by the translation vector $\mathbf{d}_p$ with low error (as shown in Figure 6b). The scoring function is defined as $f_p(\mathbf{s}, \mathbf{o}) = \|\mathbf{s}_\perp + \mathbf{p} - \mathbf{o}_\perp\|_2^2$. This method overcomes TransE's bottleneck in dealing with reflexive, one-to-many, many-to-one, and many-to-many relations while keeping the model complexity almost the same [137].

TransR, on the other hand, deals with this problem by modeling entities and relations in different semantic (vector) space bridged by relation-specific matrices [79]. The reason for this behavior is that entities can have different aspects, and different relations may focus on different aspects of entities. Afterward, the algorithm learns the embeddings by projecting entities from the entity space to the relation space and then building translations between projected entities. In TransR, for each assertion $s \xrightarrow{p} o$, entity embeddings are set as $\mathbf{s}, \mathbf{o} \in \mathbb{R}^k$ and relation embedding is set as $\mathbf{p} \in \mathbb{R}^d$. The number of dimensions of entity embeddings k and relation embeddings d are not necessarily the same. For each relation p, TransR sets a projection matrix $\mathbf{M}_p \in \mathbb{R}^{k \times d}$, which may project entities from the entity space to the relation space [79] (as shown

in Figure 6c). With the mapping matrix, TRANSR defines the projected vectors of entities as $\mathbf{s}_p = \mathbf{s}\mathbf{M}_p$ and $\mathbf{o}_p = \mathbf{o}\mathbf{M}_p$. The scoring function is the same as TRANSE, but the entities are $\mathbf{s}_p$ and $\mathbf{o}_p$.

In contrast to existing meta-path-based approaches, which use discrete representations of $\mathcal{G}$, ESTHER employs compositional embedding spaces of the input knowledge graph for the meta-path search [27]. The idea behind exploiting a compositional embedding space is to minimize the loss function between continuous representations of the given predicate embedding $\mathbf{p}$ and the embedding of the corroborative path $\Omega^k(p)$. Corroborative path embeddings are computed by concatenating the embeddings of all predicates along the path. ESTHER improves the exploration of longer meta paths, which helps in detecting supplementary evidence for validating the given assertion.

5.6 Hybrid fact-checking approaches

Each category of approaches described above comes with certain limitations. For example, a knowledge graph may contain conflicting rules or paths that support and negate the given assertion. In addition, it can also lack evidence supporting the given assertion. Moreover, text-based approaches cannot provide semantic explanations of answers. There are also some cases when individual categories of fact-checking approaches can't even obtain an answer [1, 105]. Consequently, hybrid fact-checking approaches, such as FACTY [71], MAP [1], ExFaKT [38], Tracy [39], and HybridFC [105], that exploits the diversity of existing categories of fact-checking approaches to find the veracity of a given assertion, have been proposed.

FACTY [71] is an approach that deals with only tail verticals, such as less popular domains, in a knowledge graph. A vertical is a tail vertical if its subject entities are not globally popular, i.e., in terms of search query volume. The number of assertions in a tail vertical is usually below a million in a large $\mathcal{G}$. Examples of tail verticals are pizza varieties, yoga poses, and car models. Knowledge about tail verticals is not widely known on the web, and search results can be very noisy. Hence, it is difficult to find evidence on the web or from knowledge graphs for tail verticals. According to the survey stated in the FACTY paper, only 19% of tail verticals could be verified with 22% accuracy on the web [71]. The FACTY algorithm not only leverages textual reference knowledge for evidence gathering but structural reference knowledge, such as a knowledge graph, as well. As a first step in finding evidence, FACTY uses structured reference knowledge because it is a reliable source of information. However, its coverage is limited. FACTY does so by searching for the subject s and object o of the given assertion in the knowledge graph. Evidence containing a particular predicate p is ignored because it leads to low recall. In the second step, it searches for evidence on the web, which provides wider coverage. However, the web is noisy. The web contains rich information in various structured and unstructured formats, such as web tables, DOM trees, and text. FACTY's authors apply various techniques to extract evidence from the web, ranging from co-occurrences of entity names to sophisticated supervised learning like in [45]. Finally, FACTY also enriches the evidence by analyzing search query logs that reflect users' intent. Afterward, it calculates a final veracity score for the given assertion based on collected evidence.

MAP [1] is an approach that uses both rule-mining-based and text-based methods to find evidence and validate the given assertion. Discovering rules and mining assertions from the web enable a fully automated system that covers single approaches' limitations. These approaches, however, present a substantial challenge, since rules from the knowledge graph and assertions from the web are uncertain and not infallible. To address this challenge, MAP uses probabilistic answer set programming [90]. MAP's algorithm contains a reasoner that combines all evidence from both approaches and makes a fact-checking decision with explanations. The rules from the rule-mining approach and evidence from text-based approaches are fed to the reasoner module, where different modes of computation are used to infer positive or negative evidence from the stable model (aka. answer set). The reasoner output includes a human-interpretable explanation for the

decision. For example, consider the following positive rule from our example in Figure 1:

$$?x \xrightarrow{:spouse} ?z \Leftarrow ?x \xrightarrow{:child} ?y \xrightarrow{:parent} ?z. \tag{11}$$

The claim that we want to check is: $:B.Obama \xrightarrow{:spouse} :M.Obama$. $\mathcal{G}$ contains the assertion $:B.Obama \xrightarrow{:child} :M.A.Obama$. Considering the assertion $:M.A.Obama \xrightarrow{:parent} :M.Obama$ missing from $\mathcal{G}$, MAP will get evidence for the first assertion $:B.Obama \xrightarrow{:child} :M.A.Obama$ from $\mathcal{G}$, and then it will query the web module with the second assertion $:M.A.Obama \xrightarrow{:parent} :M.Obama$ to obtain evidence. Finally, these pieces of evidence are fed to the reasoner module to compute the final output.

ExFaKT is a framework for computing pieces of evidence for the given assertion in $\mathcal{G}$ and the web [38]. ExFaKT utilizes horn rules to formulate sub-queries and explanations. In addition to automatic rules generated from existing approaches, such as AMIE [40], the system also uses a hand-annotated rule-set to improve sub-queries. Rules from the rule-set with the same head predicate as the input query are considered first. The next step is to discover the body atoms of those rules and write the sub-queries iteratively. Recursively, these sub-queries might also trigger new rules (whose head matches the body atom). These sub-queries first try to get the answer from $\mathcal{G}$. If $\mathcal{G}$does not return an answer, it searches for evidence on the web using the fact-spotting FS module. The process continues recursively until all rule atoms (sub-queries) are triggered and explanations are generated. For example, a user searches for the following query on ExFaKT: "Barack Obama is a citizen of USA". The rule-mining approach may generate the following rule against this query:

$$?x \xrightarrow{:citizenOf} ?z \Leftarrow ?x \xrightarrow{:birthPlace} ?y \xrightarrow{:country} ?z. \tag{12}$$

ExFaKT first tries to find the assertion $:B.Obama \xrightarrow{:birthPlace} :Hawaii$ in $\mathcal{G}$. Suppose the assertion $:Hawaii \xrightarrow{:country} :USA$ is missing from $\mathcal{G}$. Then the FS module will be triggered. In FS, the algorithm searches for evidence of this assertion on the web. At the end of the process, all explanations in the form of evidence will be presented to the user along with confidence scores. The purpose of ExFaKT is not to calculate the overall veracity score of an assertion, but rather to provide users with better explanations. Nevertheless, the confidence scores generated for every explanation against the given assertion are used to calculate the final veracity score. Tracing Facts over Knowledge Graphs and Text (Tracy), which is the demo extension of ExFaKT works similarly with some minor optimizations in the FSmodule [39].

In an ensemble learning setting, HybridFC [105] combines three categories of approaches, namely text-based, embedding-based, and path-based. The input to HybridFC is an assertion, which is then forwarded to individual categories of approaches. It utilizes the existing categories of approaches that generate the following outputs: text snippets, i.e., the output of evidence search on the web, pre-trained knowledge graph embeddings, and the veracity score output of the path-based approach, i.e., COPAAL. All these outputs are encoded into vector representations and fed as input to the multi-layer perceptron, which produces the final veracity score [105]. To improve prediction performance, we conclude that HybridFC utilizes a variety of existing methods within an ensemble learning framework.

6 BENCHMARK DATASETS

Fact checking approaches like the ones presented in the previous section are typically evaluated and compared based on benchmark datasets. Hence, we include a comparison of these datasets in our survey. In this section, we first describe the requirements of an ideal fact-checking dataset. Afterward, we list existing fact-checking benchmark datasets before we present strategies to generate fact-checking datasets—training examples in particular—from knowledge graphs.

6.1 Fact-checking dataset requirements

According to our analysis, a good fact checking benchmark dataset should fulfill the following requirements [4, 45, 123, 129]: it should comprise (1) assertions that should be validated by the fact checking approach, (2) the ground truth for the aforementioned assertions, (3) with an established labeling scheme, (4) with a clear definition of labels (e.g. `True` and `False`), (5) an equal number of assertions per label, and (6) enough assertions to train complex models. However, a dataset may have additional features, e.g., a temporal aspect for those assertions that are dependent on time (e.g., players often change their teams) or provenance for the ground truth.

6.2 Existing fact-checking datasets

In Table 8, we provide a comparison of different datasets proposed in the literature. There are two classes of datasets: textual datasets are those datasets that are manually or automatically created from textual sources in the form of textual claims, and knowledge graph-based datasets are those datasets that are created from large knowledge graphs or textual sources in the form of assertions. These datasets are created for evaluation purposes, specifically for the fact-checking task.

A small corpus of 106 claims from Politifact and Channel4 is one of the first fact-checking datasets [131]. A further dataset (dubbed EMERGENT) has 300 claims obtained from Twitter and Snopes [37]. These preliminary fact-checking corpora offer indicative positive progress toward a rigorous formulation of the fact-checking problem. However, a major bottleneck associated with automated fact checking is the shortage of available training data. Consequently, due to their small size, it becomes difficult to train machine learning models. The volume of data available has recently increased due to fact-checking websites, generative language models, and corpus annotation platforms [55, 64]. Since then, numerous corpora have been constructed with a number of claims that are orders of magnitude larger, e.g., in thousands or tens of thousands. LIAR consists of 12.8K claims retrieved via the Politifact API [63]. It also contains metadata for these claims, i.e., the speaker of the claim, political affiliations of the speaker, and the medium through which the claim was first published (e.g., tweet, speech, op-ed piece). Due to its large size, the dataset is better for evaluating machine learning models in fact-checking systems. However, the evidence used by fact-checkers to verify the claims is still missing from the dataset [64]. Since the publication of the LIAR dataset, many other substantially larger datasets have emerged. These datasets also include evidence and metadata to contextualize the claim. For example, the FEVER dataset consists of 185k claims from Wikipedia [129]. FEVEROUS [4] on the other hand contains 87k claims, but the average length of the claims has increased significantly. To the best of our knowledge, NELA-GT-2020 [48] and CREDBANK [88] are the largest known textual sources-based fact-checking datasets containing 17 Million and 60 Million claims, respectively.

Various labeling schemes are adopted for the construction of different datasets, e.g., binary labels [102] and graded classification [63, 131] inform the definition of veracity. These veracity labels vary considerably across datasets (see Table 9). For example, KL-dataset [24], FactBench [45], KGMiner-dataset [112], KS-dataset [114], BPDP [123], and KV-eval [58] use binary labels. FEVER [129] uses three-way classification, with the label `NOT-ENOUGH-INFO` to account for claims for which there is insufficient data to classify them as true or false. On the contrary, LIAR uses a six-point veracity system (`pants-fire`, `false`, `barely-true`, `half-true`, `mostly-true`, `true`) borrowed from Politifact [63]. Some fact checkers published the dataset with news articles labeled as `hoax`, `satire`, `propaganda`, and `trusted news` [64, 107]. They identify the veracity of an article by investigating its sources and checking whether

Table 8. Statistics of fact-checking datasets. Datasets are separated based on the property of using knowledge graphs or textual sources. "GREC" denotes Google Relation Extraction Corpora, and "WSDM" means the WSDM Cup 2017 Triple Scoring challenge. "Wiki" denotes Wikipedia info boxes. "NAET" denotes News Articles with Embedded Tweets. "en" denotes English, "de" denotes German, and "ml" denotes multilingual. IFCN denotes International Fact-Checking Network. C. denotes Celebrity. DA denotes Data Availability, C/A denotes Claims/Assertions, # C/L denotes # Classes/Labels, Ev. denotes Evidence, Ex. denotes Explanation, TA denotes Temporal aspects, and Lang. denotes Language.

	Dataset	Year	DA	C/A	Sources	# C/L	Ev.	Ex.	TA	Lang.
Textual source-based	Politifact [131]	2014	✗	106	Fact-checking websites	5	✓	✗	✗	en
	EMERGENT [37]	2016	✓	300	Twitter, Snopes	3	✓	✗	✗	en
	Some Like it Hoax [126]	2017	✓	15,500	Facebook	2	✗	✗	✗	en
	Snopes [103]	2017	✓	4,956	Snopes[31]	3	✓	✗	✗	en
	LIAR [63]	2017	✓	12,836	PolitiFact	6	✗	✗	✗	en
	FakeNewsAMT& C. [102]	2018	✓	980	News, GossipCop	2	✗	✗	✗	en
	FakeNewsNet [115]	2018	✓	23,921	PolitiFact, GossipCop	2	✓	✗	✗	en
	LIAR-PLUS [3]	2018	✓	12,836	PolitiFact	6	✓	✗	✗	en
	FEVER [129]	2018	✓	185,445	Wiki	3	✓	✗	✗	en
	BuzzFace [109]	2018	✓	2,263	FB comments and reactions	4	✗	✗	✗	en
	PERSPECTRUM [21]	2019	✓	907	debating websites	5	✓	✗	✗	en
	PTC [26]	2019	✓	451	News articles	18	✓	✗	✗	en
	UKP Snopes [51]	2019	✓	6,422	Snopes[32]	3	✓	✗	✗	en
	MultIFC [9]	2019	✓	36,534	Fact checking websites	2-40	✓	✗	✗	en
	GermanFakeNC [132]	2019	✓	490	News articles	2	✓	✗	✗	de
	NELA-GT-2020 [48]	2020	✓	1,779,127	NAET	3	✓	✓	✗	en
	WikiFactCheck [110]	2020	✓	124,821	Wiki	2	✓	✗	✗	en
	PolitiHop [96]	2020	✓	500	Politifact	3	✓	✓	✗	en
	Climate-FEVER [31]	2020	✓	1,535	Climate-change Wiki	4	✓	✗	✗	en
	PUBHEALTH [65]	2020	✓	11,832	Various	4	✓	✓	✗	en
	SCI-FACT [133]	2020	✓	1,409	S2ORC[33]	3	✓	✓	✗	en
	FEVEROUS [4]	2021	✓	87,026	Wiki	3	✓	✓	✗	en
	COVID-Fact [108]	2021	✓	4,086	Forum	2	✓	✓	✗	en
	X-Fact [50]	2021	✓	31,189	IFCN	7	✓	✓	✗	ml
	CREDBANK [108]	2021	✓	60 Million	Twitter	5	✓	✗	✗	en
	Vitamin-C [111]	2021	✓	488,904	Wiki	3	✓	✗	✗	en
	MuMiN [94]	2022	✓	12,914	Twitter	3	✓	✗	✗	ml
$\mathcal{G}$-based	KL-dataset [24]	2015	✗	11,639	GREC/Wiki	2	✓	✗	✗	en
	FactBench [45]	2015	✓	10,500	DBpedia and Freebase	2	✗	✗	✓	ml
	KGMiner-dataset [112]	2016	✓	27,313,477	DBpedia/SemMedDB	2	✗	✗	✗	en
	KS-dataset [114]	2017	✓	18,431	GREC/Wiki/WSDM	2	✓	✓	✗	en
	BPDP [123]	2018	✓	1,596	DBpedia	2	✗	✗	✓	ml
	ClaimKG 2019 [127]	2019	✓	33,261	Fact-checking websites	4	✓	✓	✗	en
	KV-eval [58]	2020	✓	1,759	Wiki	2	✓	✓	✗	ml
	ClaimKG 2022 [42]	2022	✓	59,580	Fact-checking websites	4	✓	✓	✗	en

their websites are deemed to be "fake news" or not[37] [107]. In addition, some fact checkers provide explanations that justify the labels assigned to the single claims.

Datasets like MultiFC [9], FakeNewsNet [115], and UKP Snopes [51] contain real-world claims from multiple domains. The US presidential election of 2016 has spurred the development of a growing number of datasets in the political domain (e.g., Snopes, FakeNewsAMT & Celebrity, and FakeNewsNet). However, more datasets outside the political domain, i.e.,

[37] According to a US News and world report https://www.usnews.com/.

Table 9. Statistics of fact-checking datasets. Datasets are separated based on the property of using knowledge graphs or textual sources. * indicates "all labels except pants-fire, ** indicates only the test split, not the training one, *** indicates slight imbalanced data. C. denotes Celebrity.

	Dataset	Balanced	Classes/Labels	Link to data
Textual source-based	Politifact [131]	-	True, MostlyTrue, HalfTrue, MostlyFalse, False	-
	EMERGENT [37]	no	For, Against, Observing	https://github.com/willferreira/mscproject
	Some Like it Hoax [126]	no	hoax, non-hoax	https://github.com/gabll/some-like-it-hoax
	Snopes [103]	no	true, false	link[34]
	LIAR [63]	yes*	pants-fire, false, half-true, barely true, mostly-true, true	https://www.cs.ucsb.edu/~william/data/liar_dataset.zip
	FakeNewsAMT& C. [102]	yes	legitimate, fake	https://lit.eecs.umich.edu/downloads.html#Fake%20News
	FakeNewsNet [115]	-	fake, real	https://github.com/KaiDMML/FakeNewsNet
	LIAR-PLUS [3]	yes	pants-fire, false, half-true, barely true, mostly-true, true	https://github.com/Tariq60/LIAR-PLUS
	FEVER [129]	yes**	SUPPORTED, REFUTED, NOTENOUGHINFO	https://fever.ai/
	BuzzFace [109]	no	mostly true, mixture of true and false, mostly false, no factual content	https://github.com/gsantia/BuzzFace
	PERSPECTRUM [21]	-	support, oppose, mildly-support, mildly-oppose, valid	https://github.com/CogComp/perspectrum
	PTC [26]	no	18 labels[35]	https://propaganda.qcri.org/
	UKP Snopes [51]	no	agree, refute, no stance	https://tudatalib.ulb.tu-darmstadt.de/handle/tudatalib/2081
	MultIFC [9]	-	40 labels	http://www.copenlu.com/publication/2019_emnlp_augenstein/
	GermanFackeNC [132]	no	disinformation in text: range [0.1:1.0]	https://zenodo.org/record/3375714#.YwZEnXZBxmM
	NELA-GT-2020 [48]	yes	reliable, unreliable, mixed	doi:10.7910/DVN/CHMUYZ
	WikiFactCheck [110]	no	supports, refutes	http://github.com/WikiFactCheck-English
	PolitiHop [96]	yes***	True, HalfTrue, False	https://github.com/copenlu/politihop
	Climate-FEVER [31]	yes	SUPPORTS, REFUTES, DISPUTED, NOTENOUGHINFO	http://climatefever.ai/
	PUBHEALTH [65]	no	true, false, mixture, unproven	https://github.com/neemakot/Health-Fact-Checking
	SCI-FACT [133]	yes***	supports, refutes, noinfo	https://github.com/allenai/scifact
	FEVEROUS [4]	no	SUPPORTED, REFUTED, NOTENOUGHINFO	https://fever.ai/dataset/feverous.html
	COVID-Fact [108]	no	supported, refuted	https://github.com/asaakyan/covidfact
	X-Fact [50]	yes	7 labels[36]	https://github.com/utahnlp/x-fact
	CREDBANK [108]	no	5-point Likert scale ranging from [-2 to +2]	https://compsocial.github.io/CREDBANK-data/
	Vitamin-C [111]	yes***	SUPPORTED, REFUTED, NOTENOUGHINFO	https://github.com/TalSchuster/VitaminC
	MuMiN [94]	no	misinformation, factual and other	https://github.com/MuMiN-dataset/mumin-build
$\mathcal{G}$-based	KL-dataset [24]	yes	true, false	https://github.com/shiralkarprashant/knowledgestream/
	FactBench [45]	yes	true, false	https://github.com/DeFacto/FactBench
	KGMiner-dataset [112]	yes	true, false	https://github.com/nddsg/KGMiner/
	KS-dataset [114]	yes	true, false	https://github.com/shiralkarprashant/knowledgestream
	BPDP [123]	yes	true, false	-
	ClaimKG 2019 [127]	no	FALSE, TRUE, MIXED, OTHER	https://data.gesis.org/claimskg/
	KV-eval [58]	no	true, false	https://github.com/machinereading/KV-eval-dataset
	ClaimKG 2022 [42]	no	FALSE, TRUE, MIXED, OTHER	https://doi.org/10.7802/2469

in the scientific [133] and forensic [65] domains, have emerged, recently. These datasets also include natural language explanation of the labels (e.g., a paragraph from a news article explaining the assigned label).

In addition to the above datasets, there are a few knowledge-graph-based datasets proposed in the literature. For example, FactBench, BPDP, and KV-evalare multilingual benchmark datasets for the evaluation of fact validation algorithms. FactBench is based on the DBpedia and Freebase knowledge graphs [45]. It is divided into train and test sets, both containing true and false assertions. The true assertions were generated by issuing SPARQL and MQL queries and selecting the top 1,500 results for each relation. The false assertions were derived by modifying the true assertions, and they comprises 6 categories (i.e., domain, domain-range, range, date, property, and mix). For each of the six categories of false assertions, FactBench contains 1,500 assertions that were also separated into two halves for training and testing. Furthermore, the Birth Place/Death Place (BPDP) dataset [123], was created based on the observation that some fact-checking approaches only check if the subject and object have a relation to each other. However, the type of the relation, i.e., if it matches the property of the given assertion, is not always taken into account [105]. Later, ClaimKG [127], KGMiner-dataset [112], KS-dataset [114], and KL-dataset [24] were also published to evaluate the fact-checking task over knowledge graphs.

6.3 Training examples generation

There are three methods described in the literature for producing training data [101]: (1) partial gold standard, (2) knowledge graph as a silver standard, and (3) retrospective evaluation. In the partial gold standard method, manually annotated assertions of high quality are created. In the knowledge graph as a silver standard method, true and false assertions are generated from the knowledge graph by considering OWA and other techniques, which we will discuss in the following paragraphs. In the retrospective evaluation method, assertions are generated using any automatic approach and then handed to experts, so they can annotate them. Due to manual annotation schemes in the first and third categories, most of the approaches discussed in this survey generate training data using the second category (i.e., silver standard).

Fact-checking approaches need true (Γ^+) and false (Γ^-) assertions for evaluation and training. Claims in Γ^- are also dubbed counterexamples. Γ^+ are sampled from the knowledge graph itself. However, the generation of Γ^- is a challenging task because of the OWA (as described in Section 2). Furthermore, with respect to the generation of Γ^-, there are two types of properties: (1) functional and (2) non-functional (see Section 2). Γ^- generation is complex for non-functional properties and relatively simple for functional properties. Table 8 gives an overview of how the different approaches used in this survey generate Γ^-. Following are the methods used in the literature for Γ^- generation of both functional and non-functional properties (also See Table 10).

6.3.1 Random generation. A strategy to create Γ^- is to generate random assertions. In random generation, some approaches also check whether the resulting assertions are part of the knowledge graph or not. Such random assertions are unlikely to be true, and can thus serve as Γ^- [118]. Consider a scenario from our $\mathcal{G}$ excerpt, where we have two assertions involving US presidents (i.e., :B.Obama $\xrightarrow{\text{:president}}$:USA and :G.W.Bush $\xrightarrow{\text{:president}}$:USA), as true assertions taken from a trustworthy knowledge graph. From each true assertion, we can generate many counterexamples by randomly matching any other entity in the domain of the property :president (e.g., any city's mayor, which is also a person) with the :USA entity in a :president relation. For example, :Sadiq Khan $\xrightarrow{\text{:president}}$:USA . Since the :president property is non-functional, the generated incorrect assertions could be false negative as well. Similarly, for the functional property assertion :Hawaii $\xrightarrow{\text{:locatedIn}}$:USA , we take any other country in the range of the property :locatedIn such as :Pakistan, instead of :USA, which will be generated as a counterexample. Since it is a functional property, the generated example will be truly false.

Random generation is good for functional properties. However, in non-functional properties multiple object instances are possible (see Definition 2.8). Ergo, because of the OWA, there is a probability to generate false negative counterexamples. In the following, we describe the approaches that tackle this drawback.

6.3.2 Partial-Closed World Assumption (PCWA). A common approach for generating Γ^- is to use the Partial-Closed World Assumption (PCWA). PCWA is defined in Definition 2.14. From our running example in Figure 1, given a true assertion :B.Obama $\xrightarrow{\text{:spouse}}$:M.Obama. The PCWA-based method could generate counterexamples with adjacent nodes or distant nodes (i.e., path length >=2). For example, :B.Obama $\xrightarrow{\text{:spouse}}$:S.Mirza [38] is possible for a PCWA-based negative sampling method.

6.3.3 Extended Partial Closed World Assumption (E-PCWA). Ortona et al. [95] propose a negative sampling method based on Extended Partial Closed World Assumption (E-PCWA), which generates a counterexample $s \xrightarrow{p} o'$ by replacing o with o', which has the same class as o and is adjacent to s in $\mathcal{G}$ [95]. In this method, semantic relevance

[38] :S.Mirza is a female tennis player from India.

Table 10. Counterexamples generation methods for the supervised fact-checking approaches; the abbreviations are: E−PCWA/Extended Partial-Closed World Assumption, D−PCWA/Distant Partial-Closed World Assumption, LCA/Least Common Ancestors, PPR/Personalized Pagerank.

Approach	Year	Γ^- gen. method	Approach	Year	Γ^- gen. method
AMIE [41]	2013	PCWA	PRA [44]	2014	Random
FSPG [85]	2015	LCA	AMIE+ [40]	2015	PCWA
GPARs [36]	2015	PCWA	SFE [43]	2015	PPR
KGMiner [112]	2016	Random	FACTY [71]	2017	PCWA
GFC [76]	2018	WPCA	OGFC [77]	2018	WPCA
RuDiK [95]	2018	E−PCWA	MAP [1]	2019	PCWA
AMIE−3.0 [66]	2020	PCWA	KV−Rule [58]	2020	D−PCWA

is ensured by the constraint that both entities must be directly connected by some property p [95]. From our running example in Figure 1, given a true assertion :B.Obama $\xrightarrow{\text{:spouse}}$:M.Obama, the E−PCWA-based method could generate :B.Obama $\xrightarrow{\text{:spouse}}$:M.A.Obama as a counterexample, because :B.Obama and :M.A.Obama are adjacent nodes (i.e., path length = 1). It is possible that this method also generates false negatives. For example, :Ronaldo has signed contracts with multiple teams at different points in time (i.e., :runningContractSigned relation with multiple teams in $\mathcal{G}$), so the E−PCWA-based approach could easily generate a counterexample that is actually true. The drawback of the E−PCWA-based negative sampling approach is that it suffers from generation of counterexamples that are actually not false, but true. For example, at least 47.54% of the counterexamples for relatives generated by E−PCWA-based negative sampling are actually true, according to an analysis presented in [58]. In order to counter these drawbacks, Kim et al. [58] propose the Distant Partial Closed World Assumption.

6.3.4 Distant Partial Closed World Assumption (D−PCWA). D−PCWA states that there may be other possible values next to a given subject and no other possible values far from the given subject if the knowledge graph contains one or more object values for a given subject and property [58]. We formalize the definition as follows: given a true assertion $s \xrightarrow{p} o$, D−PCWA generates a counterexample $s \xrightarrow{p} o'$ by replacing o with o' that has the same class and a distance between 2 and a given maximum distance from s [58]. From our running example in Figure 1, the D−PCWA-based method could generate :B.Obama $\xrightarrow{\text{:spouse}}$:L.Bush as a counterexample from our knowledge graph excerpt, because they are connected by some path, and the path length between :L.Bush and :B.Obama is greater than or equal to 2.

6.3.5 Lowest Common Ancestor (LCA). A disadvantage of the approaches described above is that the counterexamples could be totally irrelevant from the positive example. However, in LCA-based approaches, the counterexamples are generated by randomly replacing entities from the assertion that share the same Lowest Common Ancestor class (i.e., belong to the same class hierarchy) [85]. From our running example in Figure 1, the LCAof :B.Obama is :President. :J.W.Bush or :B.Clinton could be selected at random because they share the same LCA. Similarly, for :M.Obama and :L.Bush, the LCAis :FirstLady. These selected nodes are then randomly combined to form counterexamples like :B.Obama $\xrightarrow{\text{:spouse}}$:L.Bush . However, this approach can also produce false negatives (e.g., :B.Obama $\xrightarrow{\text{:spouse}}$:M.Obama).

Hence, we conclude from our discussion that there is no single effective method for the generation of training counterexamples. Each method has certain advantages and disadvantages.

7 DISCUSSION AND RESULTS COMPARISON

In this section, we provide a summary of published evaluation results for supervised and unsupervised approaches.

7.1 Effectiveness of unsupervised approaches

Syed et al. propose COPAAL, which is to the best of our knowledge the state-of-the-art unsupervised fact-checking approach [122]. They compared the state-of-the-art path-based approaches in their evaluation. We include some parts of their evaluation in Table 11 and Table 12. The numbers show that their approach outperforms all other unsupervised path-based approaches on real-world and synthetic datasets. For example, COPAAL achieves an AUROC that is at least 0.2 higher than that of KS when handling "Birth Place" triples on a real-world dataset. Syed et al. observe in their experiments that the anchored predicate paths $\mathcal{A}_{(k,C(s),C(o))}$ restrict KGMiner's performance by selecting only paths to the same types of subjects and objects as the input triple. Ergo, KGMiner sometimes fails to generalize well. Furthermore, KL-REL uses only the best path to validate assertions, which can limit its ability to validate assertions because one path may not be sufficient to provide strong evidence for a given assertion. In contrast to that, COPAAL identifies multiple paths and assigns weights to them. In Table 11 and Table 12, we observe that in the "Death Place" category, COPAAL has slightly lower scores as compared to other approaches. COPAAL's authors explain that this behavior is caused by numerous available paths, which lead to over-optimistic ratings for some assertions [122]. We can conclude that choosing many vs. fewer paths has an impact on the effectiveness of the fact-checking task.

Syed et al. [122] perform additional experiments on assertions from the FactBench dataset that comprise DBpedia IRIs—dubbed FactBench-DBpedia—to generalize the aforementioned insights. The results on FactBench-DBpedia in Table 13 confirm their findings. COPAAL achieves a better AUROC on almost all categories of Factbench-DBpedia and, hence, outperforms the compared unsupervised approaches.[39]

Syed et al. [122] compare efficiency in terms of runtime. They measure the average throughput of all approaches from end to end. The fastest approach is KL-REL with 29.78 triples/min. COPAAL is placed second with 21.02 triples/min. while KS achieves an average throughput of 10.05 triples/min. We can conclude that COPAAL's runtime is comparable to the other approaches while achieving significantly better results.

Table 11. Performance (AUROC) results of path-based approaches on real-world datasets [122].

	Birth Place	Death Place	Education	Nationality
COPAAL	**0.9441**	0.8997	**0.8731**	**0.9831**
KS	0.7197	0.8002	0.8651	0.9789
KL-REL	0.9254	**0.9095**	0.8547	0.9692

Table 12. Performance (AUROC) results of path-based approaches on synthetic dataset [122].

	US-CAP	NBA-Team	Oscars	CEO	US-WAR	US-VP	FLOTUS
COPAAL	**1.000**	**0.999**	**0.995**	**0.912**	**0.999**	**0.953**	**1.000**
KS	**1.000**	**0.999**	0.950	0.811	0.865	0.798	0.980
KL-REL	**1.000**	**0.999**	0.976	0.898	0.636	0.746	0.986

[39] Syed et al. used a Wilcoxon signed rank test to ensure the significance of the results.

Table 13. Performance (AUROC) results of path-based approaches on *FactBench-DBpedia* datasets [122].

	Domain	DomainRange	Mix	Property	Random	Range
COPAAL	**0.9348**	**0.9389**	**0.8561**	**0.7307**	**0.9411**	**0.8937**
KL-REL	0.8453	0.8619	0.7721	0.6154	0.8547	0.8219
KS	0.8019	0.8124	0.7215	0.6047	0.7911	0.8047

7.2 Effectiveness of supervised approaches

In Figure 7, we show a comparison of accuracy (prediction rate), precision, recall, and F1-score for supervised fact-checking approaches on different datasets based on the evaluation results of Lin et al. [76]. Lin et al. [76] propose the rule-mining-based approach $OFact_R$ and the supervised path-based approach $OFact$. They provide a comparison of their approaches to the state-of-the-art rule-mining-based approach, i.e., AMIE+ [40]; and supervised path-based approaches, i.e., PRA [44] and KGMiner [112]). For their experiments, they use four real-world knowledge graphs, i.e., MAG [135], DBpedia [8], Wikidata [83], and Yago [117]. We include their comparison results in our survey and extend it with results for SFE. Lin et al. try different configurations of the single approaches and only report the best results. We did the same for SFE. SFE inputs include the radius of the subgraphs to be sampled, the type of path extraction method, and the feature type. We set the radius value to 2, BFS as a path extraction method (a bidirectional BFS from the source and the target nodes), and selected the same feature type as PRA.[40] All the supervised path-based approaches within this comparison use a logistic regression model for the classification task. For a fair comparison, the same train and test assertions are used for all approaches. *OFact* achieves the highest accuracy across all datasets. It outperforms $OFact_R$ by 4–21%, KGMiner by 2–14%, SFE by 8–34%, PRA by 2–13%, and AMIE+ by 2–28% higher accuracy values. Also, *OFact* achieves the highest precision, except for the MAG dataset where PRA achieves a slightly higher precision. PRA's results indicate that it is insensitive to the knowledge graph size by sampling only a fixed number of paths. SFE uses subgraphs for the fact-checking task, and it has a simple and inherent computation model. However, it does not perform as well as PRA for large-scale knowledge graphs such as Yago, DBpedia, and Wikidata. For the MAG dataset, the same accuracy is observed for both SFE and PRA. We also observed that varying the bound size for AMIE+, i.e., rule size bound, and SFE, i.e., radius bound for subgraphs, affects the precision values, i.e., higher bounds lead to less precision.[41] For AMIE+ and SFE, the larger bound size could lead to more rules, which expedites hits over false assertions. We can conclude that for a fixed number of training assertions, increasing the bound size can lead to an overfitting situation that reduces the overall precision values. In general, SFE and AMIE+ achieve a high recall, but they have low accuracy and precision on average. It is also worth noticing the F1-score for these approaches, due to the large difference between the number of true and false assertions. On average, *OFact* achieves the highest F1-score across all datasets. It is followed by SFE, $OFact_R$, AMIE+, KGMiner, and PRA. When comparing only the rule-mining-based approaches AMIE+ and $OFact_R$, $OFact_R$ achieves better precision and F1-score on average. However, AMIE+ achieves higher recall values on all datasets.

Lajus et al. [66] compared the three recent rule-mining-based approaches OP [22], RuDiK [95], and AMIE-3.0 [66]. We also use their evaluation results in our survey. The results of all these approaches are calculated using their default configurations on 3 datasets (YAGO2s, DBpedia 3.8, and a Wikidata dump from December 2019). As it can be observed from Table 14, in general, the OP approach has the longest runtime, and it produces the largest number of rules [22, 23]. We investigate the reason behind this behavior and find out that OP uses heuristics to produce rules similar to the confidence approximation of AMIE+. Afterward, OP generates the support and confidence of the remaining candidates.

[40] The path between source and target node.

[41] These precision results are not shown in this survey.

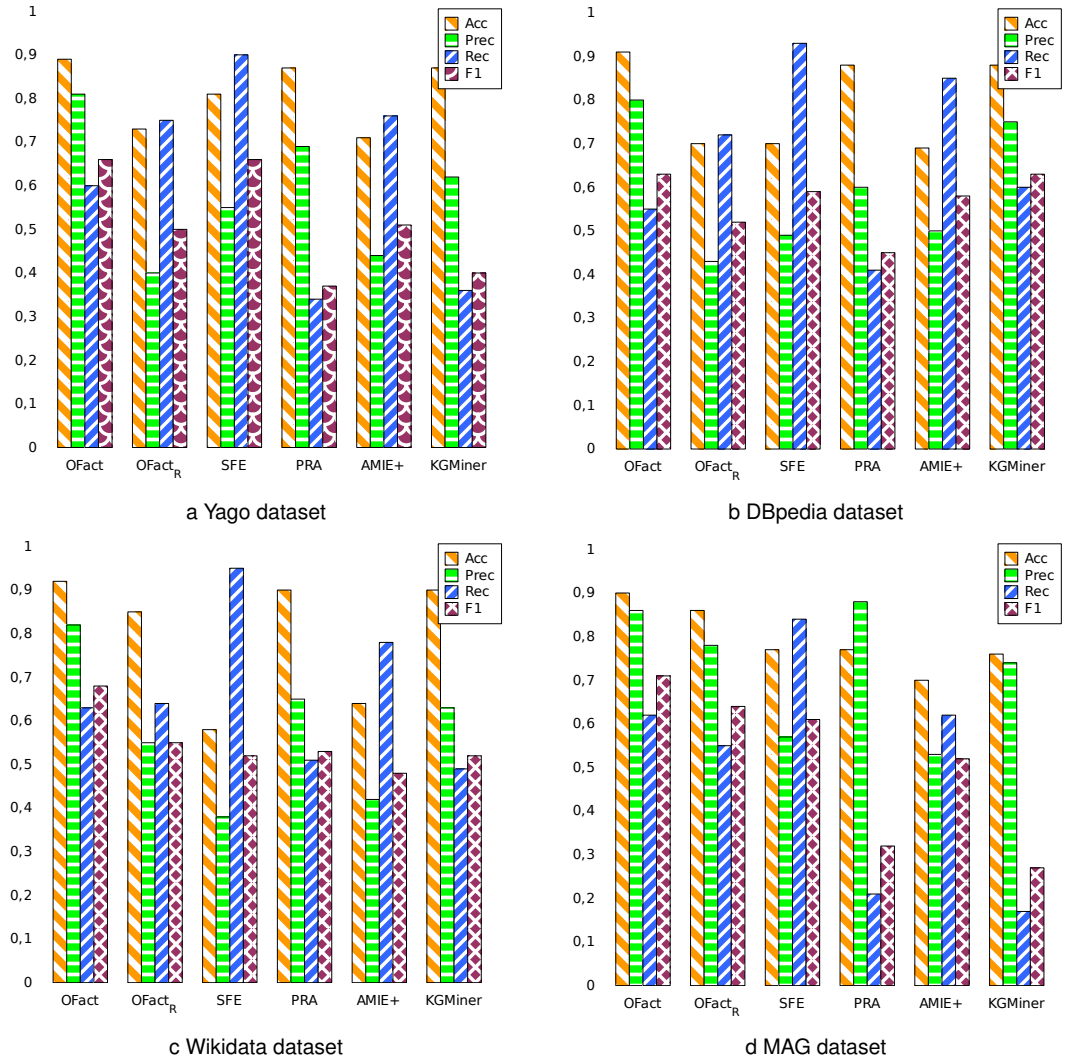


Fig. 7. Comparison of average accuracy (prediction rate), precision, recall and F1-score for supervised fact-checking approaches on different datasets [76].

However, this algorithm does not allow prior pruning based on confidence and support. Due to this nature, the confidence and support of most rules are very low (i.e., $> 90\%$). The computation of these rules with very low confidence and support leads to higher runtime. To reproduce `OP`'s created rules with `AMIE-3.0`, Lajus et al. ran `AMIE-3.0` [66] with a support threshold of 100 and a `CWA` confidence threshold of 0.1. `AMIE-3.0` reproduces all the rules produced by `OP` in less than 2 minutes and added even more rules. This runtime improvement is achieved due to the heuristic used in `AMIE-3.0`. However, when Lajus et al. [66] ran `AMIE-3.0` with a support of 1 and a minimal `CWA` confidence of 10^{-5}, they mined more rules than `OP` on the Yago2s dataset in less time (factor $25x$). On DBpedia and Wikidata, `RuDiK` mined rules in 12 and 23 hours, respectively. In comparison, `RuDiK` mined fewer rules because it could not generate positive and negative

examples of certain relations. On large datasets like Dbpedia and Wikidata, AMIE−3.0 is still faster and mines more rules than the other approaches.

A limitation of the rule-mining-based approaches is that they load the entire graph into memory in order to discover a set of rules. RuDiK proposes a different approach by loading only a chunk of the knowledge graph into the memory [95]. As we can observe from Table 14, RuDiK is faster than OP. RuDiK also uses a parallel processing architecture [95]. However in [66], authors observe that RuDiK's original parallelization mechanism does not scale well to more than 5 cores. Similarly, the average load on a system executing AMIE−3.0 on Yago, DBpedia, and Wikidata on a virtuoso server never exceeds 5 cores. Lajus et al. [66] conclude that even though RuDiK utilizes more cores, AMIE−3.0 is still a faster approach. Additionally, AMIE−3.0 was able to mine some relations that are not even covered by RuDiK or OP. For example, AMIE−3.0 mined the relation :inCountry as $(?x \xrightarrow{:inRegion} ?y \xrightarrow{:inCountry} ?z) \Rightarrow (?x \xrightarrow{:inCountry} ?z)$. However, RuDiK misses it, even though this rule has high confidence and support values [66].

We conclude from this evaluation that AMIE−3.0 is the fastest approach currently available and it finds the largest set of rules. However, it has to load the whole graph into main memory. The size of the Wikidata version that Lajus et al. [66] evaluated is 500 GB, which consumed all their system's available memory, i.e., 500GB. So this kind of approach may not be suitable for very large knowledge graphs on smaller machines, because more resources will be required for the rule-mining process. RuDiK on the other hand is based on a parallel architecture and does not load the entire graph in memory, which makes it suitable for rule-mining on large knowledge graphs on relatively smaller machines with fewer resources.

Table 14. Runtime performance and output of different rule mining approaches. *:rules with support ≥ 100 and CWA confidence ≥ 0.1 [66].

Dataset	System	Rules	Runtimes
Yago2s	OP [22]	1348 (96*)	3h 20min
	RuDiK [95]	17	37min30s
	AMIE−3.0 [66]	97	**1min50s**
	AMIE−3.0 [66](support=1)	1596	7min6s
DBpedia 3.8	OP [22]	7714 (220*)	≥ 45h
	RuDiK [95]	650	12h 10min
	AMIE−3.0 [66]	5084	**7min 52s**
	AMIE−3.0 [66](support=1)	132958	32min 57s
Wikidata 2019	OP [22]	15999 (326*)	≥ 48h
	RuDiK [95]	1145	23h
	AMIE−3.0 [66]	8662	**16h 43min**

8 RELATED FIELDS OF RESEARCH

There are some fields of research that are closely related to fact checking and are worthy to be considered for the discussion in this survey. These fields include the following.

8.0.1 Trustworthiness of sources. Trustworthiness is a key feature of text-based fact-checking and fake news detection approaches. Trustworthiness can be of three types: information−based, theoretical, and reputation−based [98]. Information−based trustworthiness is computed from sources making conflicting claims, by iteratively updating

their values based on certain parameters, such as calculating belief in facts depending on trust in their sources, and trust in sources depending on belief in their facts [141]. Dong et al. propose an extension to this approach by pre-calculating the source dependence from other sources. Hence, independent sources get higher credibility [32, 33]. Marsh et al. propose that the `theoretical` trust can be of three categories [84]: (1) global trust, i.e., user's rating system, (2) personal trust, i.e., each person has some trust values based on their prior knowledge, and (3) situational trust, i.e., specific to some context. Many algorithms are based on global trust [141, 142], however some algorithms, such as [98], use personal trust by exploiting their prior knowledge. Lastly, the approaches based on `reputation-based` trust determine source trust by computing reputation among other sources. For example, the PageRank [97] algorithm ranks web pages. The difference between `reputation-based` and `information-based` is that algorithms of the latter category use the content rather than peer recommendation in the former approach. Furthermore, Nakamura et al. propose a different approach to enhance search engine results based on the trustworthiness analysis of those results [92]. A survey was conducted to determine how often users use search engines and how much they trust the content and rankings. Nakamura et al. calculate the trustworthiness of search results using several criteria, such as locality of supporting pages, topic coverage, topic majority, and others [92]. Consequently, we conclude that their approach combines `information-` and `reputation-based` trustworthiness.

8.0.2 Truth Discovery. Truth discovery aims to identify true claims from a set of conflicting claims made by several sources [14]. Beretta et al. apply this to knowledge graphs [12]. They calculate the trustworthiness of sources and the confidence for conflicting assertions using an existing rule-mining-based algorithm [41].The assertions with the highest confidence are used to populate the knowledge graph.

8.0.3 Surveys in related disciplines. Different surveys emerged in journalism, claim or stance detection and verification, fake news detection, and evidence classification. Konstantinovskiy et al. provide definitions for claim detection proposed in the literature and also propose an updated definition based on commonalities from existing ones [62]. They also discuss the challenges faced in identifying and defining trustworthy claims for the fact-checking task on the web. Thorne et al. discuss and analyze fact-checking definitions, methodologies, and conceptualization [128]. They also discuss knowledge graph-based approaches. However, recent fact-checking approaches and path-based approaches in general, are missing from their survey. Zhou et al. consider the task of fake news identification in their survey [144]. It focuses on defining fake news and presenting related fundamental theories. Yalian et al. perform a comprehensive overview of truth discovery methods in their survey [75]. They also discuss the application aspect of these methods. All these surveys focus on claim identification, fake news detection, or evidence classification. However, in this survey, we discuss and analyze recent trends in fact checking over knowledge graphs.

9 RESEARCH FINDINGS AND FUTURE FORECASTING

In the following, we present a list of open research challenges that are important for future research in the research domain of fact checking over knowledge graphs:

- Knowledge graph veracity: To the best of our knowledge, there is currently no approach to compute the veracity of a complete knowledge graph efficiently and effectively. Current solutions check the veracity of a single assertion and take longer time on average when using a large-scale knowledge graph as the background knowledge, e.g., 2-3 mins. on average for `COPAAL` [122].

- Bias in fact-checking approaches: The majority of the fact-checking approaches rely on reference background knowledge in the form of a knowledge graph $\mathcal{G}$or a corpus, such as DBpedia as $\mathcal{G}$or Wikipedia as a corpus. Some approaches are trained for a particular dataset, such as PolitiFact[131] or CredBank[88]. These datasets and reference background knowledge are often curated by a small group of people and often annotated by non-experts, resulting in bias in the overall fact-checking process.
- Temporal assertions: Several knowledge graphs, such as ICEWS [69], GDELT [70], Wikidata [83] and Yago [117], contain temporal information augmented with assertions. However, some knowledge graphs, such as MAG [135], fail to provide such information. Considering some assertions are time sensitive (e.g., `:B.Obama` $\xrightarrow{\texttt{:president}}$ `:USA`), it would be highly relevant to know the temporal aspect of such assertions, i.e., $[2009 - 2017]$. To the best of our knowledge, the only work that looks into the temporal aspect of the fact-checking task is Temporal-DeFacto [45]. However, it faces the problem of manual feature engineering [105]. There is a need for better fact-checking datasets that cover temporal assertions as well, in order to perform time-aware fact checking.
- Ontology support: Most real-world knowledge graphs, such as MAG [135] and Offshore [73], do not provide ontological information, e.g. class information. Nevertheless, many fact-checking approaches rely on RDFS semantics, such as RDF class information [43, 67, 112, 114, 122]. How can a fact-checking algorithm find the veracity of the given assertion, provided the ontological information is missing? One possible solution is to infer this information from the knowledge graph itself, but most current fact-checking approaches fail to do so.
- Ambiguity of the claims: The generation of meaningful explanations is a challenging task. Assertions without proper explanations are ambiguous. Only a few approaches deal with this problem by generating explanations of the results. Kotonya et al. and Atanasova et al. [6, 65] provide natural language explanations of the claims. Few other approaches provide explanations of assertions via horn rules [1, 38, 39]. `COPAAL` offers the generation of explanations based on the found paths [122, 125]. But these approaches are still not up to the mark for non-technical users. A better explanation module is required to provide better evidence for non-technical users, such as journalists, bloggers, and analysts.
- Rule-mining-based solutions limitations: Rule-mining-based approaches experience a trade-off between the expressiveness of rules discovered and the runtime [41, 95]. How to handle this issue smartly remains an interesting research direction.
- Sources and subjectivity: Not all information is trustworthy, and sometimes trustworthy sources contradict each other. Most of the fact-checking approaches rely on Wikipedia (for unstructured reference knowledge-based solutions) or DBpedia (for structured reference knowledge-based solutions). As an example, in FactCheck and Temporal-DeFacto the justification for the labels (True/False) is restricted to sentences selected from a local or web corpora (i.e., Wikipedia and ClueWeb[42]); and in the case of `KS`, `KL`, `KL-REL`, and `COPAAL`, DBpedia is used as a reference corpus. Hence, there is a need for reliable or general sources or even multiple corpora.
- Evaluation of fact-checking task: AUROC and PR (Precision-Recall) curves are widely used metrics for the binary classification of assertions in fact-checking approaches. These approaches label assertions as true or false. The decision made by the approaches can be represented in a confusion matrix or contingency table. The confusion matrix has four categories: True positives (TP) are assertions correctly categorized as true. False positives (FP) refer to false assertions incorrectly categorized as true. True negatives (TN) correspond to negatives correctly labeled as false. Finally, false negatives (FN) refer to true assertions incorrectly labeled as false. The confusion

[42] https://lemurproject.org/clueweb12/

matrix can be used to construct a point in ROC or PR space. These two metrics are related to each other (i.e., equivalent spaces) and are proven to be effective methods for binary classification tasks [28]. However, there is a need for fact-checking platforms or benchmarks, that could evaluate the overall fact-checking task using these metrics.

10 APPLICATIONS

Fact-checking approaches are becoming popular among researchers and industries and are successfully applied in numerous real-world applications. Following, we summarize some applications related to fact checking over knowledge graphs:

- Commercial and open-source knowledge graphs: Many commercial and open source knowledge graphs, such as Google knowledge graph [116], Yago [117], DBpedia [8], and Wikidata [83], have emerged. Most of these knowledge graphs are incomplete and may contain incorrect information [54, 138]. For example, DBpedia version 3.6 is circa 80% correct [45], and Yago is 98% correct.[43] However, these stats are computed either manually or semi-automatically. To the best of our knowledge, there is no automatic fact-checking approach to compute the veracity of a complete knowledge graph.
- Healthcare: Knowledge graphs in healthcare are also emerging. Healthcare information quality is a critical issue, which is very essential and is being addressed using fact-checking approaches [65]. For example, SemEHR is an AI-based healthcare system, which relies on knowledge graphs as background knowledge [139].
- Information extraction: For the further growth of knowledge graphs, the extraction of structured data from unstructured or semi-structured sources is important. When information is extracted from multiple sources, the results could be conflicting. Therefore, fact-checking algorithms have been incorporated to identify and resolve these conflicts [99].
- Crowdsourcing: Crowdsourcing platforms provide an efficient way to get solicit veracity labels from the crowd [75]. However, the output quality is quite diverse. Therefore, fact-checking methods are being developed to aggregate labels and learn true labels [10, 29].
- Crowd sensing: Crowd sensing provides information about the physical world (e.g., road closure, gas shortage, or a disaster) [86]. Fact-checking approaches are developed to determine the veracity of such information [134, 136].

11 CONCLUDING REMARKS

Fact checking over knowledge graphs quickly gained massive attention and found important applications in various domains. This article provides a systematic review, research issues, merits, and demerits of available fact-checking approaches for knowledge graphs that support fact-checkers. We divided these approaches into different categories, i.e., (1) textual-reference-knowledge-based, (2) structured-reference-knowledge-based, and (3) hybrid approaches. Then we performed a comprehensive comparison of these categories in terms of used techniques, efficiency, and effectiveness. We also compared different datasets and benchmarks proposed in the fact-checking literature. We summarize the findings of our analysis as follows:

- Path features are quite discriminant in finding the veracity of the given assertion. Ergo, subgraphs and path-based methods often report higher prediction rates.

[43] https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/statistics

- In unsupervised meta-path-based approaches, `COPAAL` outperforms other approaches due to the corroborated path exploration using ontological features in the knowledge graph (Table 11 and 12).
- In rule-mining-based approaches, `AMIE-3.0` is the fastest approach that discovers more expressive rules as compared to other approaches (Table 14).
- We suggest several research directions, which might receive increasing attention in the near future.

We hope this brief exploration into the domain of fact checking over knowledge graph can provide new insights into future applications of fact checking.

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 and from the German Federal Ministry of Education and Research (BMBF) within the project NEBULA under the grant no 13N16364.

REFERENCES

[1] Naser Ahmadi, Paolo Papotti, and Mohammed Saeed. 2019. Explainable fact checking with probabilistic answer set programming. In *TTO 2019, Conference for truth and trust Online, 4-5 October 2019, London, UK*. London, UNITED KINGDOM. http://www.eurecom.fr/publication/5942

[2] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. 1993. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc., USA.

[3] Tariq Alhindi, Savvas Petridis, and Smaranda Muresan. 2018. Where is Your Evidence: Improving Fact-checking by Justification Modeling. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*. Association for Computational Linguistics, Brussels, Belgium, 85–90. https://doi.org/10.18653/v1/W18-5513

[4] Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. FEVEROUS: Fact Extraction and VERification Over Unstructured and Structured information. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*. https://openreview.net/forum?id=h-flVCIlstW

[5] Edoardo Amaldi and Viggo Kann. 1998. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science* 209, 1 (1998), 237–260. https://doi.org/10.1016/S0304-3975(97)00115-1

[6] Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020. Generating Fact Checking Explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 7352–7364. https://doi.org/10.18653/v1/2020.acl-main.656

[7] Ram G. Athreya, Axel-Cyrille Ngonga Ngomo, and Ricardo Usbeck. 2018. Enhancing Community Interactions with Data-Driven Chatbots–The DBpedia Chatbot. In *Companion Proceedings of the The Web Conference 2018* (Lyon, France) *(WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 143–146. https://doi.org/10.1145/3184558.3186964

[8] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A Nucleus for a Web of Open Data. In *The Semantic Web*, Karl Aberer, Key-Sun Choi, Natasha Noy, Dean Allemang, Kyung-Il Lee, Lyndon Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 722–735.

[9] Isabelle Augenstein, Christina Lioma, Dongsheng Wang, Lucas Chaves Lima, Casper Hansen, Christian Hansen, and Jakob Grue Simonsen. 2019. MultiFC: A Real-World Multi-Domain Dataset for Evidence-Based Fact Checking of Claims. 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)*. Association for Computational Linguistics, Hong Kong, China, 4685–4697. https://doi.org/10.18653/v1/D19-1475

[10] Bahadir Ismail Aydin, Yavuz Selim Yilmaz, Yaliang Li, Qi Li, Jing Gao, and Murat Demirbas. 2014. Crowdsourcing for Multiple-Choice Question Answering. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence* (Québec City, Québec, Canada) *(AAAI'14)*. AAAI Press, 2946–2953.

[11] Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. 2004. *OWL Web Ontology Language Reference*. W3C Recommendation. W3C. http://www.w3.org/TR/2004/REC-owl-ref-20040210/

[12] Valentina Beretta, Sébastien Harispe, Sylvie Ranwez, and Isabelle Mougenot. 2018. Combining Truth Discovery and RDF Knowledge Bases to Their Mutual Advantage. In *The Semantic Web – ISWC 2018 (The Semantic Web – ISWC 2018, Information Systems and Applications, incl. Internet/Web, and HCI series, Vol. 11136)*. Monterey, Californie, United States, 652–668. https://hal.archives-ouvertes.fr/hal-01912273

[13] Joanna Biega, Erdal Kuzey, and Fabian M. Suchanek. 2013. Inside YAGO2s: A Transparent Information Extraction Architecture. In *Proceedings of the 22nd International Conference on World Wide Web* (Rio de Janeiro, Brazil) *(WWW '13 Companion)*. Association for Computing Machinery, New York, NY, USA, 325–328. https://doi.org/10.1145/2487788.2487935

[14] Katarina Boland, Pavlos Fafalios, Andon Tchechmedjiev, Stefan Dietze, and Konstantin Todorov. 2021. Beyond Facts - a Survey and Conceptualisation of Claims in Online Discourse Analysis. (Mar 2021). https://hal.mines-ales.fr/hal-03185097 working paper or preprint.

[15] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data* (Vancouver, Canada) *(SIGMOD '08)*. Association for Computing Machinery, New York, NY, USA, 1247–1250. https://doi.org/10.1145/1376616.1376746

[16] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-Relational Data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2* (Lake Tahoe, Nevada) *(NIPS'13)*. Curran Associates Inc., Red Hook, NY, USA, 2787–2795.

[17] Gerlof Bouma. 2009. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL* (2009), 31–40.

[18] Dan Brickley and R.V. Guha. 2004. *RDF Schema 1.1*. W3C Recommendation. W3C. https://www.w3.org/TR/rdf-schema/

[19] Dan Brickley and R.V. Guha. 2014. *RDF Schema 1.1*. W3C Recommendation. W3C. https://www.w3.org/TR/rdf-schema/

[20] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka, and Tom M. Mitchell. 2010. Toward an Architecture for Never-Ending Language Learning. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence* (Atlanta, Georgia) *(AAAI'10)*. AAAI Press, 1306–1313.

[21] Sihao Chen, Daniel Khashabi, Wenpeng Yin, Chris Callison-Burch, and Dan Roth. 2019. Seeing Things from a Different Angle:Discovering Diverse Perspectives about Claims. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 542–557. https://doi.org/10.18653/v1/N19-1053

[22] Yang Chen, Sean Goldberg, Daisy Zhe Wang, and Soumitra Siddharth Johri. 2016. Ontological Pathfinding: Mining First-Order Knowledge from Large Knowledge Bases. In *Proceedings of the 2016 International Conference on Management of Data* (San Francisco, California, USA) *(SIGMOD '16)*. Association for Computing Machinery, New York, NY, USA, 835–846. https://doi.org/10.1145/2882903.2882954

[23] Yang Chen, Daisy Zhe Wang, and Sean Goldberg. 2016. ScaLeKB: Scalable Learning and Inference over Large Knowledge Bases. *The VLDB Journal* 25, 6 (dec 2016), 893–918. https://doi.org/10.1007/s00778-016-0444-3

[24] Giovanni Luca Ciampaglia, Prashant Shiralkar, Luis M. Rocha, Johan Bollen, Filippo Menczer, and Alessandro Flammini. 2015. Computational Fact Checking from Knowledge Networks. *PLOS ONE* 10, 6 (06 2015), 1–13. https://doi.org/10.1371/journal.pone.0128193

[25] Richard Cyganiak, David Wood, and Markus Lanthaler. 2014. *RDF 1.1 Concepts and Abstract Syntax*. W3C Recommendation. W3C. http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/

[26] Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019. Fine-Grained Analysis of Propaganda in News Article. 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)*. Association for Computational Linguistics, Hong Kong, China, 5636–5646. https://doi.org/10.18653/v1/D19-1565

[27] Ana Alexandra Morim da Silva, Michael Röder, and Axel-Cyrille Ngonga Ngomo. 2021. Using Compositional Embeddings for Fact Checking. In *The Semantic Web – ISWC 2021*, Andreas Hotho, Eva Blomqvist, Stefan Dietze, Achille Fokoue, Ying Ding, Payam Barnaghi, Armin Haller, Mauro Dragoni, and Harith Alani (Eds.). Springer International Publishing, Cham, 270–286. https://papers.dice-research.org/2021/ISWC2021%5FEsther/ESTHER%5Fpublic.pdf

[28] Jesse Davis and Mark Goadrich. 2006. The Relationship between Precision-Recall and ROC Curves. In *Proceedings of the 23rd International Conference on Machine Learning* (Pittsburgh, Pennsylvania, USA) *(ICML '06)*. Association for Computing Machinery, New York, NY, USA, 233–240. https://doi.org/10.1145/1143844.1143874

[29] A. P. Dawid and A. M. Skene. 1979. Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28, 1 (1979), 20–28. http://www.jstor.org/stable/2346806

[30] Caglar Demir and Axel-Cyrille Ngonga Ngomo. 2021. Convolutional complex knowledge graph embeddings. In *European Semantic Web Conference*. Springer, 409–424.

[31] Thomas Diggelmann, Jordan Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leippold. 2021. CLIMATE-FEVER: A Dataset for Verification of Real-World Climate Claims. arXiv:2012.00614 [cs.CL]

[32] Xin Luna Dong, Laure Berti-Equille, and Divesh Srivastava. 2009. Integrating conflicting data: the role of source dependence. *Proceedings of the VLDB Endowment* 2, 1 (2009), 550–561.

[33] Xin Luna Dong, Laure Berti-Equille, and Divesh Srivastava. 2009. Truth Discovery and Copying Detection in a Dynamic World. *Proc. VLDB Endow.* 2, 1 (Aug. 2009), 562–573. https://doi.org/10.14778/1687627.1687691

[34] Ivan Ermilov, Jens Lehmann, Michael Martin, and Sören Auer. 2016. LODStats: The Data Web Census Dataset. In *The Semantic Web – ISWC 2016*, Paul Groth, Elena Simperl, Alasdair Gray, Marta Sabou, Markus Krötzsch, Freddy Lecue, Fabian Flöck, and Yolanda Gil (Eds.). Springer International Publishing, Cham, 38–46.

[35] Diego Esteves, Anisa Rula, Aniketh Janardhan Reddy, and Jens Lehmann. 2018. Toward Veracity Assessment in RDF Knowledge Bases: An Exploratory Analysis. *J. Data and Information Quality* 9, 3, Article 16 (feb 2018), 26 pages. https://doi.org/10.1145/3177873

[36] Wenfei Fan, Xin Wang, Yinghui Wu, and Jingbo Xu. 2015. Association Rules with Graph Patterns. *Proc. VLDB Endow.* 8, 12 (Aug. 2015), 1502–1513. https://doi.org/10.14778/2824032.2824048

[37] William Ferreira and Andreas Vlachos. 2016. Emergent: a novel data-set for stance classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, 1163–1168. https://doi.org/10.18653/v1/N16-1138

[38] Mohamed H. Gad-Elrab, Daria Stepanova, Jacopo Urbani, and Gerhard Weikum. 2019. 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* (Melbourne VIC, Australia) *(WSDM '19)*. Association for Computing Machinery, New York, NY, USA, 87–95. https://doi.org/10.1145/3289600.3290996

[39] Mohamed H. Gad-Elrab, Daria Stepanova, Jacopo Urbani, and Gerhard Weikum. 2019. Tracy: Tracing Facts over Knowledge Graphs and Text. In *The World Wide Web Conference* (San Francisco, CA, USA) *(WWW '19)*. Association for Computing Machinery, New York, NY, USA, 3516–3520. https://doi.org/10.1145/3308558.3314126

[40] Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian M. Suchanek. 2015. Fast Rule Mining in Ontological Knowledge Bases with AMIE+. *The VLDB Journal* 24, 6 (Dec. 2015), 707–730. https://doi.org/10.1007/s00778-015-0394-1

[41] Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. 2013. AMIE: Association Rule Mining under Incomplete Evidence in Ontological Knowledge Bases. In *Proceedings of the 22nd International Conference on World Wide Web* (Rio de Janeiro, Brazil) *(WWW '13)*. Association for Computing Machinery, New York, NY, USA, 413–422. https://doi.org/10.1145/2488388.2488425

[42] Susmita Gangopadhyay, Katarina Boland, Sascha Schüller, Konstantin Todorov, Andon Tchechmedjiev, Benjamin Zapilko, Pavlos Fafalios, Hajira Jabeen, and Stefan Dietze. 2022. ClaimsKG - A Knowledge Graph of Fact-Checked Claims (August, 2022). GESIS - Leibniz Institut für Sozialwissenschaften. Datenfile Version 1.0. https://doi.org/10.7802/2469

[43] Matt Gardner and Tom Mitchell. 2015. Efficient and expressive knowledge base completion using subgraph feature extraction. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 1488–1498.

[44] Matt Gardner, Partha Talukdar, Jayant Krishnamurthy, and Tom Mitchell. 2014. 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)*. Association for Computational Linguistics, Doha, Qatar, 397–406. https://doi.org/10.3115/v1/D14-1044

[45] Daniel Gerber, Diego Esteves, Jens Lehmann, Lorenz Bühmann, Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, and René Speck. 2015. DeFacto-Temporal and Multilingual Deep Fact Validation. *Web Semant.* 35, P2 (Dec. 2015), 85–101. https://doi.org/10.1016/j.websem.2015.08.001

[46] Daniel Gerber and Axel-Cyrille Ngonga Ngomo. 2011. Bootstrapping the Linked Data Web. In *1st Workshop on Web Scale Knowledge Extraction @ ISWC 2011*.

[47] Bart Goethals and Jan Van den Bussche. 2002. Relational Association Rules: Getting WARMeR. In *Proceedings of the ESF Exploratory Workshop on Pattern Detection and Discovery*. Springer-Verlag, Berlin, Heidelberg, 125–139.

[48] Maurício Gruppi, Benjamin Horne, and Sibel Adalı. 2021. NELA-GT-2020: A Large Multi-Labelled News Dataset for The Study of Misinformation in News Articles. *ArXiv* abs/2102.04567 (2021).

[49] Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A Survey on Automated Fact-Checking. *Transactions of the Association for Computational Linguistics* 10 (02 2022), 178–206. https://doi.org/10.1162/tacl_a_00454 arXiv:https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00454/1987018/tacl_a_00454.pdf

[50] Ashim Gupta and Vivek Srikumar. 2021. X-Fact: A New Benchmark Dataset for Multilingual Fact Checking. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Association for Computational Linguistics, Online, 675–682. https://doi.org/10.18653/v1/2021.acl-short.86

[51] Andreas Hanselowski, Christian Stab, Claudia Schulz, Zile Li, and Iryna Gurevych. 2019. A Richly Annotated Corpus for Different Tasks in Automated Fact-Checking. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. Association for Computational Linguistics, Hong Kong, China, 493–503. https://doi.org/10.18653/v1/K19-1046

[52] Naeemul Hassan, Chengkai Li, and Mark Tremayne. 2015. Detecting Check-Worthy Factual Claims in Presidential Debates. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management* (Melbourne, Australia) *(CIKM '15)*. Association for Computing Machinery, New York, NY, USA, 1835–1838. https://doi.org/10.1145/2806416.2806652

[53] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. 2013. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artificial Intelligence* 194 (2013), 28–61. https://doi.org/10.1016/j.artint.2012.06.001 Artificial Intelligence, Wikipedia and Semi-Structured Resources.

[54] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D'amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. 2021. Knowledge Graphs. *ACM Comput. Surv.* 54, 4, Article 71 (July 2021), 37 pages. https://doi.org/10.1145/3447772

[55] Ali Hur, Naeem Janjua, and Mohiuddin Ahmed. 2021. A Survey on State-of-the-art Techniques for Knowledge Graphs Construction and Challenges ahead. In *2021 IEEE Fourth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*. 99–103. https://doi.org/10.1109/AIKE52691.2021.00021

[56] Viet-Phi Huynh and Paolo Papotti. 2018. Towards a Benchmark for Fact Checking with Knowledge Bases. In *Companion Proceedings of the The Web Conference 2018* (Lyon, France) *(WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1595–1598. https://doi.org/10.1145/3184558.3191616

[57] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. 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)*. Association for Computational Linguistics, Beijing, China, 687–696. https://doi.org/10.3115/v1/P15-1067

[58] Jiseong Kim and Key-sun Choi. 2020. 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*. International Committee on Computational Linguistics, Barcelona, Spain (Online), 1677–1686. https://doi.org/10.18653/v1/2020.coling-main.147

[59] Barbara Kitchenham. 2004. Procedures for performing systematic reviews. *Keele, UK, Keele University* 33, 2004 (2004), 1–26.

[60] Jon M. Kleinberg. 1999. Authoritative Sources in a Hyperlinked Environment. *J. ACM* 46, 5 (Sept. 1999), 604–632. https://doi.org/10.1145/324133.324140

[61] Daphne Koller and Mehran Sahami. 1996. Toward Optimal Feature Selection. In *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning* (Bari, Italy) *(ICML'96)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 284–292.

[62] Lev Konstantinovskiy, Oliver Price, Mevan Babakar, and Arkaitz Zubiaga. 2021. Toward Automated Factchecking: Developing an Annotation Schema and Benchmark for Consistent Automated Claim Detection. *Digital Threats: Research and Practice* 2, 2, Article 14 (April 2021), 16 pages. https://doi.org/10.1145/3412869

[63] Andrew Koster, Ana Bazzan, and Marcelo de Souza. 2017. 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

[64] Neema Kotonya and Francesca Toni. 2020. Explainable Automated Fact-Checking: A Survey. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Barcelona, Spain (Online), 5430–5443. https://www.aclweb.org/anthology/2020.coling-main.474

[65] Neema Kotonya and Francesca Toni. 2020. Explainable Automated Fact-Checking for Public Health Claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 7740–7754. https://doi.org/10.18653/v1/2020.emnlp-main.623

[66] Jonathan Lajus, Luis Galárraga, and Fabian Suchanek. 2020. Fast and Exact Rule Mining with AMIE 3. In *The Semantic Web*, Andreas Harth, Sabrina Kirrane, Axel-Cyrille Ngonga Ngomo, Heiko Paulheim, Anisa Rula, Anna Lisa Gentile, Peter Haase, and Michael Cochez (Eds.). Springer International Publishing, Cham, 36–52.

[67] Ni Lao and William Cohen. 2010. Relational Retrieval Using a Combination of Path-Constrained Random Walks. *Machine Learning* 81 (10 2010), 53–67. https://doi.org/10.1007/s10994-010-5205-8

[68] Ni Lao and William W. Cohen. 2010. Relational retrieval using a combination of path-constrained random walks. *Mach. Learn.* 81, 1 (2010), 53–67. https://doi.org/10.1007/s10994-010-5205-8

[69] Jennifer Lautenschlager, Steve Shellman, and Michael Ward. 2015. ICEWS Event Aggregations. https://doi.org/10.7910/DVN/28117

[70] Kalev Leetaru and Philip A. Schrodt. 2013. GDELT: Global data on events, location, and tone. *ISA Annual Convention* (2013).

[71] Furong Li, Xin Luna Dong, Anno Langen, and Yang Li. 2017. Knowledge Verification for Long-Tail Verticals. *Proc. VLDB Endow.* 10, 11 (Aug. 2017), 1370–1381. https://doi.org/10.14778/3137628.3137646

[72] Jia Li, Yang Cao, and Shuai Ma. 2017. Relaxing Graph Pattern Matching With Explanations. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (Singapore, Singapore) *(CIKM '17)*. Association for Computing Machinery, New York, NY, USA, 1677–1686. https://doi.org/10.1145/3132847.3132992

[73] Oliver Zhen Li and Guang Ma. 2019. Offshore Leaks, Taxes and Capital Structure. *Taxes and Capital Structure (February 15, 2019)* (2019).

[74] X. Li, W. Meng, and C. Yu. 2011. T-verifier: Verifying truthfulness of fact statements. In *2011 IEEE 27th International Conference on Data Engineering*. 63–74. https://doi.org/10.1109/ICDE.2011.5767859

[75] Yaliang Li, Jing Gao, Chuishi Meng, Qi Li, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han. 2016. A Survey on Truth Discovery. *SIGKDD Explor. Newsl.* 17, 2 (Feb. 2016), 1–16. https://doi.org/10.1145/2897350.2897352

[76] Peng Lin, Song Qi, Jialiang Shen, and Yinghui Wu. 2018. *Discovering Graph Patterns for Fact Checking in Knowledge Graphs*. 783–801. https://doi.org/10.1007/978-3-319-91452-7_50

[77] Peng Lin, Qi Song, and Yinghui Wu. 2018. Fact Checking in Knowledge Graphs with Ontological Subgraph Patterns. *Data Sci. Eng.* 3, 4 (2018), 341–358. https://doi.org/10.1007/s41019-018-0082-4

[78] Peng Lin, Qi Song, Yinghui Wu, and Jiaxing Pi. 2019. Discovering Patterns for Fact Checking in Knowledge Graphs. *J. Data and Information Quality* 11, 3, Article 13 (May 2019), 27 pages. https://doi.org/10.1145/3286488

[79] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence* (Austin, Texas) *(AAAI'15)*. AAAI Press, 2181–2187.

[80] Yunfei Long, Qin Lu, Rong Xiang, Minglei Li, and Chu-Ren Huang. 2017. Fake News Detection Through Multi-Perspective Speaker Profiles. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Asian Federation of Natural Language Processing, Taipei, Taiwan, 252–256. https://aclanthology.org/I17-2043

[81] Shuai Ma, Yang Cao, Wenfei Fan, Jinpeng Huai, and Tianyu Wo. 2011. Capturing Topology in Graph Pattern Matching. *Proc. VLDB Endow.* 5, 4 (Dec. 2011), 310–321. https://doi.org/10.14778/2095686.2095690

[82] Farzaneh Mahdisoltani, Joanna Biega, and Fabian M. Suchanek. 2015. YAGO3: A Knowledge Base from Multilingual Wikipedias.. In *CIDR*. www.cidrdb.org. http://dblp.uni-trier.de/db/conf/cidr/cidr2015.html#MahdisoltaniBS15

[83] Stanislav Malyshev, Markus Krötzsch, Larry González, Julius Gonsior, and Adrian Bielefeldt. 2018. Getting the Most Out of Wikidata: Semantic Technology Usage in Wikipedia's Knowledge Graph. In *The Semantic Web – ISWC 2018*, Denny Vrandečić, Kalina Bontcheva, Mari Carmen Suárez-Figueroa, Valentina Presutti, Irene Celino, Marta Sabou, Lucie-Aimée Kaffee, and Elena Simperl (Eds.). Springer International Publishing, Cham, 376–394.

[84] Stephen Paul Marsh. 1994. Formalising trust as a computational concept. (1994).

[85] Changping Meng, Reynold Cheng, Silviu Maniu, Pierre Senellart, and Wangda Zhang. 2015. Discovering Meta-Paths in Large Heterogeneous Information Networks. In *Proceedings of the 24th International Conference on World Wide Web* (Florence, Italy) *(WWW '15)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 754–764. https://doi.org/10.1145/2736277.2741123

[86] Chuishi Meng, Wenjun Jiang, Yaliang Li, Jing Gao, Lu Su, Hu Ding, and Yun Cheng. 2015. Truth Discovery on Crowd Sensing of Correlated Entities. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems* (Seoul, South Korea) *(SenSys '15)*. Association for Computing Machinery, New York, NY, USA, 169–182. https://doi.org/10.1145/2809695.2809715

[87] T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, B. Yang, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. 2018. Never-Ending Learning. *Commun. ACM* 61, 5 (apr 2018), 103–115. https://doi.org/10.1145/3191513

[88] Tanushree Mitra and Eric Gilbert. 2021. CREDBANK: A Large-Scale Social Media Corpus With Associated Credibility Annotations. *Proceedings of the International AAAI Conference on Web and Social Media* 9, 1 (Aug. 2021), 258–267. https://doi.org/10.1609/icwsm.v9i1.14625

[89] David Moher, Alessandro Liberati, Jennifer Tetzlaff, Douglas G Altman, Prisma Group, et al. 2009. Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement. *PLoS med* 6, 7 (2009), e1000097.

[90] Eduardo Menezes de Morais and Marcelo Finger. 2013. Probabilistic Answer Set Programming. In *2013 Brazilian Conference on Intelligent Systems*. 150–156. https://doi.org/10.1109/BRACIS.2013.33

[91] Stephen Muggleton. 1996. Learning from positive data. In *International Conference on Inductive Logic Programming*. Springer, 358–376.

[92] Satoshi Nakamura, Shinji Konishi, Adam Jatowt, Hiroaki Ohshima, Hiroyuki Kondo, Taro Tezuka, Satoshi Oyama, and Katsumi Tanaka. 2007. Trustworthiness Analysis of Web Search Results. In *Proceedings of the 11th European Conference on Research and Advanced Technology for Digital Libraries* (Budapest, Hungary) *(ECDL'07, Vol. 4675)*. Springer-Verlag, Berlin, Heidelberg, 38–49. https://doi.org/10.1007/978-3-540-74851-9_4

[93] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich. 2016. A Review of Relational Machine Learning for Knowledge Graphs. *Proc. IEEE* 104, 1 (2016), 11–33. https://doi.org/10.1109/JPROC.2015.2483592

[94] Dan S. Nielsen and Ryan McConville. 2022. MuMiN: A Large-Scale Multilingual Multimodal Fact-Checked Misinformation Social Network Dataset. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Madrid, Spain) *(SIGIR '22)*. Association for Computing Machinery, New York, NY, USA, 3141–3153. https://doi.org/10.1145/3477495.3531744

[95] Stefano Ortona, Venkata Vamsikrishna Meduri, and Paolo Papotti. 2018. RuDiK: Rule Discovery in Knowledge Bases. *Proc. VLDB Endow.* 11, 12 (Aug. 2018), 1946–1949. https://doi.org/10.14778/3229863.3236231

[96] Wojciech Ostrowski, Arnav Arora, Pepa Atanasova, and Isabelle Augenstein. 2021. Multi-Hop Fact Checking of Political Claims. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, Zhi-Hua Zhou (Ed.). International Joint Conferences on Artificial Intelligence Organization, 3892–3898. https://doi.org/10.24963/ijcai.2021/536 Main Track.

[97] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank citation ranking: Bringing order to the web.* Technical Report. Stanford InfoLab.

[98] Jeff Pasternack and Dan Roth. 2010. Knowing What to Believe (When You Already Know Something). In *Proceedings of the 23rd International Conference on Computational Linguistics* (Beijing, China) *(COLING '10)*. Association for Computational Linguistics, USA, 877–885.

[99] Jeff Pasternack and Dan Roth. 2011. Making Better Informed Trust Decisions with Generalized Fact-Finding. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three* (Barcelona, Catalonia, Spain) *(IJCAI'11)*. AAAI Press, 2324–2329.

[100] Jeff Pasternack and Dan Roth. 2013. Latent Credibility Analysis. In *Proceedings of the 22nd International Conference on World Wide Web* (Rio de Janeiro, Brazil) *(WWW '13)*. Association for Computing Machinery, New York, NY, USA, 1009–1020. https://doi.org/10.1145/2488388.2488476

[101] Heiko Paulheim. 2017. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic web* 8, 3 (2017), 489–508.

[102] Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2018. Automatic Detection of Fake News. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, Santa Fe, New Mexico, USA, 3391–3401. https://www.aclweb.org/anthology/C18-1287

[103] Kashyap Popat, Subhabrata Mukherjee, Jannik Strötgen, and Gerhard Weikum. 2017. Where the Truth Lies: Explaining the Credibility of Emerging Claims on the Web and Social Media. In *Proceedings of the 26th International Conference on World Wide Web Companion, Perth, Australia, April 3-7, 2017*, Rick Barrett, Rick Cummings, Eugene Agichtein, and Evgeniy Gabrilovich (Eds.). ACM, 1003–1012. https://doi.org/10.1145/3041021.3055133

[104] Eric Prud'hommeaux and Andy Seaborne. 2006. *SPARQL Query Language for RDF*. W3C Recommendation. W3C. https://www.w3.org/2001/sw/DataAccess/rq23/

[105] Umair Qudus, Michael Röder, Muhammad Saleem, and Axel-Cyrille Ngonga Ngomo. 2022. HybridFC: A Hybrid Fact-Checking Approach for Knowledge Graphs. In *The Semantic Web – ISWC 2022*. Springer International Publishing, Cham. https://papers.dice-research.org/2022/ISWC%5FHybridFC/public.pdf

[106] J. Ross Quinlan. 1990. Learning logical definitions from relations. *Machine learning* 5, 3 (1990), 239–266.

[107] Victoria Rubin, Niall Conroy, Yimin Chen, and Sarah Cornwell. 2016. Fake News or Truth? Using Satirical Cues to Detect Potentially Misleading News. In *Proceedings of the Second Workshop on Computational Approaches to Deception Detection*. Association for Computational Linguistics, San Diego, California, 7–17. https://doi.org/10.18653/v1/W16-0802

[108] Arkadiy Saakyan, Tuhin Chakrabarty, and Smaranda Muresan. 2021. COVID-Fact: Fact Extraction and Verification of Real-World Claims on COVID-19 Pandemic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Online, 2116–2129. https://doi.org/10.18653/v1/2021.acl-long.165

[109] Giovanni Santia and Jake Williams. 2018. BuzzFace: A News Veracity Dataset with Facebook User Commentary and Egos. *Proceedings of the International AAAI Conference on Web and Social Media* 12, 1 (Jun. 2018), 531–540. https://doi.org/10.1609/icwsm.v12i1.14985

[110] Aalok Sathe, Salar Ather, Tuan Manh Le, Nathan Perry, and Joonsuk Park. 2020. Automated Fact-Checking of Claims from Wikipedia. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*. European Language Resources Association, Marseille, France, 6874–6882. https://aclanthology.org/2020.lrec-1.849

[111] Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get Your Vitamin C! Robust Fact Verification with Contrastive Evidence. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 624–643. https://doi.org/10.18653/v1/2021.naacl-main.52

[112] Baoxu Shi and Tim Weninger. 2016. Discriminative Predicate Path Mining for Fact Checking in Knowledge Graphs. *Know.-Based Syst.* 104, C (July 2016), 123–133. https://doi.org/10.1016/j.knosys.2016.04.015

[113] Chuan Shi, Xiangnan Kong, Yue Huang, Philip S. Yu, and Bin Wu. 2014. HeteSim: A General Framework for Relevance Measure in Heterogeneous Networks. *IEEE Transactions on Knowledge and Data Engineering* 26, 10 (2014), 2479–2492. https://doi.org/10.1109/TKDE.2013.2297920

[114] P. Shiralkar, A. Flammini, F. Menczer, and G. L. Ciampaglia. 2017. Finding Streams in Knowledge Graphs to Support Fact Checking. In *2017 IEEE International Conference on Data Mining (ICDM)*. 859–864. https://doi.org/10.1109/ICDM.2017.105

[115] Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. 2020. FakeNewsNet: A Data Repository with News Content, Social Context, and Spatiotemporal Information for Studying Fake News on Social Media. *Big Data* 8, 3 (2020), 171–188. https://doi.org/10.1089/big.2020.0062 arXiv:https://doi.org/10.1089/big.2020.0062 PMID: 32491943.

[116] Amit Singhal. 2012. Introducing the knowledge graph: Things, not strings. https://blog.google/products/search/introducing-knowledge-graph-things-not/.

[117] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A Core of Semantic Knowledge. In *Proceedings of the 16th International Conference on World Wide Web* (Banff, Alberta, Canada) *(WWW '07)*. Association for Computing Machinery, New York, NY, USA, 697–706. https://doi.org/10.1145/1242572.1242667

[118] Fabian M. Suchanek, Jonathan Lajus, Armand Boschin, and Gerhard Weikum. 2019. *Knowledge Representation and Rule Mining in Entity-Centric Knowledge Bases*. Springer International Publishing, Cham, 110–152. https://doi.org/10.1007/978-3-030-31423-1_4

[119] Tangina Sultana and Young-Koo Lee. 2021. Expressive Rule Pattern Based Compression with Ranking in Horn Rules on RDF Style KB. In *2021 IEEE International Conference on Big Data and Smart Computing (BigComp)*. 13–19. https://doi.org/10.1109/BigComp51126.2021.00012

[120] Y. Sun, R. Barber, M. Gupta, C. C. Aggarwal, and J. Han. 2011. Co-author Relationship Prediction in Heterogeneous Bibliographic Networks. In *2011 International Conference on Advances in Social Networks Analysis and Mining*. 121–128. https://doi.org/10.1109/ASONAM.2011.112

[121] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. 2011. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment* 4, 11 (2011), 992–1003.

[122] Zafar Habeeb Syed, Michael Röder, and Axel-Cyrille Ngonga Ngomo. 2019. Unsupervised Discovery of Corroborative Paths for Fact Validation. In *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 11778)*, Chiara Ghidini, Olaf Hartig, Maria Maleshkova, Vojtech Svátek, Isabel F. Cruz, Aidan Hogan, Jie Song, Maxime Lefrançois, and Fabien Gandon (Eds.). Springer, 630–646. https://doi.org/10.1007/978-3-030-30793-6_36

[123] Zafar Habeeb Syed, Michael Röder, and Axel-Cyrille Ngonga Ngomo. 2018. FactCheck: Validating RDF Triples Using Textual Evidence. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (Torino, Italy) *(CIKM '18)*. Association for Computing Machinery, New York, NY, USA, 1599–1602. https://doi.org/10.1145/3269206.3269308

[124] Zafar Habeeb Syed, Nikit Srivastava, M. Röder, and Axel-Cyrille Ngonga Ngomo. 2019. COPAAL - An Interface for Explaining Facts using Corroborative Paths. In *ISWC Satellites*.

[125] Zafar Habeeb Syed, Nikit Srivastava, Michael Röder, and Axel-Cyrille Ngonga Ngomo. 2019. COPAAL - An Interface for Explaining Facts using Corroborative Paths. In *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)*, Mari Carmen Suárez-Figueroa, Gong Cheng, Anna Lisa Gentile, Christophe Guéret, C. Maria Keet, and Abraham Bernstein (Eds.). CEUR-WS.org, 201–204. http://ceur-ws.org/Vol-2456/paper52.pdf

[126] Eugenio Tacchini, Gabriele Ballarin, Marco Della Vedova, Stefano Moret, and Luca Alfaro. 2017. Some Like it Hoax: Automated Fake News Detection in Social Networks.

[127] Andon Tchechmedjiev, Pavlos Fafalios, Katarina Boland, Malo Gasquet, Matthäus Zloch, Benjamin Zapilko, Stefan Dietze, and Konstantin Todorov. 2019. ClaimsKG: A Knowledge Graph of Fact-Checked Claims. In *The Semantic Web – ISWC 2019*, Chiara Ghidini, Olaf Hartig, Maria Maleshkova, Vojtěch Svátek, Isabel Cruz, Aidan Hogan, Jie Song, Maxime Lefrançois, and Fabien Gandon (Eds.). Springer International Publishing, Cham,

309–324.

[128] James Thorne and Andreas Vlachos. 2018. Automated Fact Checking: Task Formulations, Methods and Future Directions. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, Santa Fe, New Mexico, USA, 3346–3359. https://www.aclweb.org/anthology/C18-1283

[129] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a Large-scale Dataset for Fact Extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 809–819. https://doi.org/10.18653/v1/N18-1074

[130] Th'eo Trouillon, Johannes Welbl, Sebastian Riedel, 'Eric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*. 2071–2080.

[131] Andreas Vlachos and Sebastian Riedel. 2014. Fact Checking: Task definition and dataset construction. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*. Association for Computational Linguistics, Baltimore, MD, USA, 18–22. https://doi.org/10.3115/v1/W14-2508

[132] Inna Vogel and Peter Jiang. 2019. Fake News Detection with the New German Dataset "GermanFakeNC". In *Digital Libraries for Open Knowledge*, Antoine Doucet, Antoine Isaac, Koraljka Golub, Trond Aalberg, and Adam Jatowt (Eds.). Springer International Publishing, Cham, 288–295.

[133] David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or Fiction: Verifying Scientific Claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 7534–7550. https://doi.org/10.18653/v1/2020.emnlp-main.609

[134] Dong Wang, Lance Kaplan, Hieu Le, and Tarek Abdelzaher. 2012. On Truth Discovery in Social Sensing: A Maximum Likelihood Estimation Approach. In *Proceedings of the 11th International Conference on Information Processing in Sensor Networks* (Beijing, China) *(IPSN '12)*. Association for Computing Machinery, New York, NY, USA, 233–244. https://doi.org/10.1145/2185677.2185737

[135] Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia. 2020. Microsoft Academic Graph: When experts are not enough. *Quantitative Science Studies* 1, 1 (02 2020), 396–413. https://doi.org/10.1162/qss_a_00021 arXiv:https://direct.mit.edu/qss/article-pdf/1/1/396/1760880/qss_a_00021.pdf

[136] Shiguang Wang, Lu Su, Shen Li, Shaohan Hu, Tanvir Amin, Hongwei Wang, Shuochao Yao, Lance Kaplan, and Tarek Abdelzaher. 2015. Scalable Social Sensing of Interdependent Phenomena. In *Proceedings of the 14th International Conference on Information Processing in Sensor Networks* (Seattle, Washington) *(IPSN '15)*. Association for Computing Machinery, New York, NY, USA, 202–213. https://doi.org/10.1145/2737095.2737114

[137] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence* (Québec City, Québec, Canada) *(AAAI'14)*. AAAI Press, 1112–1119.

[138] Gabriel Weaver, Barbara Strickland, and Gregory Crane. 2006. Quantifying the Accuracy of Relational Statements in Wikipedia: A Methodology. In *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries* (Chapel Hill, NC, USA) *(JCDL '06)*. Association for Computing Machinery, New York, NY, USA, 358. https://doi.org/10.1145/1141753.1141853

[139] Honghan Wu, Giulia Toti, Katherine I Morley, Zina M Ibrahim, Amos Folarin, Richard Jackson, Ismail Kartoglu, Asha Agrawal, Clive Stringer, Darren Gale, Genevieve Gorrell, Angus Roberts, Matthew Broadbent, Robert Stewart, and Richard JB Dobson. 2018. SemEHR: A general-purpose semantic search system to surface semantic data from clinical notes for tailored care, trial recruitment, and clinical research*. *Journal of the American Medical Informatics Association* 25, 5 (01 2018), 530–537. https://doi.org/10.1093/jamia/ocx160 arXiv:https://academic.oup.com/jamia/article-pdf/25/5/530/34150672/ocx160.pdf

[140] Y. Wu, S. Yang, and X. Yan. 2013. Ontology-based subgraph querying. In *2013 IEEE 29th International Conference on Data Engineering (ICDE)*. 697–708. https://doi.org/10.1109/ICDE.2013.6544867

[141] X. Yin, J. Han, and P. S. Yu. 2008. Truth Discovery with Multiple Conflicting Information Providers on the Web. *IEEE Transactions on Knowledge and Data Engineering* 20, 6 (2008), 796–808.

[142] Honglei Zeng, Maher Alhossaini, Li Ding, Richard Fikes, and Deborah Mcguinness. 2006. Computing trust from revision history. https://doi.org/10.1145/1501434.1501445

[143] Xia Zeng, Amani Abumansour, and Arkaitz Zubiaga. 2021. Automated fact-checking: A survey. *Language and Linguistics Compass* 15 (10 2021). https://doi.org/10.1111/lnc3.12438

[144] Xinyi Zhou, Reza Zafarani, Kai Shu, and Huan Liu. 2019. Fake News: Fundamental Theories, Detection Strategies and Challenges. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining* (Melbourne VIC, Australia) *(WSDM '19)*. Association for Computing Machinery, 836–837. https://doi.org/10.1145/3289600.3291382

[145] Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. 2018. Detection and Resolution of Rumours in Social Media: A Survey. *ACM Comput. Surv.* 51, 2, Article 32 (Feb. 2018), 36 pages. https://doi.org/10.1145/3161603