Knowledge-Based Systems 104 (2016) 123–133

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Discriminative predicate path mining for fact checking in knowledge graphs

Baoxu Shi [a], Tim Weninger [b,*]

[a] Department of Computer Science and Engineering, University of Notre Dame, 384E Nieuwland Science Hall, Notre Dame, IN 46556, USA
[b] Department of Computer Science and Engineering, University of Notre Dame, 353 Fitzpatrick Hall, Notre Dame, IN 46556, USA

ARTICLE INFO

Article history:
Received 24 February 2016
Revised 15 April 2016
Accepted 16 April 2016
Available online 19 April 2016

Keywords:
Fact checking
Path finding
Knowledge graph

ABSTRACT

Traditional fact checking by experts and analysts cannot keep pace with the volume of newly created information. It is important and necessary, therefore, to enhance our ability to computationally determine whether some statement of fact is true or false. We view this problem as a link-prediction task in a knowledge graph, and present a *discriminative path*-based method for fact checking in knowledge graphs that incorporates connectivity, type information, and predicate interactions. Given a statement $\mathcal{S}$ of the form (subject, predicate, object), for example, (Chicago, capitalOf, Illinois), our approach mines discriminative paths that alternatively define the generalized statement (U.S. city, predicate, U.S. state) and uses the mined rules to evaluate the veracity of statement $\mathcal{S}$. We evaluate our approach by examining thousands of claims related to history, geography, biology, and politics using a public, million node knowledge graph extracted from Wikipedia and PubMedDB. Not only does our approach significantly outperform related models, we also find that the discriminative predicate path model is easily interpretable and provides sensible reasons for the final determination.

1. Introduction

> *If a Lie be believ'd only for an Hour, it has done its Work, and there is no farther occasion for it. Falsehood flies, and the Truth comes limping after it.*
>
> – Jonathan Swift (1710) [1]

Misinformation in media and communication creates a situation in which opposing assertions of fact compete for attention. This problem is exacerbated in modern, digital society, where people increasingly rely on the aggregate ratings from their social circles for news and information. Although much of the information presented on the Web is a good resource, its accuracy certainly cannot be guaranteed. In order to avoid being fooled by false assertions, it is necessary to separate fact from fiction and to assess the credibility of an information source.

Knowledge graphs. We represent a *statement of fact* in the form of (subject, predicate, object) triples, where the subject and the object are entities that have some relationship between them as indicated by the predicate. For example, the *"Springfield is the capital of Illinois"* assertion is represented by the triple (Springfield, capitalOf, Illinois). A set of such triples is known as a knowledge base, but can be combined to produce a multi-graph where nodes represent the entities and directed edges represent the predicates. Different predicates can be represented by edge types, resulting in a heterogeneous information network that is often referred to as a *knowledge graph*. Given a knowledge base that is extracted from a large repository of statements, like Wikipedia or the Web at large, the resulting knowledge graph represents *some* of the factual relationships among the entities mentioned in the statements. If there existed an ultimate knowledge graph which knew everything, then fact checking would be as easy as checking for the presence of an edge in the knowledge graph. In reality, knowledge graphs have limited information and are often plagued with missing or incorrect relations making validation difficult.

Although a knowledge graph may be incomplete, we assume that most of the edges in the graph represent true statements of fact. With this assumption, existing fact checking [2] and link prediction methods [3–7] would rate a given statement to be true if it exists as an edge in the knowledge graph or if there is a short path linking its subject to its object, and false otherwise. Statistical relational learning models [8–11] can measure the truthfulness by calculating the distance between the entities and predicate in a given statement. However, the limitation of existing models make them unsuitable for fact checking. Link prediction methods, Adamic/Adar [4] and personalized PageRank [7] for example,

* Corresponding author. Tel.: +5746316770.
E-mail addresses: bshi@nd.edu (B. Shi), tweninge@nd.edu (T. Weninger).

http://dx.doi.org/10.1016/j.knosys.2016.04.015
0950-7051/

work on untyped graphs and are incapable of capturing the heterogeneity of knowledge graphs; other heterogeneous link prediction algorithms, *e.g.*, PathSim [12] and PCRW [14], not only need human annotated meta paths but also have strict constraints on the input meta paths. Statistical relational learning models such as RESCAL [8], NTN [9], TransE [10], and other variants [11,17] utilize the type information in knowledge graphs but can not work with unseen predicate types and do not model the complicated interactions among relations explicitly.

In the present work, we present a discriminative path-based method for fact checking in knowledge graphs that incorporates connectivity, type information, and predicate interactions. Given a statement $\mathcal{S}$ of the form (subject, predicate, object), for example, (Chicago, capitalOf, Illinois), our approach mines discriminative paths that alternatively define the generalized statement (U.S. city, predicate, U.S. state) and uses the mined rules to evaluate the veracity of statement $\mathcal{S}$.

Unlike existing models, the proposed method simulates how experienced human fact-checkers examine a statement: fact-checkers will first attempt to *understand* the generalized notion of the statement using prior knowledge, and then validate the specific statement by applying their knowledge. The statement usually can be generalized by replacing the specific entities by their type-labels. In the present work, we show how to understand a statement by inspecting the related discriminative paths retrieved from the knowledge graph. Returning to the "*Chicago is the capital of Illinois*" example, a fact checker, as well as our model, will learn to understand what it means for a U.S. city to be the capitalOf a U.S. state. In this trivial example, a fact checker may come to understand that a city is the capital of a state if the state agencies, governor and legislature are located in the city; from this understanding the fact checker ought to rule that Chicago is not the capital of Illinois because this statement does not satisfy the fact checker's understanding of what capitalOf means.

The advantages of this fact checking procedure is in its *generality* and *context-dependency*. Just as humans learn unknown words, model generality means the predicate of a statement can be arbitrary and is not required to be presented in the knowledge base. Moreover, once a prior knowledge is learned, it is associated with a certain type of entity pair relation and can be used for different tasks including general question answering or knowledge base completion. The notion of context-dependency allows the fact checker to discern different definitions of a predicate in different situations. For example, capitalOf could define the capitals of US states, colloquialisms such as "Kansas City is the soccer capital of America", or historical or time-sensitive predicates such as "Calcutta *was* the capital of India" depending on the context.

When performed computationally, the task of discovering interesting relationships between or among entities is known generally as association rule mining. Although there has been some effort to adapt association mining for knowledge graph completion, these methods are not well suited for fact-finding and often resort to finding global rules and synonyms [18,19] rather than generating a robust understanding of the given context dependent predicate [20].

Fig. 1 illustrates three graph fragments from the DBpedia knowledge base [21] containing cities and states. This example demonstrates, via actual results, how the proposed automatic fact checker is able to determine relationships that uniquely define what it means for an entity to be the capitalOf another entity. Association rule miners [19] and link prediction models [5,6] incorrectly indicate that the largestCity is most associated with the capitalOf predicate. In contrast, our framework, indicated by solid edges, finds the rules that most uniquely define what it means to be the capitalOf a state. In this example, our top result indicates that a US state capital is the city in which the headquarters of entities that have jurisdiction in the state are located. In other words, we find that a US state capital is indeed the city where the state agencies, like the Dept. of Transportation, or the Dept. of Health, have their headquarters.

To summarize, we show that we can leverage a collection of factual statements for automatic fact checking. Based on the principles underlying link prediction, similarity search and network closure, we computationally gauge the truthfulness of an assertion by mining connectivity patterns within a network of factual statements. Our current work focuses on determining the validity of factual assertions from simple, well-formed statements; the related problems of information extraction [22], claim identification [23], answering compound assertions [24], and others [25] are generally built in-support-of or on-top-of this central task.

Recent work in general heterogeneous information networks, of which knowledge graphs are an example, has led to the development of meta path similarity metrics that show excellent results in clustering, classification and recommendation [12,14,26,27]. The state of the art in meta path mining works by counting the path-instances or randomly walking over a constrained set of hand-annotated typed-edges [12]. Unfortunately, this means that a human has to understand the problem domain and write down relevant meta paths before analysis can begin. In this work, our focus is on methods that automatically determine the set of path-descriptions called **discriminative paths** that uniquely encapsulate the relationship between two entities in a knowledge graph.

The specific contributions of this paper are as follows:

1. We developed a fast discriminative path mining algorithm that can discover "definitions" of an RDF-style triple, *i.e.*, a statement of fact. The algorithm is able to handle large scale knowledge graphs with millions of nodes and edges.
2. We designed a human interpretable fact checking framework that utilizes discriminative paths to predict the truthfulness of a statement.
3. We modeled fact checking as a link prediction problem and validated our approach on two real world, large scale knowledge graphs, DBpedia [21] and SemMedDB [28]. The experiments showed that the proposed framework outperforms alternative approaches and has a similar execution time.

In this paper, we incorporate lessons learned from association rule mining and from heterogeneous information network analysis in order to understand the meanings of various relationships, and we use this new framework for fact-checking in knowledge graphs. To describe our approach we first formalize the problem in Section 2 and define our solution in Section 3. Section 4 presents extensive experiments on two large, real world knowledge graphs. We present related work in Section 5 before drawing conclusions and discussing future work in Section 6.

2. Problem definition

We view a knowledge graph to be a special case of a heterogeneous information network (HIN) where nodes represent entities and edges represent relationships between entities, and where heterogeneity stems from the fact that nodes and edges have clearly identified type-definitions. The type of an entity is labeled by some ontology, and the type of an edge is labeled by the predicate label. With the above assumptions, we formally define a knowledge graph as follows:

Definition 1 (Knowledge Graph)**.** A knowledge graph is a directed multigraph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R}, \mathcal{O}, \psi, \phi)$, where $\mathcal{V}$ is the set of entities, $\mathcal{E}$ is a set of labeled directed edges between two entities, $\mathcal{R}$ represents the predicate label set, and $\mathcal{O}$ is the ontology of the entities in $\mathcal{G}$. The ontology mapping function $\psi(v) = \mathbf{o}$, where $v \in \mathcal{V}$