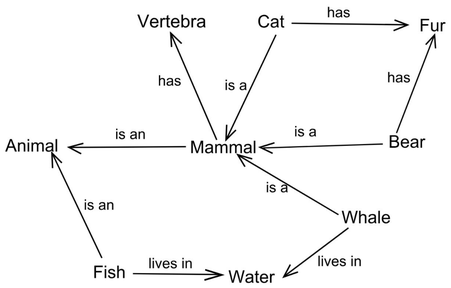
Computational Fact-Checking through Relational Similarity based Path Mining

Himanshu Ahuja and Alexander Michels

Introduction

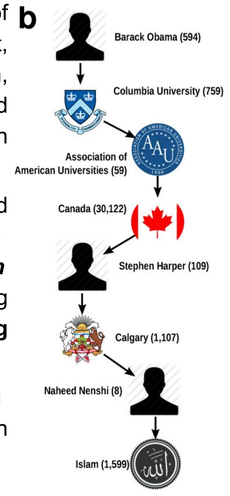
What is a Knowledge Graph?

A **knowledge graph** *G* is an ordered pair *G = (E, R)* where *E* is the set of entity or concept nodes and *R* is the set of relation or predicate edges. Within these systems, a factual statement is represented as a ***subject-predicate-object triple*** where the subject and object are concepts and the predicate describes the relationship between the two concepts.



How can one fact-check on a knowledge graph?

Fact-checking is the process of putting a claim into context, gathering relevant information, conducting thorough analysis, and reporting a conclusion with explanations and evidence.



We combine the ideas presented in *PredPath* and *Knowledge Stream* & devised ***RelPredPath*** which mines valid facts along **relatively short** paths **containing low-degree specific** nodes.

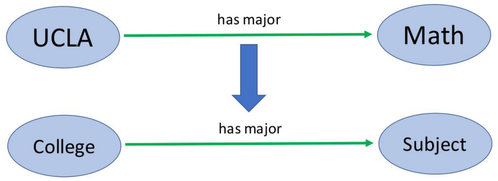
Novelty: Facts can lie along longer paths if they contain information relative to the claim.

Abstract

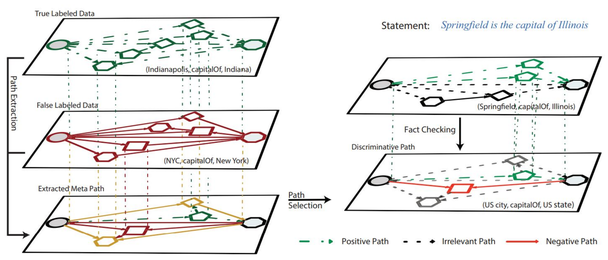
The volume of information today is outpacing the capacity of experts to fact-check it, and in the Information Age the real-world consequences of misinformation are becoming increasingly dire. Recently, computational methods for tackling this problem have been proposed with many of them revolving around knowledge graphs. We present a novel computational fact-checking algorithm, ***RelPredPath***, inspired by and improving on the techniques used in state-of-the-art fact-checking algorithms, *PredPath* and Knowledge Stream. Our solution views the problem of fact-checking as a link-prediction problem which relies on discriminative path model, but draws on relational similarity and node generality to redefine path length. This gives our solution the advantage of training on more specific paths consisting of edges whose predicates are more conceptually similar to the target predicate. *RelPredPath* shows performance at-par with other state-of-the-art fact-checking algorithms, but leads to a more robust and intuitive model for computational fact-checking. Work partially completed during the Research in Industrial Projects for Students program at UCLA's Institute for Pure and Applied Mathematics.

PredPath

The *PredPath* algorithm works by viewing fact-checking as a link-prediction task in a knowledge graph. *PredPath* works by abstracting the target fact's subject and object types and looking for paths of length *k* in a knowledge graph of with the same abstracted type endpoints. This representation captures connectivity, type information, and predicate interactions.



Then, it looks for paths where the subject and object are related by the target predicate (positive examples) and are not related by the target predicate (negative examples) within the set of discriminative paths to understand what the target subject-predicate-object relationship means.



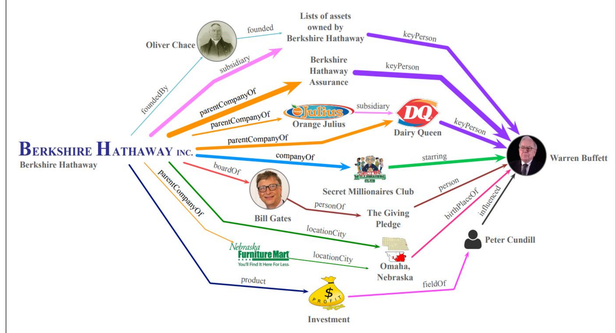
Knowledge Stream (KS)

Knowledge Stream works by viewing fact-checking as a network flow problem, "pushing" knowledge from the subject to the object.

It relies on the notion of **relational similarity** (denote *u)* which is defined as the cosine similarity between rows of the co-occurrence matrix after TF-IDF weighting is applied. The cost of each node v_j is the log of the degree of the node *log(k(v_j))* and the capacity of each edge e= (v_i, v_j) and a target fact (s,p,o) the capacity of an edge is:

$$U_{s,p,o}(e) = \frac{u(g(e),p)}{1+\log(k(v_j)}$$

Now we have transformed the fact-checking problem into a minimum cost maximum flow problem which can be solved algorithmically:



RelPredPath

Both *PredPath* and *Knowledge Stream* have interesting mathematical properties and intuitive interpretations. We utilized *PredPath*'s idea of "understanding" what the target relation meant through discriminative path mining, and combined it with *Knowledge Stream*'s intuitive use of relational similarity and node generality. We wanted to know what would happen if we employed *Knowledge Stream*'s relational similarity and node generality to substitute the path length in the *PredPath* algorithm and mine *k* weighted paths instead.

The authors of the *Knowledge Stream* paper had already used their concepts to define a new path length:

$$S'(P_{s,p,o}) = \left[\sum_{i=2}^{n-1} \frac{\log k(v_i)}{u(r_{i-1},p)} + \frac{1}{u(r_{n-1},p)}\right]^{-1}$$

RelPredPath (Relational *PredPath*) uses this definition of path length and discriminative predicate path mining to fact-check on knowledge graphs. Our use of a new definition of path length requires us to alter how we collected our set of short discriminative paths to try to keep our algorithm effective under a variety of graphs. We chose to use a k-shortest path algorithm to collect our paths (specifically Yen's k-Shortest Paths).

Feel free to view the code at: http://github.com/himahuja/StreamMiner

Discussion

We use the area under the **Receiver Operating Characteristic curve (AUROC)** as a metric to evaluate algorithms. Each method emits a list of probabilistic scores, one for each triple, and the AUROC expresses the probability that a true triple receives a higher score than a false one.

Dataset	*RelPredPath*	*PredPath*	*KS*
Presidents/First Ladies	1.0000	1.0000	0.9895
Movies/Directors	0.9741	0.9997	0.8500
Nationality	0.8400	0.9520	0.9792
Profession	0.9455	0.9271	0.9866
Place of Birth	0.6498	0.8464	0.7292
Place of Death	0.6858	0.7654	0.8002
NBA Player/Team	0.9634	0.93.31	0.9996
Civil War Battles	0.6704	0.9951	0.7780
Company/President	0.7936	0.8867	0.8119
State/Capital	1.0000	0.9968	1.0000
Vice Presidents	0.8537	0.9440	0.7780

RelPredPath performed at par with *Knowledge Stream* and *PredPath* on many datasets, but still has room for improvement through optimization and possibly a different choice of algorithm for compiling a set of shortest paths.

References

G. L. Ciampaglia, P. Shiralkar, L. M. Rocha, J. Bollen, F. Menczer, and A. Flammini, *Computational fact checking from knowledge networks*, PLOS ONE, 10 (2015).

B. Shi and T. Weninger, *Fact checking in large knowledge graphs - A discriminative predicate path mining approach,* CoRR, abs/1510.05911 (2015).

P. Shiralkar, A. Flammini, F. Menczer, and G. L. Ciampaglia, *Finding streams in knowledge graphs to support fact checking*, CoRR, abs/1708.07239 (2017).

Acknowledgements

We would like to thank the Institute for Pure and Applied Mathematics, Praedicat, Inc., and the NSF for giving us the opportunity of being a part of Research in Industrial Projects for Students (RIPS).

Special thanks are due to Stephen DeSalvo, Urjit Patel, and Susana Serna for their guidance.