COMPUTATIONAL FACT CHECKING BY MINING KNOWLEDGE GRAPHS

Prashant Shiralkar

Submitted to the faculty of the University Graduate School
in partial fulfillment of the requirements
for the degree
Doctor of Philosophy
in the School of Informatics and Computing,
Indiana University

September 2017

Accepted by the Graduate Faculty, Indiana University, in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Doctoral Committee

Filippo Menczer, Ph.D.

Alessandro Flammini, Ph.D.

Predrag Radivojac, Ph.D.

Sriraam Natarajan, Ph.D.

August 18, 2017

To my parents and family

To my uncle and aunt, Deepak Kamath and Vinita Kamath

To my wife, Rutuja

Acknowledgements

I'm thankful to a number of people who directly or indirectly have helped or supported me during my doctoral studies. In particular, I am extremely grateful to my advisor, Filippo Menczer, for his support and guidance during all phases of my PhD. Among many things, I thank him for introducing me to the problem of fact checking, for his continual advice and encouragement, and for his help in editing my papers and providing feedback on this dissertation. Without his supervision, this work wouldn't have been possible. I would also like to extend my sincere thanks to other members of my committee, Alessandro Flammini, Predrag Radivojac, and Sriraam Natarajan, for their advice and feedback on this dissertation. I am grateful to Giovanni Luca Ciampaglia for bringing me up to speed on the problem, and for all the stimulating discussions we have had over the years. Thanks are also due to Karissa McKelvey, Johan Bollen and Luis Rocha, who originally conceived the idea of measuring relatedness between concepts by performing graph traversals on a semantic network, which later grew into this thesis.

I would like to thank many members of my lab (not in any particular order) for creating a vibrant environment for the exchange of ideas and for helping me hone my communication skills as I presented my work each semester: Emilio Ferrara, Diego Fregolente, Jasleen Kaur, Lilian Weng, Onur Varol, Tak-Lon Wu, Pikmai Hui, Clayton Davis, Chengcheng Shao, Pablo Moriano, Dimitar Nikolov, Aditya Tandon, John Bollenbacher, Mihai Avram, Wen Chen, Qing Ke, Nathan Ratkiewicz, Gregory Maus, Ian Wood, and Jaehyuk Park. Besides my colleagues in the lab, I also appreciate the support of my friends: Shantanu Jain, Alisa Kaylor, Arkajyoti Sengupta, Zeeshan Sayeed, and Srikanth Iyer, who made living in Bloomington fun and exciting.

The technology and administrative staff of the School of Informatics, Computing and Engineering has been very supportive in providing me with needed facilities. I thank all members of this group, including Tara Holbrook, Regina Helton, Rob Henderson, and Shing-Shong Shei. I also thank Google and the WSDM Cup Triple Scoring challenge organizers for providing real datasets on which I could test my proposed methods.

I would like to thank my parents, Prakash and Alka Shiralkar, my brother Madhur, and sister-

in-law Sunita, for their endless support, sacrifice and love. I wouldn't have considered pursuing doctoral studies without the continual encouragement and support of my maternal uncle and aunt, Deepak and Vinita Kamath. I am also grateful to my in-laws for having faith in me and for their love and support during my doctoral studies. Finally, I thank my wife Rutuja, for her patience, love, support and encouragement during the highs and lows of my PhD studies — thanks for always being there for me.

Prashant Shiralkar

COMPUTATIONAL FACT CHECKING BY MINING KNOWLEDGE GRAPHS

Misinformation and rumors have become rampant on online social platforms with adverse consequences for the real world. Fact-checking efforts are needed to mitigate the risks associated with the massive spread of digital misinformation. However, the pace at which information is generated online limits the capacity to fact-check claims at the same rate using current journalistic practices. Computational approaches may be a key for achieving scalable fact checking. To this end, this dissertation introduces network science and machine learning methods for fact checking by leveraging high-quality information in large knowledge bases, commonly known as knowledge graphs (KGs).

We consider two variations of the fact-checking task. The first consists in assessing the truthfulness of a statement of fact as simple as a `(subject, predicate, object)` triple, where the subject entity is related to the object entity by the predicate relation. We show that a broad class of triples pertaining to generic relationships among entities in the real world, e.g., `(Indianapolis, capitalOf, Indiana)`, can be checked effectively by finding a shortest path connecting their subject and object entities in the KG under appropriately designed semantic proximity metrics. To take advantage of the broader structural and semantic context of a triple, we also extend this approach by considering multiple paths, following ideas from network flow theory. Evaluation on a range of facts related to entertainment, sports, geography and more reveals that our methods are effective in discerning true statements from false ones, often outperforming state-of-the-art algorithms. Moreover, our approaches are unsupervised, and produce meaningful explanations of predictions by automatically discovering useful patterns and contextual facts.

The second task consists in computing a relevance score that expresses the degree to which a person is associated with different professions or nationalities. For example, we wish to determine which of *Theoretical Physicist*, *Mathematician* or *Philosopher* best describes *Albert Einstein*. We introduce a supervised learning approach for assessing such relevance by extracting useful features

from the KG. Results show that our approach is effective, despite the limited information in the graph.

Filippo Menczer, Ph.D., Chairperson

Alessandro Flammini, Ph.D., Member

Predrag Radivojac, Ph.D., Member

Sriraam Natarajan, Ph.D., Member

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The Spread of Digital Misinformation and Rumors

The Internet era unlocked the potential for ordinary citizens to publish content at their own will. In the last couple decades, besides content created by journalistic organizations and mainstream media, we have seen tremendous rise in information in the form of blogs, websites, alternative media, and other content produced by citizens. Moreover, social platforms such as Facebook, Twitter, Flickr, Tumblr, and YouTube have enabled people to connect, share information and voice opinions online, thus further fuelling the production and consumption of information in novel forms. In recent times, these platforms have also become the source of news for many people. On many occasions, citizens use these social networks to report a developing incident, thus making them the first sources of news (Goode, 2009). Political figures and organizations also actively reach out to masses leveraging the connected nature of these platforms.

Unfortunately, the open and uncontrolled nature of the Web and these social networks has also led to their abuse. In recent times, we see growing incidence of disinformation and misinformation (Del Vicario et al., 2016; Hernon, 1995), rumors (Zubiaga et al., 2017), hoaxes (Kumar et al., 2016), and fake news (Allcott and Gentzkow, 2017) on these platforms. Often, topics revolve around repeated false statements made by political leaders and organizations. In addition, political astroturf (Ratkiewicz et al., 2011b), spam (Cranor and LaMacchia, 1998) and phishing attacks (Jagatic et al., 2007) have become commonplace.

The spread of misinformation and rumors has real-life consequences in society. For example, in the context of U.S. political elections, a political astroturf campaign (Ratkiewicz et al., 2011a) smeared a Democratic candidate of Delaware running for the U.S. Senate. As another example, in

2016, fake-news writers made money through Google and Facebook's self-service ad technology by daily posting fabricated and misleading news stories designed to be believable and shared (Ohlheiser, 2016). We have also seen examples of "influence bots" or fake social media accounts operated via computer scripts, whose sole purpose is to influence others' opinions around a particular topic (Ferrara et al., 2016; Subrahmanian et al., 2016). Ferrara (2015) provides an account of many such stories and their effects on the public at large. The motives behind the spread of misinformation include profit-making, altering public opinion for financial or political gains, exerting influence for a cause, defaming important public figures, and more.

One of the factors facilitating the spread of misinformation is that most social networks have low barriers to entry, and also allow a single individual to hold multiple accounts. Recent work has also shown that *homophily* is one of the main drivers behind the misinformation spread. Homophily is the tendency of like-minded people, who have similar information consumption and behavioral characteristics, to be connected in the social network. It often leads to polarized groups of users called *echo chambers* or *filter bubbles* (Bessi et al., 2016; Nikolov et al., 2015), where there is little room for appreciation of diverse or contrary perspectives.

If the spread of such misinformation and rumors is left unregulated, it could potentially wreak havoc on society and only exacerbate global risks (Forum, 2013). To address these problems, fact checking is an important activity traditionally performed by journalists and professional fact checkers to debunk false claims and thereby promote accountability and peaceful public discourse. However, the volume and velocity of claims, combined with a deluge of related information, make it difficult for fact checkers to check claims at the same pace. To deal with the problem, a number of fact-checking websites and institutes have emerged around the world. Websites such as *Snopes.com*,[1] *PolitiFact.com*[2] and *FactCheck.org*[3] in the U.S., and *FullFact*[4] in the U.K., check claims made by politicians, pundits, etc., or ones worthy of public interest, and assign them a truthfulness rating as their verdict. The precise truthfulness ratings that may be assigned differ across fact-checking sites.

[1] http://www.snopes.com/
[2] http://www.politifact.com/
[3] http://www.factcheck.org/
[4] https://fullfact.org/

For example, *PolitiFact.com* flags a claim as either True, Mostly True, Half True, Mostly False, False, or "Pants on Fire," whereas *Snopes.com* gives ratings such as True, Mostly True, Mixture, Mostly False, False, Unproven, or Legend. Nevertheless, the rise of so many fact-checking websites and organizations is a testament to the mounting global concern. The Reporters' Lab at Duke University[5] maintains a growing list of such websites.

1.2 What is Fact Checking?

Fact checking a claim involves researching its topic, putting it into broader context, gathering relevant information and data (e.g., interviews, speech transcripts, reports, spreadsheets, and statistics), conducting thorough analysis, and reporting conclusions supplemented with explanations and evidence. It is an intellectually painstaking, time-consuming, and laborious investigative activity, requiring a different set of research, analytic and writing skills than those required by traditional journalism.

Aside from the challenges inherent to the activity, human fact checkers of today often lack the right set of resources and know-how to query, analyze and visualize information from disparate sources (Borel, 2016; Cohen et al., 2011a; Hassan et al., 2015). Depending on the complexity of a claim, its investigation can take from a few minutes to days, ultimately leading to the claim being debunked, corroborated, or even remaining unresolved. Although in many situations professional fact checkers have access to an abundance of information in the form of archived stories and other data, their inability to efficiently navigate to the right information means that fact checking often lags behind the rate at which claims are made. This time-gap implies no corrective actions or checks are made early on to stem the spread of a claim. In many cases, claims go viral in cyberspace before any fact checking takes place.

[5] http://reporterslab.org/fact-checking/

1.3 The Need to Automate Fact Checking

In light of the challenges described above, it has been suggested that computation can be the key to make fact checking efficient and effective (Cohen et al., 2011a,b; Flew et al., 2012). Automating the process can not only benefit human fact checkers, but it can also assist ordinary citizens to avoid the spread of false or dubious information (Vlachos and Riedel, 2014).

Some of the current journalistic challenges as outlined by Cohen et al. (2011a) include a human fact checker's inability to (1) explore more information about known entities in articles, (2) decipher audio/video streams to efficiently navigate to relevant pieces for investigation, (3) integrate data from disparate sources, (4) find patterns from groups of documents, and (5) effectively analyze and visualize information. Thus, modern journalism needs a new generation of tools and techniques for data integration, collaborative sense-making & annotation (e.g., *hypothes.is*[6], *Check*[7]), information extraction, data mining, and visualization. Computer science can help in these respects not only through the development of appropriate tools and interfaces, but also through research and development of computational approaches for fact checking. Thus, a new discipline called *computational journalism* was born to bridge the divide. Flew et al. (2012) discuss some of the expectations and promises of computational journalism.

Initial studies of online misinformation spread revealed that, in absence of any regulation of content on social media platforms, users mostly relied on elementary cues surrounding a piece of information to judge its credibility. These clues included strange presentation formats, typos, perceived trustworthiness of source, etc. Such contextual indicators of claims have collectively come to be known as *credibility* in the literature (Castillo et al., 2011; Gupta et al., 2014; Rieh and Danielson, 2007). Subsequently, many systems have emerged to study information diffusion, and for automatic detection of rumors and misinformation in online social media (Ratkiewicz et al., 2011a; Resnick et al., 2014; Zubiaga et al., 2017).

The systems to detect and track rumors mostly rely on indicators such as number of inquiring

[6]https://web.hypothes.is/
[7]https://meedan.com/en/check/

tweets about a circulating piece of content, reporting dynamics during breaking news, temporal patterns, and source credibility. None of these systems, however, attempt to understand the content of claims in order to check them at an early stage. Thus, there is a critical need for intelligent tools and systems (d'Aquin and Motta, 2016; Davis et al., 1993) that can go beyond basic contextual credibility indicators, and assess a claim's truthfulness by reasoning about its content and relevant facts.[8]

The availability of copious amounts of digital information, and the ability of improved computing hardware to process massive quantities of data, suggest that creating an automated fact-checking system may be a key to addressing this acute need. However, the fact that most of this information is available as unstructured natural language text indicates that it is not directly suitable for understanding by current machines. The reason lies in the fundamental challenges of named entity recognition and relation extraction in natural language processing (NLP) (Cambria and White, 2014; Manning and Schütze, 1999). Nevertheless, with recent advances in NLP, information extraction (IE) and Semantic Web (Berners-Lee et al., 2001) technologies, large repositories of structured knowledge have become available in the form of knowledge graphs (KGs), which makes them amenable to computational analysis. Like any other complex network (Newman, 2003), these knowledge graphs can be analyzed using the theory and tools of network science (Newman, 2010) and machine learning (Mitchell, 1997). The KGs contain vast amounts of high-quality knowledge about real-world entities and their relationships, and thus could be, in principle, used toward creating automated fact-checking systems. In the past, these KGs have also powered question-answering (QA) systems such as IBM Watson (Ferrucci, 2012) and PowerAqua (Lopez et al., 2012).

Nodes in a KG typically represent real-world entities such as a person, place, organization, or event; and edges, also known as semantic predicates or relations, correspond to the relationships between entities. A statement of fact about a real-world relationship is represented in the KG by a (s, p, o) triple, where a *subject* entity s is related to an *object* entity o by a *predicate* relation p. An example of triple representing a fact is `(France, capital, Paris)`, which indicates

[8]Here, we mean *reasoning* in a broad sense, unlike *reasoning* in the Artificial Intelligence community, which is a sub-field that deals with proving theorems.

that Paris is the capital of France. In this work, we often use the terms *fact*, *triple*, and *edge* interchangeably. Many KGs exist today with a wide variety of facts covering diverse topical domains. KGs also differ in their size and quality, a topic we briefly discuss later. A few examples of publicly available ones are DBpedia (Bizer et al., 2009), YAGO2 (Hoffart et al., 2013) and Wikidata (Erxleben et al., 2014).

1.4 Contributions of this Dissertation

This work takes steps toward the creation of a computational fact checker by introducing network science and machine learning approaches for assessing veracity of simple claims based on the background knowledge in a KG. Although KGs often contain facts that may not be true in reality, in our research, we assume that all facts in a given KG are true.

The key research question we address is:

> Given a large-scale knowledge graph of facts, how can we assess the veracity of a claim, simple enough to be represented by a triple?

Framed in this manner, fact checking a claim triple (henceforth, just claim) is equivalent to checking whether an edge exists in the KG. If it does, trivially, the claim can be considered to be true. The case becomes challenging when such an edge does not exist, and hence its likelihood of existence needs to be predicted using other facts in the KG. This problem is commonly known as the *link prediction* problem (Liben-Nowell and Kleinberg, 2007; Lü and Zhou, 2011) for knowledge graphs.

The central thesis of this research is that the set of short paths connecting a claim's subject and object entities holds explanatory power to assign it a *truth value*, signifying its degree of truthfulness.

We consider two variations of the fact-checking task. They consist in assessing —

1. a broad class of triples that represent general facts about entities. An example of a triple in this category is `(Berkshire Hathaway, keyPerson, Warren Buffet)`, and
2. a special class of triples that specify the type of person entities, e.g., `(Albert Einstein,`

`profession, Theoretical Physicist)`, where the type *Theoretical Physicist* comes from a pre-defined set of types denoting a person's profession.

To fact-check triples in the first class, we introduce three network-theoretic approaches. The first two approaches, namely Knowledge Linker (KL) and Relational Knowledge Linker (KL-REL), assign a truth value by identifying the shortest path between its node pair under appropriately defined semantic proximity metrics. The third approach, called Knowledge Stream (KS), generalizes the strategy followed by KL and KL-REL, and employs ideas from network flow theory to assess veracity. It uses multiple short paths connecting the node pair, and assigns a truth value based on their capacity to carry flow.

Evaluation of these approaches on a range of hand-crafted and real-world datasets shows that they are effective at discerning true claims from false ones. Moreover, their performance is comparable to the state of the art in computational fact checking. Unlike prior methods, all three approaches operate in an unsupervised manner, and also offer rich interpretability by automatically surfacing relevant facts and patterns for a claim.

In pursuing the three approaches, we make two novel contributions: (1) we provide a new definition of path length based on a concept called *specificity of a node*, and (2) we introduce an approach for measuring semantic similarity between a pair of predicate relations.

The second class of triples demands predicting a relevance score that expresses the degree to which humans associate a person (e.g., Albert Einstein) to a particular type (e.g., Theoretical Physicist, Mathematician or Philosopher) under a specific relation such as *profession*. In the last part of this thesis, we introduce a novel supervised learning approach called RelSifter for assessing such relevance by extracting useful features from the KG. This relevance assessment can be seen as a special case of fact checking. We find that our approach is effective for the problem, and its performance lags behind the state of the art only by a small margin.

The following publications were produced while pursuing this work.

— Giovanni Luca Ciampaglia, Prashant Shiralkar, Luis M. Rocha, Johan Bollen, Filippo Menczer, Alessandro Flammini. "Computational Fact Checking from Knowledge Networks." *PLoS*

ONE, 10(6):e0128193, 2015.

— Prashant Shiralkar, Mihai Avram, Giovanni Luca Ciampaglia, Filippo Menczer, Alessandro Flammini. "RelSifter: Scoring Triples from Type-like Relations." To appear in *Proceedings of the Web Search and Data Mining (WSDM) Cup 2017*.

— Prashant Shiralkar, Giovanni Luca Ciampaglia, Alessandro Flammini, Filippo Menczer. "Finding Streams in Knowledge Graphs to Support Fact Checking." Accepted for *Proceedings of the International Conference on Data Mining 2017*.

Besides work on fact checking, I also contributed to detection of social bots (Subrahmanian et al., 2016), which is related to the spread of misinformation online.

1.5 Outline

We first review in Chapter 2 a few important graph theory and network flow concepts essential to understanding this work. In Chapter 3, we look at some prior work relevant to fact checking to understand the problem and put our contributions into context. The next three chapters contain the main contributions of this thesis: In Chapter 4, we present KL and KL-REL, the two shortest path approaches for fact checking. We introduce KS in Chapter 5, and also make a comparative evaluation with prior algorithms designed for fact checking, knowledge graph completion and link prediction. In Chapter 6, we look at the problem of assigning relevance scores to triples, introducing RelSifter as a solution. Finally, we summarize our contributions in Chapter 7, discussing its strengths and limitations, which also speak to some extensions. We also discuss research problems to explore to advance the state of the art in fact checking.

Chapter 2

Background

In the last chapter, we briefly introduced knowledge graphs (KGs) and a few of the intelligent applications that they have enabled. In this chapter, we discuss them in a bit more detail, looking at an example and a couple of KGs we use in this work. We then review a few essential concepts and algorithms related to graph theory and network flow theory to facilitate understanding of ideas in later chapters.

2.1 Knowledge Graphs

The vision of Semantic Web as outlined by Berners-Lee et al. (2001) is to enable intelligent reasoning based on information available on the Web. Knowledge graphs are an example of structured data, created to enable intelligent processing. As previously mentioned, KGs contain a wide variety of facts in the form of (s, p, o) triples. All three pieces of a triple, namely the subject s, predicate p, and object o are identified by a Uniform Resource Identifier (URI), and a set of triples or graph is available in a serialization format called *Resource Description Framework* (RDF) (Klyne and Carroll, 2004). The ability to refer to any entity over the Web using its URI allows us to specify relationships between entities from distinct KGs created from heterogeneous sources, thereby creating a seamless, global data space called *Linked Data* (Bizer, 2009). Such a KG or a collection can be used toward building any intelligent application, including fact checking.

A KG is an example of Linked Data with one differentiating factor: it is meant to preserve facts that are accurate, which may not be the case with Linked Data, since the latter aims to model any kind of digital content in a structured form, and thus often contains incorrect and contradictory information due to its organic nature. As an example, the birth date of a notable person may be

listed differently on Wikipedia than that on his own blog. This is allowed in Linked Data, but an ideal knowledge graph would be free from such conflicts; however, in practice, this is unfortunately not true even for KGs.

Different kinds of knowledge can be expressed in a KG using different language tools provided by the Semantic Web community. All facts are represented in the same form however, i.e., the RDF triples. Relationships among real-world entities (or *instances*), are known as *instance data* or *ground facts*. An example of instance data is the triple `(Barack Obama, spouse, Michelle Obama)` which indicates that Barack Obama is the spouse of Michelle Obama.

One can also add background information about the entities in this triple. For example, one could specify by the `(Barack Obama, type, Politician)` triple the fact that Barack Obama is an instance of a class called Politician. Classes such as Politician generally come from a pre-defined hierarchy of type relationships called an *ontology* or a *schema*, and one could also state meta-facts, or facts about things in the ontology, e.g., `(Politician, type, Person)`. An ontology may also contain a hierarchy of relation types. A lot can be expressed about an application domain using constructs in RDF and RDF Schema (RDFS) (Brickley et al., 2014), a language to express ontological facts. If greater expressiveness is desired to express knowledge that cannot be represented using RDF or RDFS, one could use the Web Ontology Language (OWL) (McGuinness et al., 2004), which is provided in a few different flavors called *profiles*. Each OWL profile is a subset of first-order logic and has unique constructs to express different knowledge. For example, restrictions can be placed on the kind of instance triples allowed in a particular class. One could also create a new class based on existing classes using set difference, or by enumeration of specific instance triples. The choice of language depends on an application's modeling requirements. As an example, for fact checking, on recommendation of human fact checkers, any of these language tools could be used to represent certain facts about the world at a desired level of detail.

2.1.1 An Example

Let us build a toy KG to illustrate how facts are typically represented based on natural language sentences. Along the way, we will see the kinds of facts one can express using the languages discussed above. Consider the following information about former U.S. President Barack Obama:

> "Barack Obama was born on April 8, 1961 in Honolulu which is the capital of Hawaii."

Based on this information, we can express the following few facts as triples in Turtle format:[1]

```
ex:Barack_Obama ex:birthDate "04-08-1961" .
ex:Barack_Obama ex:bornIn ex:Honolulu,_Hawaii .
ex:Honolulu,_Hawaii ex:capitalOf ex:Hawaii .
```

Here, the prefix `ex:` represents a namespace corresponding to our example domain, and stands as a placeholder for the domain-level URI. All entities and relations are said to belong to this namespace. The specified triples constitute the instance data.

We can also express additional "background" knowledge using a pre-defined formal ontology:

```
ex:Barack_Obama rdf:type ex:Person .
ex:Honolulu,_Hawaii rdf:type ex:City .
ex:Hawaii ex:type ex:State .
ex:capitalOf rdf:type rdf:Property .
ex:capitalOf rdfs:subPropertyOf ex:locatedIn .
ex:capitalOf rdfs:domain ex:City .
ex:capitalOf rdfs:range ex:State .
```

Here, *ex:Person*, *ex:City* and *ex:State* are pre-defined classes in the ontology. Notice how the constructs of RDF and RDFS (e.g., *rdf:type*, *rdfs:subPropertyOf*, *rdfs:domain*) allow us to say more about these entities.

[1] A succinct alternative to RDF. More details at https://www.w3.org/TeamSubmission/turtle/

What does it mean for a relation to be a sub-property of another, e.g., `ex:capitalOf rdfs:subPropertyOf ex:locatedIn .`? The exact semantics associated with an RDF/RDFS relation are provided by the RDFS standard in the form of *inference rules*. Similar inference rules can also be specified by the knowledge representer (commonly referred to as *knowledge modeler* in the Semantic Web terminology) for relations not part of the standards. Additionally, as previously mentioned, knowledge relating to restrictions on instance memberships for classes, or constraints involving sets of classes can be specified using OWL. A gentle introduction to such capabilities and more can be found in a book by Allemang and Hendler (2011). Figure 2.1 shows a graph representation of the triples above.

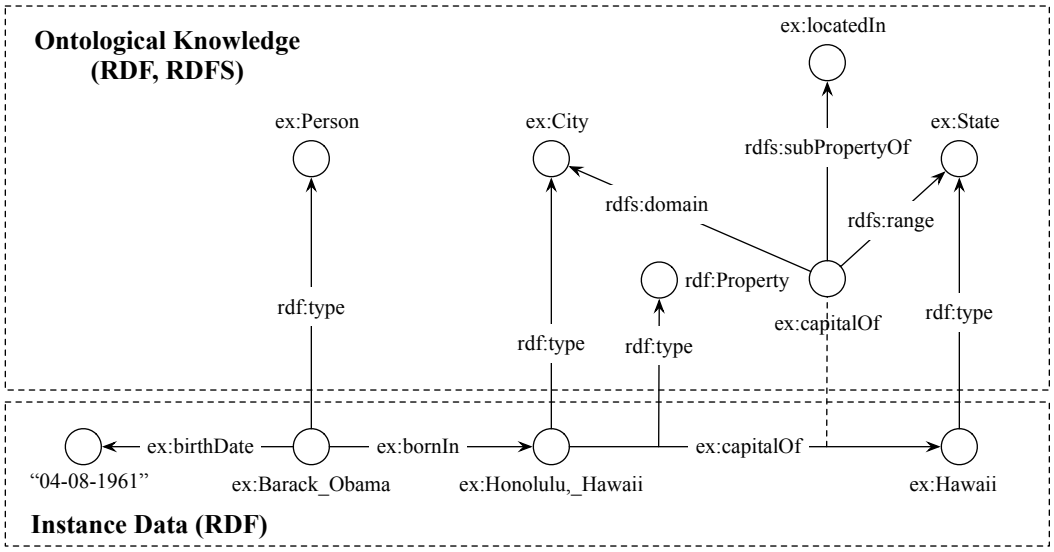


Figure 2.1: An example of Knowledge Graph.

2.1.2 Large-Scale Knowledge Graphs

We overview two KGs used in this work, namely DBpedia (Bizer et al., 2009) and Wikidata (Erxleben et al., 2014). Comparisons of these and many other publicly available knowledge graphs are provided by Färber et al. (2015) and Paulheim (2017).

DBpedia

DBpedia[2] is a multi-lingual KG resulting from a large-scale community effort to extract structured facts from semi-structured and unstructured information in Wikipedia. Semi-structured information may include facts from infoboxes, and unstructured information, in general, includes any free-form text in an article. DBpedia has its own ontology maintained by the community. Utmost efforts are made to represent facts using the concepts in this ontology. For example, a community of volunteers maintains a set of mappings that allow Wikipedia editors to use terms from the ontology, and also help reconcile equivalent information specified in diverse ways. Many datasets are available[3] as part of the DBpedia distribution, each representing specific subsets of data such as infobox information, images, URL links, geo-coordinates, redirect links, and many more. In our work we use the ontology[4], instance-types and mapping-based properties of the English DBpedia version. By pooling together this information, we form a subset version of the KG, consisting of 6M nodes, 663 relations, and over 24M triples.

Wikidata

Wikidata is another example of a growing large-scale KG created using information in Wikipedia and many other sources. One of its distinctive features is that it allows anyone to directly edit its information through a web user interface. Compared to DBpedia, it follows a richer data model. Besides basic information expressed as triples, known as *simple statements*, the Wikidata design also allows one to represent facts about facts, e.g., contextual and provenance information. Like DBpedia, Wikidata has its own taxonomy, a class hierarchy of relations, and class membership information. In our work we use this information from the RDF dump of August 2016.[5] The resulting version of our graph consists of 29.4M nodes, 839 relations and over 234M triples.

[2] http://wiki.dbpedia.org/
[3] http://wiki.dbpedia.org/downloads-2016-04
[4] http://mappings.dbpedia.org/server/ontology/classes/
[5] tools.wmflabs.org/wikidata-exports/rdf/exports/20160801/dump_download.html

2.2 Representations of Knowledge Graphs

Graph Representation

Mathematically, a KG is a directed graph $G = (V, E, \mathcal{R}, g)$, where V is the set of $|V|$ nodes corresponding to entities in the graph, E is the set of $|E|$ edges connecting these nodes, $\mathcal{R}$ is the set of R distinct relations, and $g : E \rightarrow \mathcal{R}$ is a mapping that labels each edge with a semantic relation or predicate. Each edge in G corresponds to a (s, p, o) triple in the RDF formalism, and represents a simple statement of fact about the real world. While a KG in its original form represents a directed graph, in most of our work, we represent G as an undirected graph by replacing each directed edge by two opposing edges. Our experience suggests that doing so improves reachability between node pairs. At the same time, we also risk destroying inherent information in the KG by enforcing connections that may not exist in reality or may not make sense. We leave the assessment of effects due to the use of undirected graph vs. directed graph as future work, and use the undirected version in this dissertation.

Line Graphs

The line graph $L(G) = (V', E')$ of an undirected graph $G = (V, E)$ is a graph whose nodes set is $V' = E$, and two nodes in $L(G)$ are adjacent *iff* the corresponding edges of G are incident on the same node in G, i.e, $E' = \{(e_1, e_2 : e_1, e_2 \in E \wedge e_1 \cap e_2 \neq \emptyset)\}$. In other words, two nodes in $L(G)$ are connected if they are *co-incident* on the same node in G. See Figure 2.2 for an example. Note that $L(G)$ contains adjacency of relations (e.g., A, B, C and D), and there can be multiple nodes for the same relation in G (e.g., B and C). Such a graph is often useful in understanding co-occurrence patterns of relations.

Matrix Representation

A knowledge graph G can be alternatively seen as a collection of R *adjacency* matrices, each of size $|V| \times |V|$ corresponding to a distinct relation in $\mathcal{R}$. The rows and columns in these matrices

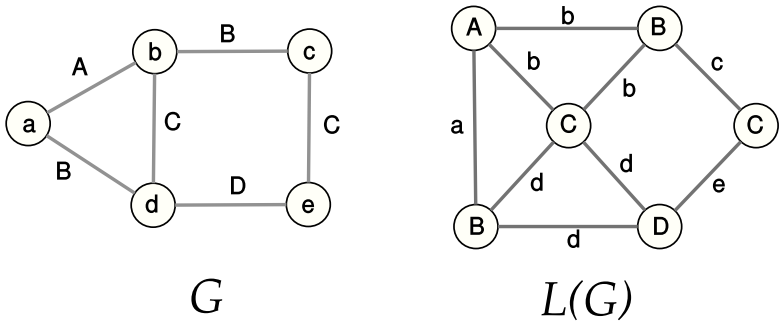
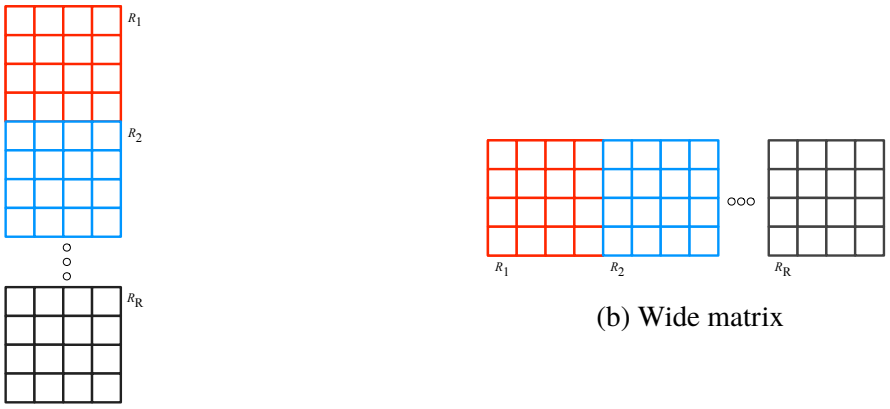


Figure 2.2: An example of knowledge graph G and its line graph $L(G)$.

represent the entity set V in G, and an entry of 1 in the i'th row and j'th column of a specific matrix indicates the existence of a relationship between i'th and j'th entities. Some methods applicable for fact checking (Chapter 3) model KG as a single matrix instead of the collection. They do so by transforming the collection into one giant *tall* (or *wide*) matrix by stacking the matrices vertically (or horizontally) as shown in Figure 2.3.



(a) Tall matrix

(b) Wide matrix

Figure 2.3: A giant tall and wide matrix representation of a KG.

Tensor Representation

As a collection of R matrices, a KG can be represented as a third-order *adjacency tensor* (a cube) in which the first dimension models the subject entities, the second dimension models the object entities, and the third dimension models the relations. See Figure 2.4 for an example. Typically, entities along the first and second dimensions are same for multi-relational data in KGs, and each

frontal slice (a matrix) of the tensor corresponds to a distinct relation. An entry of 1 in this tensor indicates the existence of a triple involving entities in i'th row and j'th column under the k'th relation.

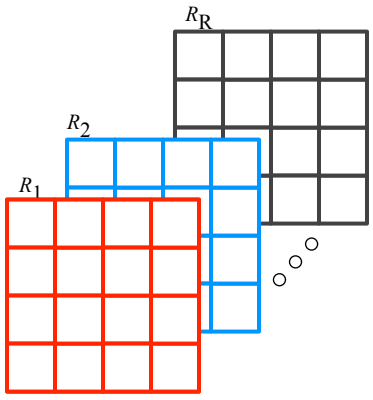


Figure 2.4: An tensor representation of a KG.

2.3 Graph Theory Concepts

Degree of a Node

The degree $k(v_i)$ of a node $v_i \in V$ is given by the number of edges incident to it. If G is directed, one may want to distinguish the incoming edges from the outgoing ones, and accordingly use an *in-degree* $k^{\text{in}}(v_i)$ for incoming edges, and *out-degree* $k^{\text{out}}(v_i)$ for outgoing ones. The degree of a node helps understand many functional properties of the node.

Weight of an Edge

An edge $e \in E$ can be associated with a number w_e in an appropriate interval, to mean a variety of things, depending on the context, e.g., it could reflect the confidence in a fact encoded by the edge. This number implicitly represents the *weight* of the edge under a specific interpretation. If the weight indicates strength of relationship between its entity pair, then the higher the weight, the stronger is their relationship, and vice versa. In this case, G is known as a *proximity network*; on

the other hand, if the weight indicates distance, the higher the distance, the more dissimilar are the entities. In this case, G represents a *distance network*.

Path

A path $P_{s,o}$ of length n between a pair of distinct nodes s and o in G is a sequence of n nodes connected by $n-1$ relations:

$$v_1 \xrightarrow{\mathrm{r}_1} v_2 \xrightarrow{\mathrm{r}_2} \cdots \xrightarrow{\mathrm{r}_{n-1}} v_n$$

A direct edge between s and o represents the simplest path. Each edge on a path is associated with a relation $r \in \mathcal{R}$. An alternative definition of path called the *relational path* can be given in terms of a sequence of relations. When the context is clear, we just refer to it as a path.

Distance-to-Proximity Function

A distance-to-proximity function ϕ is a mapping $\phi : [0, \infty] \rightarrow [0, 1]$ that converts edge weights $d_{ij} \in [0, \infty]$ in a distance network to edge weights $p_{ij} \in [0, 1]$ in the proximity network, and satisfies the following properties:

1. ϕ is strictly monotonically decreasing, $\forall a, b \in [0, \infty] : a > b \rightarrow \phi(a) < \phi(b)$;
2. $\phi(0) = 1$ and $\phi(1) = 0$.

In general, this mapping can take many forms. One popular choice that we use in this dissertation is the following:

$$\phi(d) = \frac{1}{1+d}, \tag{2.1}$$

where $d \in [0, \infty]$ represents the distance between two nodes in G, and $\phi(d)$ their proximity or similarity. This function ensures that the transitive closure of proximity network is isomorphic to that of the distance network. For more details, refer the reader to (Simas and Rocha, 2015).

2.4 Network Flow Theory

Network flow theory (Ahuja et al., 1993) enables modeling of many situations in transportation networks, manufacturing, engineering and other applications. We briefly overview the general terminology used in the literature, and a few important problems relevant to our work.

2.4.1 Concepts

Flow Network

A flow network represents an alternative view of a graph G, in which an abstract entity is assumed to flow over the network. An example is a network of pipes. Typically, some nodes in such a network act as *source* nodes and *sink* nodes. The source nodes are assumed to have a certain amount of excess of the abstract entity, whereas the sink nodes have a certain demand for the entity. Each edge e in the network has a *capacity* $\mathcal{U}(e)$ to carry flow. The entity emanates from the source nodes and flow over their outgoing edges, and is eventually transmitted to the sink nodes through the edges of the network. In our work, we have a single source and sink node. The abstract entity depends on the application being modeled. For example, for transportation networks, it may represent the traffic between two intersections; or in a logistics planning network, it may represent goods being transported from manufacturing plants to warehouses.

Flow

A flow f between nodes s and o is a mapping $f : E \to \mathbb{R}^+$ that assigns a non-negative real value $f(e)$ to each edge e in E, quantifying the amount of flow carried by e. In general, the flow f must satisfy two kinds of constraints: (1) for every edge e, the flow is bounded, i.e.

$$0 \leq f(e) \leq \mathcal{U}(e) \text{ (edge capacity constraints)}, \tag{2.2}$$

and (2) for every node v_i other than s and o, the amount of flow entering the node is same as that leaving the node, i.e.

$$\sum_{v_j \in V} f(v_j, v_i) = \sum_{v_j \in V} f(v_i, v_j) \text{ (node conservation constraints).} \tag{2.3}$$

The value of flow γ between s and o is upper bounded by the total capacity of outgoing edges at s, i.e., $\gamma \leq \sum_{v_j \in V} f(s, v_j)$.

Bottleneck of a Path

The maximum flow on a path $P_{s,o}$ connecting s and o is the minimum of capacities of its edges. The edge on this path with the minimum capacity is known as its *bottleneck*, and its flow value is denoted by $\beta(P_{s,o})$.

Residual Graph

A residual graph $G(f)$ of G is a graph with same set of nodes V as G, but with two kinds of edges: (1) *forward edges* which already carry some flow, but have some "leftover capacity" over which one can push additional flow, and (2) *backward edges* over which one can push flow in order to undo flow in their original (forward) edges. A residual graph indicates the state of the graph with respect to current flow f. The capacity of an edge in the residual graph is called the *residual capacity*.

Stream

We call a set of paths carrying flow between a pair of nodes s and o a stream $\mathcal{P}$.

Cost of an Edge

Earlier, we discussed the notion of edge weight. One such kind of weight that may be associated with an edge e in E is a cost $c(e) \in \mathbb{R}$ to indicate the cost incurred per unit flow over the edge. Such

cost may be modeled as per application requirements. For example, in a transportation network, it may mean the distance between two cities.

Demand and Supply of a Node

One may also associate with each node $v \in V$ a quantity $b(v) \in \mathbb{R}$ that indicates its supply or demand of the abstract entity flowing over the network. If $b(v) > 0$, it indicates a supply and the node is called a source node; if $b(v) < 0$ it indicates a demand and is called a sink node. See Section 2.4.1. Like cost of an edge, supply or demand depends on the modeling requirements. For example, in a logistics planning network, nodes representing manufacturing units may have supply of a commodity, while other nodes may have an equivalent demand.

We now review a few network flow problems and algorithms to solve them.

2.4.2 Shortest Path Problem

The shortest path problem seeks to find a shortest-length directed path from s to o in G. When G is interpreted as a flow network having unit capacity edges, it can be stated as the problem of sending one unit of flow as cheaply as possible from s to o. Typically, the length of an edge is taken as the cost of an edge. Mathematically, the problem is formulated as

$$\text{Minimize} \sum_{(v_i,v_j)\in E} c(v_i, v_j) \cdot f(v_i, v_j)$$

subject to

$$\sum_{v_j \in V} f(v_i, v_j) - \sum_{v_j \in V} f(v_j, v_i) = \begin{cases} 1 & \text{for } v_i = s \\ -1 & \text{for } v_i = o \\ 0 & \text{for } v_i \in V - \{s, o\} \end{cases} \tag{2.4}$$

and

$$f(v_i, v_j) \geq 0 \tag{2.5}$$

where the summations in Eqn. 2.4 represent the total flow exiting and entering a node v_i.

The most common way of solving the shortest path problem is by using Dijkstra (1959)'s algorithm. In our work, we use its variant with a binary heap implementation, which has the worst-case running time complexity of $O(|E| \log_2 |V|)$ and performs fairly well on large-scale KGs.

2.4.3 Maximum Flow Problem

The maximum flow problem seeks to find the maximum flow that can be pushed from s to o, while satisfying edge capacity and node conservation constraints. Mathematically, this problem is stated as follows.

$$\text{Maximize } \gamma$$

subject to

$$\sum_{v_j \in V} f(v_i, v_j) - \sum_{v_j \in V} f(v_j, v_i) = \begin{cases} \gamma & \text{for } v_i = s \\ -\gamma & \text{for } v_i = o \\ 0 & \text{for } v_i \in V - \{s, o\} \end{cases} \tag{2.6}$$

and

$$0 \leq f(v_i, v_j) \leq \mathcal{U}(v_i, v_j). \tag{2.7}$$

A solution γ^* is called the *feasible* flow.

A number of algorithms exist to solve the problem, differing in their theoretical and run-time behavior. They can be said to follow either of the following two strategies: (1) incrementally push flow on paths with available capacity, or (2) flood all of the outgoing edges of s by pushing flow simultaneously, and further transmitting it downstream until it reaches o. Excess flow that cannot reach o is returned back to s. The most popular algorithm following the first strategy is called the *augmenting path algorithm* by Ford and Fulkerson (1956). Generally, algorithms following the second strategy are more efficient. An example in this category is the *highest-label preflow-push algorithm* by Goldberg and Tarjan (1988), with a worst-case complexity of $O(|V|^2\sqrt{|E|})$.

2.4.4 Minimum Cost Flow Problem

The minimum cost flow problem seeks to find a least-cost way to move the supply at supply nodes (with $b(v) > 0$) to demand nodes (with $b(v) < 0$). Mathematically, it is stated as follows:

$$\text{Minimize} \sum_{(v_i,v_j)\in E} c(v_i, v_j) \cdot f(v_i, v_j)$$

subject to

$$\sum_{v_j \in V} f(v_i, v_j) - \sum_{v_j \in V} f(v_j, v_i) = b(v_i) \text{ for all } v_i \in V \tag{2.8}$$

and

$$0 \leq f(e) \leq \mathcal{U}(v_i, v_j). \tag{2.9}$$

There are many algorithms in literature to solve this problem: Successive Shortest Path (SSP), Cycle Cancelling, Relaxation, Network Simplex, and many others. They differ in their worst-case performance based on the strategies they employ. Some are pseudo-polynomial, others are weakly and strongly polynomial. Here, we describe at a high level the Successive Shortest Path (SSP) algorithm, since we use it in this dissertation. For a more detailed introduction of SSP and other algorithms, we refer the reader to the introductory text on network flows by Ahuja et al. (1993).

Successive Shortest Path relies on the concept of *pseudoflow* which is a mapping from E to $\mathbb{R}^+$ which satisfies edge capacity constraints but need not satisfy node conservation constraints. The idea is to start with a pseudoflow, whereby some nodes have supply and others have an equivalent demand, and iteratively send flow along a shortest path in the residual network from a node with available supply to meet the demand at other nodes. The algorithm eventually aims to convert the pseudoflow into a flow, where the node conservation constraints are satisfied. The resulting flow is the optimal feasible flow.

2.4.5 Minimum Cost Maximum Flow Problem

The minimum cost, maximum flow problem is a special case of the minimum cost flow problem, in which we have only two nodes, a source node s with a non-zero supply, and a target node o with an

equivalent demand, and we are interested in transferring the maximum possible flow from s to o, while minimizing the total cost. Mathematically, the problem is stated as follows:

$$\text{Maximize } \gamma \text{ while minimizing } \sum_{(v_i,v_j)\in E} c(v_i,v_j)\cdot f(v_i,v_j)$$

subject to

$$\sum_{v_j\in V} f(v_i,v_j) - \sum_{v_j\in V} f(v_j,v_i) = \begin{cases} \gamma & \text{for } v_i = s \\ -\gamma & \text{for } v_i = o \\ 0 & \text{for } v_i \in V - \{s,o\} \end{cases} \tag{2.10}$$

and

$$0 \leq f(v_i,v_j) \leq \mathcal{U}(v_i,v_j). \tag{2.11}$$

Being a special case of the minimum cost flow problem, algorithms designed for the latter are applicable to this problem. For example, Successive Shortest Path can solve this problem by iteratively sending flow along paths from the source to the sink node, as long as a path exists with a non-zero capacity.

2.5 Evaluation

2.5.1 Assumptions about Unobserved Triples

Observed triples in a knowledge graph are considered to be true. However, for fact checking, one often makes assumptions about the truthfulness of unobserved triples. We discuss below three such assumptions commonly made in practice:

- **Closed World Assumption (CWA)**: Unobserved triples are interpreted to be false under this assumption. For example, database systems have always operated under this assumption.

- **Open World Assumption (OWA)**: Under this assumption, the veracity of unobserved triples is assumed to be unknown. This assumption is justified while working with KGs extracted from open domains such as the Web, because KGs are known to be incomplete, and hence an unobserved triple may not necessarily represent a false fact.

- **Local-Closed World Assumption (LCWA)**: Under this assumption, for an observed triple, all other triples with the same subject-predicate pair are considered to be false. For example, for the triple `(Barack Obama, bornIn, Honolulu, Hawaii)`, triples such as `(Barack Obama, bornIn, Chicago, Illinois)` are taken as false facts. This assumption is commonly made in practice to create an artificial sample of false triples, required for training fact-checking models. It generally holds for functional relations such as *bornIn*, but may not hold for other relations, such as set-valued relations like *studiedAt*. Dong et al. (2014) empirically show that this serves well in practice.

2.5.2 Metrics

In this section, we review a few metrics that are used in subsequent chapters.

Confusion matrix

Given a classifier that outputs a score in the interval [0, 1] for a binary classification problem, one generally uses a threshold to obtain binary predictions, i.e., predicted true and false labels. Based on the ground truth and classifier's predictions, there are four possible outcomes, which can be conveniently represented in a two-by-two table called the *confusion matrix* (see Figure 2.5). Here, the ground truth labels (P and N) and predicted labels (P' and N') are shown along the columns and rows respectively. True facts are considered positive examples, whereas false ones as negatives. Each cell in this matrix contains the number of instances corresponding to a combination of these labels, namely True Positives (TP), False Positives (FP), False Negatives (FN) and True Negatives (TN). The numbers along the diagonal represent the correct decisions, while the off-diagonal numbers represent the "confusion" on part of the classifier. The total number of actual positives and negatives in the test sample can be determined by the total along the columns, i.e., total positives = $TP + FN$ and total negatives = $FP + TN$.

Based on a confusion matrix, one can derive a few basic useful quantities:

- **True Positive Rate**: The True Positive Rate (TPR) of a classifier is the fraction of total

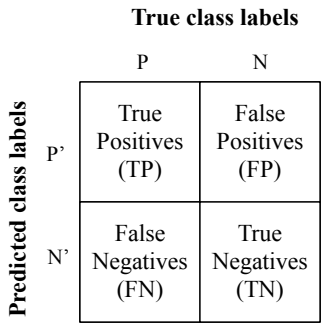


Figure 2.5: Confusion matrix.

positives that are correctly predicted by the classifier (given a specific threshold on its score):

$$\text{True Positive Rate} = \frac{TP}{TP + FN}.$$

The True Positive Rate is also called *recall* in the information retrieval literature, since it indicates the number of relevant items (positive examples are considered to be entities of interest) correctly surfaced by the classifier.

- **False Positive Rate (FPR)**: The False Positive Rate (FPR) of a classifier is the fraction of total negatives that are incorrectly flagged as positive by the classifier (given a specific threshold on its score):

$$\text{False Positive Rate} = \frac{FP}{FP + TN}.$$

- **Precision**: The precision of a classifier is the fraction of positive examples that are correctly predicted:

$$\text{Precision} = \frac{TP}{TP + FP}.$$

The term "precision" also comes from the information retrieval literature where precision measures the ability of a classifier to identify the correct relevant entities among the set of

items that the classifier marks to be relevant.

- **F-score**: F-score (or F_1-score) is a single number summary that combines precision and recall of a classifier. It is obtained by taking the harmonic mean of precision and recall:

$$\text{F-score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Area under the Receiver Operating Characteristic curve (AUROC)

One plots the true positive rate on the Y-axis and false positive rate on the X-axis to obtain what is known as the *Receiver Operating Characteristic (ROC)* curve. Each point on this curve represents a distinct classifier obtained by using a specific threshold on its scores, and multiple such points are obtained by varying the threshold from $-\infty$ to $+\infty$. In practice, an ROC curve is produced more efficiently using alternative methods, without having to generate points for all thresholds in this range.

An ROC curve shows the trade-off between the true positives and false positives of a classifier, thus allowing a visual comparison between different classifiers in a single plot. The top-left corner of this plot represents a perfect classifier, i.e., a classifier that is 100% accurate. All points falling on the diagonal line are considered to be random classifiers, because they result in equal fractions of correct and incorrect predictions. Classifiers in the bottom-left region are generally considered to be conservative because they aim to minimize false positives. The top-right region on the other hand represents liberal classifiers because such classifiers predict most examples as positive, at the cost of accepting many false positives. Classifiers in the top-left region are considered to be promising ones.

Although an ROC curve is visually illuminating for choosing optimal classifiers, in practice, a single number summary based on the curve is generally preferred. This number is called the *Area under ROC curve* (AUROC) and represents the fraction of the unit square that falls under the ROC curve. The AUROC also represents the probability of the classifier to rank a randomly

chosen positive example ahead of a randomly chosen negative example. Moreover, since changes in class distribution do not affect the ROC curve and hence AUROC, AUROC is a reliable metric for classifier evaluation. We use AUROC extensively to evaluate fact-checking algorithms. For more details about ROC curves and AUROC, we refer the reader to an introductory paper by Fawcett (2006).

Area under the Precision-Recall curve (AUPR)

Another way to evaluate the performance of a classifier is to understand the trade-off between its precision and recall. A plot with precision along Y-axis and recall (or True Positive Rate) along X-axis is called the *Precision-Recall* (PR) curve. Like ROC curve, a point on a Precision-Recall (PR) curve corresponds to a distinct classifier using a specific decision threshold. Unlike ROC curve however, in this curve, the top-right corner represents a perfect classifier, and classifiers in the top-right region are considered to be promising. As in ROC space, the fraction of unit square under the PR curve is called the *Area under Precision-Recall curve* (AUPR), and serves as a useful single number metric.

Accuracy by Maximum Deviation

Given an integer sequence s representing a true ordering of items in some domain, the accuracy of a scoring classifier to reproduce this sequence can be measured by the percentage of items for which the classifier's score deviated from the true score (i.e., an integer in s) by at most δ. Such accuracy is mathematically defined as

$$\text{Acc-}\delta = |\{i : |s_i - s'_i| \leq \delta\}|,$$

where s' is an integer sequence produced by the classifier.

Term Frequency $\times$ Inverse Document Frequency (TF-IDF)

Term Frequency-Inverse Document Frequency (TF-IDF) is a single number statistic that reflects the importance of a term in a document within a corpus. The idea is that a term t is important for a document d if it occurs often within the document, and is relatively infrequent among other documents in the corpus D.

TF-IDF for a term-document pair is measured as a product of the term's frequency $tf(t, d)$ in d, and its inverse relative frequency across documents in the corpus, denoted as $idf(t, D)$. The specific definitions of tf and idf vary per application context. In the simplest case, the raw frequency of t's occurrence is taken as its term-frequency, and the logarithm of the ratio of total documents $|D|$ in the corpus to the number of documents containing t is taken as the inverse document frequency.

By measuring TF-IDF for each term in the corpus with V distinct terms (called its *vocabulary*), one can represent a document as a vector in $|V|$-dimensional space, whose entries represent TF-IDF values corresponding to terms in the vocabulary. Such a "vector space" representation enables comparison between any two documents in the corpus. For example, a popular way to assess similarity between two documents is to measure the cosine of the angle between their vectors. Such similarity is known as *cosine similarity* between the documents. As we will see at multiple occasions in this dissertation, TF-IDF can be a simple and effective weighting scheme in many situations, even when the conventional notions of terms and documents do not apply.

Chapter 3

Related Work

In this chapter, we discuss the problem of automated fact checking, looking at the state-of-the-art fact-checking algorithms, and methods designed for other tasks that are applicable to fact checking. Our focus is on methods designed to deal with information in knowledge graphs (KG).

Like any intelligent system based on KGs (d'Aquin and Motta, 2016; Russell and Norvig, 2003), a fact-checking system can be imagined to consist of two main components: (1) a *knowledge base* representing a vast continuously evolving repository of facts, and (2) a *fact-checking engine* that assesses the accuracy of dubious claims by reasoning about them and any previously known facts. In the following, we look at prior work related to both of these components.

3.1 Knowledge for Reasoning

Knowledge sources for fact checking can be in any form – they may be unstructured sources such as free-form text in blogs, encyclopedias, or Web documents; semi-structured sources such as Wikipedia infoboxes or web tables (Embley et al., 2006); or structured sources such as knowledge graphs like DBpedia (Bizer et al., 2009), YAGO (Hoffart et al., 2013), or Wikidata (Erxleben et al., 2014). Although all these sources may be, in principle, valuable for fact checking, challenges in natural language processing (NLP) such as named entity recognition and relation extraction (Cambria and White, 2014; Manning and Schütze, 1999; Sarawagi et al., 2008) limit the ability of current machines to truly understand unstructured text. Nevertheless, the availability of large quantities of structured information in KGs make them promising resources for designing novel fact-checking algorithms. Paulheim (2017) identifies a few defining characteristics of a KG, which include:

1. a KG describes facts about real-word entities and their relationships,

2. it defines a schema for representing relationships among its classes,

3. it allows to link arbitrary entities from heterogeneous sources, and

4. it covers a diverse set of topical domains.

In the previous chapter, we saw what knowledge graphs look like. Now, we look at some of their important properties such as their size and quality, which determine their usefulness for fact checking.

Creation techniques

KGs are either curated by human experts (e.g., WordNet (Miller, 1995), Cyc (Lenat, 1995)), constructed in a collaborative manner by a group of volunteers (e.g., Freebase (Bollacker et al., 2008), Wikidata (Erxleben et al., 2014)), created using automated extractors that uncover facts from semi-structured or unstructured Web sources (e.g., DBpedia (Bizer et al., 2009), YAGO (Hoffart et al., 2013)), or some combination of them (e.g., Knowledge Vault (Dong et al., 2014), NELL (Carlson et al., 2010)). The approach followed also determines the quality of extracted facts.

Coverage

Most KGs available today are far from complete in their coverage of facts available on the Web. Moreover, they often contains erroneous or conflicting facts due to imperfect creation techniques. Table 3.1 shows a summary of a few publicly available KGs. As seen, even the largest of these

Table 3.1: Size of a few large Knowledge Graphs.

KG	Entities	Relations	Facts
YAGO2[1]	9.7 M	114	447 M
DBpedia (English)[2]	4.6 M	3,441	104 M
Wikidata[3]	25.3 M	3,332	141 M
NELL (Paulheim, 2017)	2 M	425	432 K
OpenCyc[4] (Paulheim, 2017)	118 K	18.5 K	2.4 M

[1] Hoffart et al. (2013), Table 5

KGs are largely incomplete. This incompleteness has multiple dimensions: (1) the number of facts is quite small compared to the number of entities they have. (2) For many KGs, certain facts just cannot be stored due to limitations in design of their data model. For example, YAGO and Wikidata can store time and location information about facts, whereas DBpedia does not; Wikidata can store contextual and provenance information about facts they store, however, this capability is not available in most other KGs. And (3) the number of relations is relatively small compared to what can be expressed in a natural language. Paulheim (2017) surveys many approaches designed to refine KGs, and thereby improve their coverage.

Quality

To verify information about any entity under the sun, a fact checking agent requires access to facts that are accurate and consistent. Zaveri et al. (2016) unify and formalize many diverse notions of Linked Data quality, and provide a comprehensive list of the dimensions and metrics, which also apply to facts in KGs. They identify 23 dimensions of data quality grouped under the banners of accessibility, intrinsic, trust, data sets dynamicity, contextual, and representational dimensions. They note that although a number of tools exist to measure qualities of a KG or Semantic Web dataset along some of these dimensions, they generally require considerable amount of configuration, with limited usefulness in return. Although there are a few proposals for approaches to evaluate data quality (Kontokostas et al., 2014; Zaveri et al., 2016), we do not know of any work giving statistics of data quality in KGs measured along these formally defined dimensions. There is however some work on estimating the trustworthiness of Web sources (Dong et al., 2015) and evaluating the quality of facts extracted from noisy relation extraction techniques (Jain and Pantel, 2010; Paulheim and Bizer, 2014). Nevertheless, current research on KG refinement (Paulheim, 2017) and quality assessment (Zaveri et al., 2016) suggests that one can expect enhanced quality in the future.

[2]Type, mapping, raw-infobox based statements, http://wiki.dbpedia.org/dbpedia-2016-04-statistics

[3]As on Mar 20, 2017, http://tools.wmflabs.org/wikidata-todo/stats.php and https://www.wikidata.org/wiki/Category:Properties

[4]OpenCyc is a reduced version of Cyc, which is publicly available.

3.2 Detection and Tracking of Misinformation and Rumors

The design of methods to counteract the effects of misinformation spread on online social platforms is an active area of research. Just like the fight against web spam (Heymann et al., 2007), strategies underlying these methods can be broadly categorized into two main types: (1) a detection and containment strategy, whereby first, a piece of misinformation or rumor is detected by manual or semi-automatic ways, and then an action is taken to isolate and reduce its effect, and (2) a prevention strategy in which misinformation or rumor is verified in its early stages of circulation, and is prevented from propagating further across the network.

Methods following the first strategy delve into understanding factors that facilitate misinformation spread, identifying distinctive patterns of fake news or rumors vs. true organic content, and assessing credibility based on a set of discriminative features and patterns. In Chapter 1 we discussed approaches of this kind meant to detect bots, political astroturf campaigns, spam, and rumors. Briefly, methods assessing credibility of social content do so based on contextual information such as temporal patterns, source credibility, etc. For example, Mitra et al. (2017) assign credibility scores to tweets by learning a model based on a set of extracted linguistic cues. Ruchansky et al. (2017) use neural networks for learning temporal activities of users and source characteristics, which are then integrated to classify content as fake or real. Tacchini et al. (2017) show that posts on Facebook can be classified as hoax or non-hoax by learning a discriminative model based on a user's interaction with the posts, without using any cues from the content. A recent survey by Zubiaga et al. (2017) discusses existing work on rumor characterization, classification, and diffusion approaches.

3.3 Finding and Monitoring Claims to Check

By design, none of the aforementioned research attempts to check claims by reasoning about their content, thus limiting their efficacy in stemming the diffusion of false claims at an early stage. Reasoning methods for checking claims seek to follow the prevention strategy. In order to advance toward the goal of building a completely automated fact-checking system, the so-called "holy grail"

(Hassan et al., 2015), a few key sub-tasks have been identified that need further research: (1) monitor claims or articles worthy of public interest, (2) detect ones that are worthy of checking, and (3) verify them by reasoning about them and any relevant facts. These steps have also been identified as part of a broad roadmap for fact checking as laid out by *FullFact*.[5]

A few tools and systems have been developed recently to address the first two sub-tasks. FactMinder by Goasdoué et al. (2013) is a tool that allows fact checkers to annotate articles and match known entities to sources. Dispute Finder (Ennals et al., 2010) is a browser extension that automatically highlights a snippet of text in a web document as potentially related to a disputed issue, after consulting a corpus of disputed claims. It also shows a list of articles supporting alternative points of view. FactWatcher by Hassan et al. (2014) is a system that helps monitor interesting or attention-seizing facts as new information is added to an evolving database. It focuses on finding three kinds of facts:

— *Situational fact*: A situational fact (Sultana et al., 2014) is a statement about statistics of an interesting object in a specific context. Typically, such a statement talks only about a single object or entity of interest, e.g.,"The social world's most viral photo ever generated 3.5 million likes, 170,000 comments and 460,000 shares by Wednesday afternoon."[6] Here, the most viral photo in a social setting is the only entity being discussed, and hence an algorithm can safely consider the number of likes, comments, or shares to be about the photo.

— *One-of-the-few fact*: A one-of-the-few fact (Wu et al., 2012) is a statement about a particular object in a small collection of other objects, e.g., "Karl Malone is one of the only two players in NBA history with at least 25,000 points, 12,000 rebounds, and 5,000 assists in one's career."[7]

— *Prominent streak*: A prominent streak (Zhang et al., 2014) is a statement about a long-running event which is considered to be unusual in its context, e.g., "This month the Chinese capital has experienced 10 days with a maximum temperature in around 35 degrees Celsius—the

[5]https://fullfact.org/media/uploads/full_fact-the_state_of_automated_factchecking_aug_2016.pdf
[6]http://www.cnbc.com/id/49728455
[7]An example by Wu et al. (2012).

most for the month of July in a decade."[8] Here, the high temperature of around 35 degrees Celsius during a 10-day period in July constitutes an unusual event, considering the average temperature in July in Beijing's history.

Walenz et al. (2014) present two systems, namely *uClaim* and *iClaim*, that respectively focus on monitoring and checking claims by treating a claim as a parameterized query over a structured database. ClaimBuster (Hassan et al., 2017) is a system to identify check-worthy factual sentences in the transcripts of debates in U.S. presidential election. It uses natural language processing and supervised machine learning techniques to classify a sentence into one of three pre-defined types: non-factual sentence, unimportant factual sentence, and check-worthy factual sentence. The approach has been shown to be effective in that a set of sentences selected by journalists at two professional journalism organizations, namely CNN and PolitiFact, consistently received scores by their system that aligned with the rankings pre-determined by the journalists. Wu et al. (2017a) present an approach to surface a high-quality, diverse and representative set of top-k answers with applications for finding interesting leads or counterarguments to claims.

3.4 Veracity Assessment Methods

We now discuss the set of methods that are applicable to the problem of automatic verification of claims, to which we make our contributions through this dissertation. Most of these methods apply only to claims as simple as a triple, or an edge in a KG. A primary reason for this special attention to triples is the desire to make progress in checking elementary claims first, before moving on to more complex statements that can be represented as higher-arity tuples. For example, a ternary relationship `birthState(Barack Obama, Honolulu, Hawaii)` representing the sentence "Barack Obama was born in Honolulu which is located in Hawaii" can be expressed via two triples: `(Barack Obama, bornIn, Honolulu)` and `(Honolulu, locatedIn, Hawaii)`.

[8] http://www.chinadaily.com.cn/china/2010-07/27/content_11055675.htm

Before launching into the discussion, it is worth noting some of the desirable properties of a computational fact checker:

1. **Accuracy**. Veracity assessment of a claim should be accurate in that its decision should align with how a human fact checker would judge a claim.

2. **Scalability**. It should have the ability to efficiently navigate through the abundance of continually evolving background knowledge.

3. **Robustness**. It should be able to deal with missing and/or contradictory information, since consistency can be hard to expect in the realm of Semantic Web.

4. **Interpretability**. It should offer some form of explanation to the human fact checker as a justification for its decision.

Only a few algorithms we discuss below have all of these characteristics, making them promising choices for fact checking.

At a high level, approaches applicable to fact checking can be placed into five categories: (1) logical reasoning and rule mining approaches, (2) approaches based on vector space representations, (3) approaches that combine rule mining and vector space strategies, (4) similarity scoring approaches, and (5) relational graphical models.

Given the rules of inference for predicates in KGs, classic logical reasoning methods work by logical entailment, i.e., by application of these rules on an observed set of facts to deduce new facts. Being sound, these methods can be highly precise in checking whether a fact is true. However, they have limited applicability for information in KGs primarily due to two reasons: first, they require that knowledge expressed in the KG is consistent, i.e., there are no contradictory statements, which is at odds with how things operate on the Semantic Web where conflicting information is unavoidable; and second, they cannot scale in the face of large search spaces as found in KGs. Moreover, logical reasoning requires that the rules of inference are pre-determined. However, typically, such rules are not available. It is also hard to develop precise rules, because rules can often depend on the context

they are meant to cover. For example, one way to express two individuals are married is to specify a rule that says the two are married if they have kids together. In the context of spousal relationships of U.S. presidents however, an alternative rule could be one that says two individuals are married if their predecessors are married. To account for such context-specific rules besides natural semantics associated with a relation, rule-mining approaches (e.g., ILP (d'Amato et al., 2010), concept learning for building ontologies (Lehmann, 2009)) have been developed that use machine learning to mine interesting rules in a data-driven way. Such approaches seek to combine the expressive power of logical reasoning with the strengths of machine learning, namely the ability to deal with uncertainty and capture statistical patterns.

Vector space approaches on the other hand do not suffer from the challenges of scalability and noise in the KG data. They work by creating a global model of multi-relational data, taking into account many of the commonly known statistical patterns like homophily, block structure, etc. They are generally very accurate in their predictions, because of their ability to model implicit (or latent) knowledge and patterns in the KG. However, they too have their shortcomings: they can be quite complex to train, and offer little interpretability for fact checking.

Some approaches combine the best of both worlds, i.e., they seek to combine rule-mining and vector space models in certain ways, which not only improves performance, but helps circumvent shortcomings of the individual models.

Similarity-based approaches typically use information-theoretic or structural properties of the KG to assign a truth value or score. These methods are typically very efficient at making predictions, but they tend to perform sub-optimally because they do not take advantage of rich semantic information in KGs such as node and edge types.

Relational graphical models offer another alternative for modeling multi-relational data. They work by learning a model expressed as a graph of random variables, where the variables represent types of objects in the KG and the edges between them encode dependencies between the objects. In general, these models can be computationally complex to learn, thereby limiting their potential for use toward fact checking.

The relational character of KGs and the desire to capture statistical patterns makes Machine Learning (Mitchell, 1997) an ideal toolbox for learning problems in KGs. In the following, we look at a few representative methods from each of the aforementioned categories, many of which are machine learned models proposed by the Statistical Relational Learning (SRL) community (Getoor and Taskar, 2007). Note that methods from different categories often overlap in their underlying ideas.

3.4.1 Logical Reasoning and Rule Mining

First Order Inductive Learner (FOIL)

First Order Inductive Learner (FOIL) (Quinlan, 1990) is a representative inductive logical programming (ILP) technique (Muggleton et al., 1992) that learns a set of rules (called Horn clauses, or clauses in first-order logic) to define a relation in the KG. These rules can then be used to assess whether a given claim triple is true. It follows a covering paradigm: the first rule is learned such that the number of true triples that adhere to (or are covered by) the rule is maximized, while minimizing false positives. The next rule is then learned to cover true triples not covered by the previous rule, and so on. Although quite effective, FOIL struggles to scale up in case of large KGs.

Most ILP algorithms like FOIL follow a search strategy to define the optimal set of rules, either starting from an empty set and appending to it new rules as they are learned, or starting from a long rule and pruning them by removing sub-rules (called *atoms*) without compromising much in performance. Such rule-refining strategy determines an algorithm's efficiency; however, in general, learning first-order inference rules is computationally expensive. All ILP approaches operate under a Closed World Assumption (CWA) (see Section 2.5.1), which limits their applicability to KGs covering open domains. nFOIL (Landwehr et al., 2005) is a scalable variant of FOIL, and has been used for building the NELL (Carlson et al., 2010) KG. However, it works only for functional predicates.

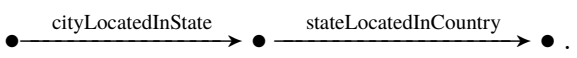
AMIE

AMIE (Galárraga et al., 2013) is an algorithm that uses association rule mining (Agrawal et al., 1993) to learn rules defining a predicate, which can then be applied to fact checking. It creates new rules by using a few rule-refinement operators that work by appending atoms to an existing rule. By retaining only those rules that have a *head coverage* above an empirical threshold, and by pruning long rules, AMIE ensures that a rule covers a substantive set of triples. Head coverage is a refined notion of the commonly known notion of *support* in association rule mining literature. Although AMIE is expressive because of its focus on mining interesting rules, it has recently been shown to be impractical for large knowledge graphs (Shi and Weninger, 2016).

Path Ranking Algorithm (PRA)

Path Ranking Algorithm (PRA) is a knowledge-base inference algorithm by Lao and Cohen (2010b), which assigns a score to a triple using a learned combination of paths (also called *experts*) connecting the triple's node pair. It has been successfully applied to information retrieval (Lao et al., 2011) and knowledge base completion (Dong et al., 2014) tasks.

PRA assumes that path patterns around true triples of a relation can be useful to predict future triples from that relation. The path patterns are assumed to take the form of *sequence of edge types* (or relational paths). For example, the true triples `(Pittsburgh, locatedIn, United_States)` and `(Cincinnati, locatedIn, United_States)` may share a relational path such as



Given a set of true and false triples from the `locatedIn` relation, PRA first mines such relational paths by simple graph traversal techniques, and uses these paths as features. A random walker then computes the probability of starting at the source node and arriving at the target node by following each such path. Such walks have been termed as *path-constrained random walks* (PCRW).

The probabilities computed based on these walks form the feature values of a learning problem. The resultant feature matrix is then fed to a standard supervised learning algorithm such as logistic regression, to learn a model and thereby make future predictions. Thus, PRA learns a model for each relation in the KG, which could be used for checking unseen claims.

PRA is a promising candidate for fact checking because: (1) it is known to have good performance for knowledge base completion tasks, and (2) it offers interpretability through its ranked features or relational paths, which can be seen as bodies of Horn clauses. The approach however has two drawbacks: (1) the calculation of feature values for a new triple based on PCRW can be computationally intensive due to the need to perform several random walks, and (2) the method results in sub-optimal performance when lexically different, but semantically similar relations exist in KG. Subsequent work by Lao and Cohen (2010a), Gardner et al. (2013) and Gardner and Mitchell (2015) has addressed some of these problems.

Predicate Path Mining (PredPath)

Closely related to PRA and designed specifically for fact checking using KGs, *PredPath* is an approach by Shi and Weninger (2016) that defines each relation in a KG using *discriminative predicate paths*. Discriminative predicate paths are relational paths (as defined for PRA), but with a difference that they also have node type information besides edge type. For example, one example of a discriminative path defining the *capitalOf* relation is

$$\{\text{city, settlement}\} \xrightarrow{\text{headquarterOf}} \{\text{state agency}\} \xrightarrow{\text{jurisdiction}} \{\text{state}\},$$

which intuitively captures what it means for a certain city to be the capital of a state. Here, the endpoint types or *anchors*, namely {city, settlement} and {state}, come from the types defined in an ontology.

The method follows a high level strategy similar to that of PRA — first, anchored predicate paths are extracted using graph traversal algorithms, they are then filtered using a feature selection scheme to retain a set of discriminative paths as features, and then a supervised learning framework

is used to predict unseen triples using a learned model. PredPath has been shown to be effective on many synthetic fact-checking datasets.

The specific differences between PredPath and PRA lie in the definition of an "expert," how such experts are extracted from the graph, the feature selection scheme used, and the statistics of the features used for learning a model. PRA simply uses a sequence of relations as an expert, whereas PredPath uses the notion of anchored predicate path; PRA employs random walks, whereas PredPath uses depth-first search to navigate the path space; PRA uses precision and recall for selecting features, whereas PredPath uses information gain to weed out uninformative paths.

In both PRA and PredPath, significant computation effort is spent on enumerating paths, many of which are eventually thrown away. In passing, we note that PredPath is closely related to similarity-based approaches by Sun et al. (2011) and Shi et al. (2014) designed for heterogeneous information networks. All the three approaches make use of predicate paths, and the latter two are also applicable to fact checking. However, they are not scalable because they assume that predicate paths are pre-determined and provided as input, whereas for fact checking, such paths are generally not available.

3.4.2 Vector Space Approaches

The key idea behind vector space approaches is to model the observed triples in a KG via a set of latent features for nodes and/or relations. By doing so, the resulting models aim to explain observed statistical dependencies (or patterns) using latent features. Many models have been proposed in the context of knowledge graph completion (KGC), which can be also applied to fact checking. Methods differ in the kind of assumptions they make about patterns in a KG. These assumptions also determine their model complexity and efficacy for link prediction. Model complexity for vector space methods is measured in terms of number of parameters required by a model.

One way to model relational data in KGs is to consider a matrix representation of the graph (i.e., one matrix per relation), and perform a low-rank matrix factorization (e.g., by a principal component analysis (Pearson, 1901)). The similarity between the factored representations of two

nodes can then be used for assigning a score to a triple. However, such approach does not capture the dependencies across relations. To overcome the problem, the idea of factorizing individual matrices has been extended to factorizing sets of related matrices in such a way that latent factors of entities are shared among relations. Collective matrix factorization proposed by Singh and Gordon (2008) is a technique following this idea, and has been proven to be effective for movie recommendation and brain imaging tasks. However, even these extended approaches are only applicable when there is a small set of relations in the domain, which limits their use for fact checking based on KGs, which typically have in the order of thousands of relations.

Another idea using matrices is to model a set relation matrices as a tall (or wide) matrix obtained by stacking them vertically (or horizontally). E.g., see work by Jiang et al. (2012) and Riedel et al. (2013). However, in these approaches, there is usually a substantial loss of useful information due to the transformation. For example, models learned using these approaches fail to account for the fact that an entity may be a subject as well as an object under different relationships. Not accounting for such situations limits their effectiveness at link prediction and other tasks.

Approaches based on a third-order tensor representation and neural networks allow for richer representations that can capture many patterns. Examples in this line of work include Bayesian Clustered Tensor Factorization by Sutskever et al. (2009), RESCAL by Nickel et al. (2011), Structured Embedding by Bordes et al. (2011), Semantic Matching Energy by Bordes et al. (2014), and many improved variants such as ones by Franz et al. (2009), Jenatton et al. (2012), Drumond et al. (2012), and Erdos and Miettinen (2013). In recent years, numerous proposals have been made for KGC to improve their coverage, which follow a tensor factorization approach, e.g., TransE by Bordes et al. (2013), TransH by Wang et al. (2014), TransR by Lin et al. (2015), HoIE by Nickel et al. (2016b), ProjE by Shi and Weninger (2017), TransHR by Zhang et al. (2017), TransPES by Wu et al. (2017b), and Poincaré embeddings by Nickel and Kiela (2017). Guu et al. (2015) and Neelakantan et al. (2015) show that a successive application of these models (also called *compositionalization*) can further improve performance at KGC task. DeepPath by Xiong et al. (2017) is a latest variant of this family, which uses a reinforcement learning approach to learn multi-hop paths using vector space

embeddings.

All approaches designed for KGC learn so-called *vector embeddings*, an alternative term for vector space representation of entities and relations. In general, they offer better generalization due to their ability to capture rich statistical patterns. However, the majority of them can be quite complex to train for large KGs, limiting their applicability to fact checking. See (Shi and Weninger, 2017, Table 1) for a comparison of model complexity of many of these models. Nickel et al. (2016a) comprehensively survey many of these models proposed for learning from KGs. We review a few of them below.

RESCAL

RESCAL[9] by Nickel et al. (2011) views a KG as a third-order adjacency tensor, and models each frontal slice corresponding to a relation as a product of an entity matrix, a relation matrix and the entity matrix transposed. By minimizing reconstruction error, it learns the entity matrix and relation matrix that symbolize the latent representation for entities and relations respectively. It is one of the first tensor models for relational data. A unique feature of RESCAL is that it finds a unique representation for each entity, which captures its dual roles of subject and object in distinct triples. RESCAL has been shown to be effective at link prediction and taxonomy learning tasks on large-scale KGs such as YAGO and DBpedia, and thus is also applicable to fact checking. However, the model's expressiveness comes at a cost of higher computational complexity.

TransE

TransE by Bordes et al. (2013) aims to create an easy-to-train, scalable model for link prediction, while capturing hierarchical relationships in a KG. It assumes that relationships between entities in a triple can be measured by their closeness in latent space. It assumes that for a true triple (s, p, o), the translation of s's vector by an amount equivalent to p's vector should be close to o's vector. That is, if $\mathbf{s}, \mathbf{p}$ and $\mathbf{o}$ respectively represent the vectors of s, p and o, then the distance between $\mathbf{s} + \mathbf{p}$ and

[9] **RE**lational **SCAL**ing

o should be small. The embeddings are learned by minimizing an energy function that represents the notion of distance. TransE has been shown to be computationally efficient, and works fairly well for one-to-one relationships in KGs such as Freebase. However, it fails to model one-to-many and many-to-many relationships, which are also prevalent in KGs.

TransH

To overcome the limitations of TransE, Wang et al. (2014) proposed TransH, which models each relation by a hyperplane, and follows the idea that similarity between a pair of nodes should be based on a translation operation carried out in the triple's relation-specific hyperplane. Like TransE, it first seeks an embedding of subject and object entities, **s** and **o**, and then performs a projection of these vectors to the hyperplane associated with relation p, obtaining $\mathbf{s}_\perp$ and $\mathbf{o}_\perp$. It assumes that if the triple is true, the translation on this hyperplane $\mathbf{s}_\perp + \mathbf{d}_p$ should be close to $\mathbf{o}_\perp$, where $\mathbf{d}_p$ is a translation vector on the hyperplane. The nature of objective function, learning recipe and evaluation criteria are essentially the same as those for TransE. By focusing on translations only in a relation's hyperplane, TransH is known to have improved performance on same tasks and data sets. Thus, it overcomes some of the problems of TransE, while retaining its computational efficiency.

TransR

TransE and TransH create embeddings of entities and relations in the same latent space. Lin et al. (2015) argue that such a common space may not be adequate, and thus propose to use distinct spaces for entities and relations in their model called TransR. In a similar vein as TransH, TransR projects vectors of s and o onto p's space (relation-specific space), and expects that, for a true triple, the translation of s' projected vector is close to that of o. Again, the nature of objective function, learning and evaluation process are the same as those of TransE. TransR has shown to outperform TransE and TransH at link prediction tasks in WordNet and Freebase. However, its richer expressiveness comes at the expense of higher computational complexity, which is unattractive for large KGs.

ProjE

The key assumption made by TransE, TransH, and TransR is that vector representations of two entities of a true triple are close when a translation operation is at work. Instead of translation, a few other models such as Knowledge Vault (Dong et al., 2014) and HoIE (Nickel et al., 2016b) have proposed alternative *combination* operators to improve performance. Projection embedding model (ProjE) by Shi and Weninger (2017) is one of the recent approaches that defines yet another combination operator. This operator consists of a linear combination of the entity and relation embedding vectors, whose coefficients represent a learned set of global entity and relationship weights represented as diagonal matrices. It models data using far fewer parameters, and is comparably more efficient that prior models. Despite its low model complexity, it has shown to offer superior performance at link prediction on previously considered datasets.

3.4.3 Combining Rule Mining and Vector Space Approaches

There is also some research on designing approaches that combine the complementary strengths of supervised rule learning models and vector space models, which can be used for predicting a truth value. Rule learning approaches like PRA are promising for their ability to capture local patterns such as relational paths while being computationally efficient, but may not capture patterns that can be explained via latent features. Vector space approaches can fill this gap. One explored way of combining such models is to learn them jointly, which can also reduce the complexity of the vector space models. See for example approaches by Nickel et al. (2014) and Jiang et al. (2012). Another approach is to build different prediction models separately and use their outputs as features (or inputs) to build a high level classifier. This approach is called stacking (Breiman, 1996). It has been shown to be successful for information extraction from heterogeneous sources on the Web (Dong et al., 2014).

3.4.4 Similarity-Based Approaches

Similarity-based measures work by assigning a score to a pair of nodes by either leveraging a KG's topological characteristics, or information content of its nodes, or following some combination of the two.

Early measures were mainly designed to work for taxonomic resources such as WordNet (Miller, 1995). Here, the goal was to improve upon simple edge-counting strategies (e.g., by Rada et al. (1989)), and take into account the semantics encoded in their hierarchical arrangement, or through the derived information content of their concept nodes. For example, Resnik (1995) assigned node similarity based on the information content in the least common ancestor of the two nodes. This meant that all nodes with a common ancestor were assigned same scores, which was limiting. Hence, improved approaches (e.g., by Jiang and Conrath (1997) and Lin (1998)) followed that took into account both commonalities and differences between the nodes. However, these improved approaches were not designed to work on graphs or ontologies, which, besides a hierarchical component, contain links between arbitrary nodes (called cross-links) as in a KG.

Maguitman et al. (2006) were the first to propose an information-theoretic semantic similarity measure for nodes in a graph. The idea was an extension of Lin's approach for taxonomies, which now accounted for a few types of cross-links. Subsequently, in the context of a folksonomy, Markines and Menczer (2009) proposed a measure called *maximum information path* (MIP) to assess similarity between tags and resources. MIP extended Lin's idea as well as the traditional notion of shortest path by using Shannon's information content of tags/resources. Other graph-based measures either use in-link/out-link structure (Milne and Witten, 2008), compute personalized-PageRank on Wikipedia article graph (Yeh et al., 2009), or use a combination of Wikipedia category system and article text (Gabrilovich and Markovitch, 2007) to measure semantic similarity between node pairs.

In principle, any link prediction algorithm is applicable to fact checking. There is a panoply of link prediction algorithms that work by assigning a similarity score to a pair of nodes, far too many to cover them all here. They are mainly categorized into three types: (1) local indices: algorithms that use structural properties local to the node pair, such as neighborhood information; (2) global indices:

algorithms that take advantage of complete network topology, such as ensemble of paths connecting the nodes; and (3) quasi-local indices: algorithms that use more topological information than just node neighborhood, yet do not require complete network topology. Methods such as Adamic & Adar (Adamic and Adar, 2003), Leicht-Holme-Newman index (Leicht et al., 2006) and Degree Product (Shi and Weninger, 2016) are examples of local indices, whereas Katz (Katz, 1953) and SimRank (Jeh and Widom, 2002) are examples of the global indices. Local indices can be very efficient to compute, but normally have low predictive power. On the other hand, global indices have good predictive power, however suffer from a high computational cost. Quasi-local indices such as Local Path Index (LP) (Lü et al., 2009), Local Random Walk (LRW) (Liu and Lü, 2010) and Superposed Random Walk (SRW) (Liu and Lü, 2010) offer trade-offs between local and global indices by only considering a subset of paths (e.g., paths up to length three) or few-step random walks. Getoor and Diehl (2005), Liben-Nowell and Kleinberg (2007) and Lü and Zhou (2011) survey a range of such measures proposed for machine learning settings, social networks and complex networks. Work by Harispe et al. (2015) also offers an in-depth account of many of the methods designed for assessing semantic relatedness. However, most of these link prediction algorithms tend to perform poorly because they do not take advantage of semantic information (e.g., edge and node type information) available in KGs.

3.4.5 Relational Graphical Models

Relational Graphical Models aim to capture the inherent structure of a KG using a set of inter-connected random variables. The random variables represent the type of objects in the KG, and the directed or undirected links represent the statistical dependencies between these variables. Learning the model entails learning the topology of the graph of variables and estimating a group of parameters for the assumed distributions around these variables. Such graphical models allow us to combine the expressive power of logic (the representation language of Semantic Web) and the probabilistic semantics encoded in the graph topology to handle noisy and missing data in KGs. Predicting a missing link then amounts to estimating the conditional probability of the link given the estimated

structure and parameters. This is also called relational inference, and is equivalent to performing fact checking.

Markov Logic Network (MLN) proposed by Richardson and Domingos (2006) is an attractive language in the undirected models category, which allows one to define a probability distribution over possible worlds using first-order logic formulas (or clauses), and put a weight on each formula. Learning in MLN amounts to learning the structure (i.e., the set of clauses) as well as their associated weights, both of which can be very expensive (Natarajan et al., 2012; Neville and Jensen, 2007) because of the size of the grounded graph. A gradient-boosting algorithm by Khot et al. (2015) has however shown that both, learning weights and structure, can be performed simultaneously. Work by Niu et al. (2011) has also aimed to improve scalability of MLNs. Although MLNs and other graphical models may be useful for fact checking, they are in general difficult to scale with the size of a KG, namely the large number of relations and node types.

3.5 Remarks

In this chapter, we covered a wide variety of approaches applicable to fact checking. We saw that they mainly differ in (1) their ability to adequately capture logical relationships and statistical dependencies, (2) computational efficiency, and (3) interpretability. Models inspired by logical reasoning tend to be accurate and interpretable, but suffer from poor scalability, whereas vector space models are highly accurate and often scalable, but can be quite complex to train, offering little interpretability as needed for fact checking. Although all of the discussed approaches qualify for fact checking, one should prefer a method that strikes a balance between accuracy, scalability, robustness, and interpretability.

Chapter 4

Fact Checking by Shortest Path

In this chapter, we look at intuitive models for fact-checking triples, which rely on explanations provided by an alternative chain of facts.

4.1 Motivation

Consider the task of checking the claim triple `(Barack Obama, spouse, Michelle Obama)`. We have at our disposal a large-scale knowledge graph such as DBpedia that contains information about entities Barack Obama and Michelle Obama. Figure 4.1 shows a sub-graph of DBpedia as it pertains to these entities. If a direct edge of type *spouse* already exists between them, one can say that the claim is true, and fact checking trivially amounts to looking up whether such edge exists. However, an absence of this direct edge makes veracity assessment of the triple challenging. In such cases, we hypothesize that the set of indirect paths between the node pair can be useful for fact checking the triple.

A path in the sub-graph represents a high-level fact formed by a chain of simple facts. For example, the path *Barack Obama* $\xrightarrow{\text{religion}}$ *Protestantism* $\xrightarrow{\text{religionOf}}$ *Michelle Obama* represents the fact the two individuals follow the same religion of Protestantism. When their *spouse* relationship is under question, some paths stand out to be more relevant than others. For example, one path corresponding to the fact that Barack Obama has a relative named Marian Shields Robinson who is also the mother of Michelle Obama, is a strong indicator that the two might be married. Another path corresponding to fact that the two are respectively successors of the couple, George W. Bush and Laura Bush, is another useful evidence of their spousal relationship. However, other facts which indicate that they both are lawyers by profession, or belong to the same party (Democratic) provide

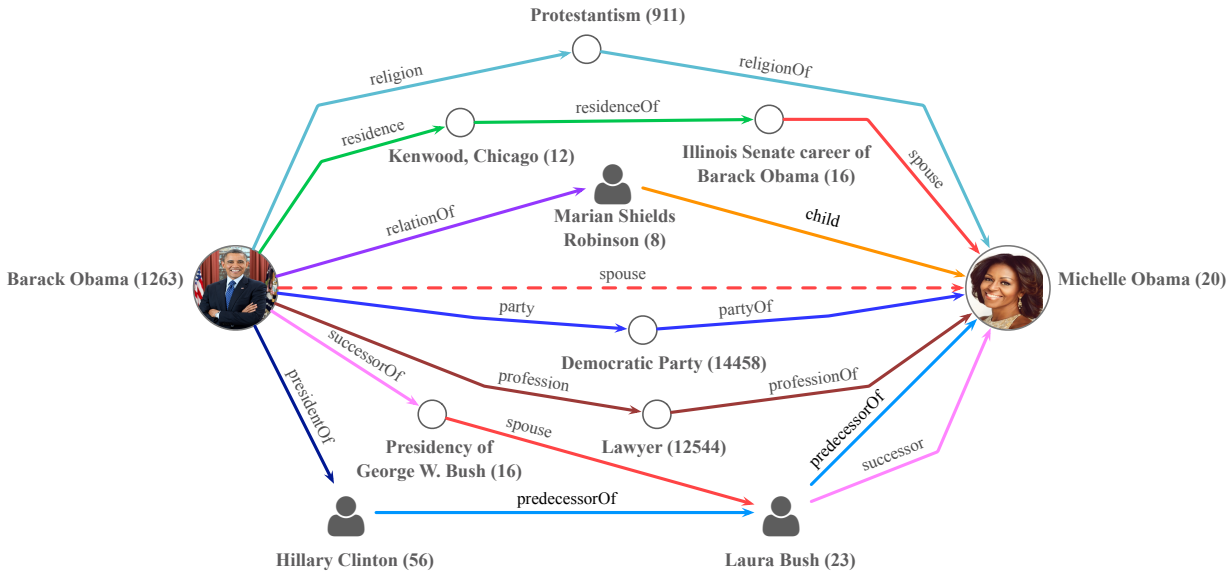


Figure 4.1: A sub-graph of DBpedia showing a few paths connecting Barack Obama and Michelle Obama. Colors of edges represent distinct relations in the graph. Numbers in parenthesis indicate node degrees. The dotted edge represents the predicate *spouse* under evaluation.

little support for the claim, since such facts may also apply to other triples, and can be considered to be quite *general*. Moreover, these facts are not intuitively as meaningful for the claim in question. The previous two facts on the other hand are more *specific* to the couple in question, and intuitively convey useful information in support of the claim.

Based on these observations, we assume that a path may be useful for fact checking if it is (1) specific (or peculiar to a claim, rather than general), and (2) semantically related to the predicate in question. In this chapter, we look at identifying such path for claim triples. We first look at fact checking simply based on specific paths, and then extend the approach to include their edge semantics.

Let a knowledge graph be denoted by an undirected graph $G = (V, E, \mathcal{R}, g)$, where V is the set of $|V|$ concept nodes, E is the set of $|E|$ edges inter-connecting these nodes, $\mathcal{R}$ is the set of $|\mathcal{R}|$ distinct relations (or edge types), and $g : E \rightarrow \mathcal{R}$ is a mapping that labels each edge with a semantic relation or predicate.

4.2 Specificity of a Path

What makes a path indicative of a general fact is that it includes general concept nodes. A crude way to measure how general a concept node is by its structural property, the degree k. The higher the degree of a node – the more the facts in KG about it – the more general the associated concept is. In the example above, Democratic Party, Lawyer and Protestantism represent general concept entities since they have a large number of connections (14458, 12544 and 911 respectively in DBpedia). We thus define the *generality* of a node v simply as the logarithm of its degree $\log(v)$. We use logarithm of the degree based on information-theoretic arguments; alternative choices could of course be explored.

Conversely, the lower the degree of a node, the more *specific* is the corresponding concept. For example, Marian Shields Robinson, Kenwood, Chicago and Presidency of George W. Bush represent specific concepts with very few connections (8, 12 and 16 respectively in DBpedia). Therefore, analogously, the *specificity* of a node v can be defined as inversely proportional to its generality:

$$s(v) = \frac{1}{1 + \log k(v)}, \tag{4.1}$$

where we consider generality of v as a measure of length of the edge (u, v) (and specificity of v as a proximity weight of the edge (u, v)), and use the distance-to-similarity function (Eqn. 2.1) to convert generality into specificity.

If we create a distance graph in which each edge (u, v) is weighted by the generality of the incident node v, one could apply the notion of transitive closure defined for distance graphs (Simas and Rocha, 2015) to derive the *generality* of a path. The transitive closure of an unweighted graph is a graph in which the set of edges are closed under adjacency, i.e., two nodes are adjacent *iff* there is at least one path that connects them. The notion of transitive closure of unweighted graphs has been extended to weighted graphs (Simas and Rocha, 2015). Accordingly, the specificity of a path $P_{s,p,o}$ with n nodes between s and o (i.e., $P_{s,p,o} = v_1 v_2 \dots v_n$) can be derived by aggregating the

specificity of its intermediate nodes:

$$\mathcal{S}(P_{s,p,o}) = \frac{1}{1 + \sum_{i=2}^{n-1} \log k(v_i)}. \tag{4.2}$$

Note that we do not penalize a path for the generality of the target node o. Intuitively, two nodes are close if there are only a few intermediate nodes with a small number of connections.

Given the set of paths $\mathcal{P}_{s,p,o}$ between s and o, we model the task of fact checking a triple (s, p, o) as one of finding a path with maximum specificity. We assign a truth value to the triple equivalent to the path's specificity:

$$\tau^{\text{KL}}(s, p, o) = \max_{P_{s,p,o} \in \mathcal{P}_{s,p,o}} \mathcal{S}(P_{s,p,o}) \tag{4.3}$$

We call this approach for fact checking the *Knowledge Linker* (KL).

Since specificity is inversely proportional to generality, and generality represents the notion of length of a path, we can say that fact checking by KL amounts to finding the shortest path between s and o. Such shortest path can be found by a transitive closure of the distance graph using a pair of operators $(\min, +)$, where the $+$ operator integrates along a path by summing up the node generalities, and the $\min$ operator evaluates the set of paths $\mathcal{P}_{s,p,o}$ and chooses one with minimum generality. This transitive closure is called the *metric closure*.

Conversely, we can derive the specificity of a path by computing the transitive closure of an equivalent proximity graph in which the proximity weight associated with an edge is equal to the specificity of the incident node. We want to prove that the specificity of a path $P_{s,p,o}$ connecting s and o is given by Eqn. 4.2. The proof is straightforward by induction. Consider a path $v_1, v_2, v_3, v_4, \ldots, v_n$ where $v_1 = s$ and $v_n = o$. We consider as the base case its sub-path of length two: v_1, v_2, v_3 for

which we derive its specificity:

$$\begin{aligned}
\mathcal{S}(P_{v_1,v_3}) &= \wedge(\mathcal{S}(P_{v_1,v_2}), \mathcal{S}(P_{v_2,v_3})) = \wedge(w(v_1,v_2), w(v_2,v_3)) \\
&= \frac{\frac{1}{1+\log k(v_2)} \cdot \frac{1}{1+\log k(v_3)}}{\frac{1}{1+\log k(v_2)} + \frac{1}{1+\log k(v_3)} - \frac{1}{1+\log k(v_2)} \cdot \frac{1}{1+\log k(v_3)}} \\
&= \frac{\frac{1}{1+\log k(v_2)} \cdot \frac{1}{1+\log k(v_3)}}{\frac{1+\log k(v_3)+1+\log k(v_2)-1}{(1+\log k(v_2))\cdot(1+\log k(v_3))}} \\
&= \frac{1}{1+\log k(v_2)+\log k(v_3)} \\
&= \left[1+\log k(v_2)+\log k(v_3)\right]^{-1}.
\end{aligned}$$

By induction, assuming that we know the specificity of the sub-path of length $n-1$, we can calculate the specificity of the path $v_1, v_2, v_3, v_4, \ldots, v_n$:

$$\begin{aligned}
\mathcal{S}(P_{v_1,v_n}) &= \wedge(\mathcal{S}(P_{v_1,v_{n-1}}), \mathcal{S}(P_{v_{n-1},v_n})) = \wedge(\mathcal{S}(P_{v_1,v_{n-1}}), w(v_{n-1},v_n)) \\
&= \left[1+\log k(v_2)+\log k(v_3)+\cdots+\log k(v_{n-1})\right]^{-1} \\
&= \left[1+\sum_{i=2}^{n-1}\log k(v_i)\right]^{-1}
\end{aligned}$$

where we have used the fact that for the last edge, $S(v_{n-1}, o) = 1$. Thus, in general, for a path $P_{s,p,o}$ of length n, $\mathcal{S}(P_{s,p,o}) = \left[1+\sum_{i=2}^{n-1}\log k(v_i)\right]^{-1}$.

However, instead of $(\min, +)$, Simas and Rocha (2015) show that one could employ an alternative pair of operators $(\max, \min)$ to compute a distinct transitive closure called the *ultra-metric closure*, in which adjacency of two nodes is defined based on the widest bottleneck of paths connecting them. Such closure suggests an alternative way for fact checking. The specificity of a path under this closure is given as:

$$\mathcal{S}_{\mathrm{u}}(P_{s,p,o}) = \begin{cases} 1 & n = 2 \\ \left[1+\max_{i=2}^{n-1}\log k(v_i)\right]^{-1} & n > 2, \end{cases} \tag{4.4}$$

and the truth value is equal to the maximum of specificities across all paths

$$\tau_{\mathrm{u}}(s, p, o) = \max_{P_{s,p,o} \in \mathcal{P}_{s,p,o}} \mathcal{S}_{\mathrm{u}}(P_{s,p,o}). \tag{4.5}$$

Given two approaches for fact checking, one based on metric closure and another on ultra-metric closure, which one should be used? We empirically answer this question in Section 4.4. In answering the question, we also evaluate how these approaches fare when a directed graph, in which edges retain their original directionality, is used instead of an undirected graph as we have assumed thus far.

4.3 Relational Similarity

KL performs fact checking by finding the most specific path connecting s and o. However, this approach ignores the semantics of target predicate p, which implies two issues. First, two triples with same node pair but different predicates will receive the same score. Second, two distinct shortest paths with same specificity are equally plausible candidates, which may not be meaningful because their semantics might differ.

To overcome these problems, we hypothesize that biasing the search for specific paths while favoring edges that are semantically related to p should improve KL. Intuitively, the more similar each edge predicate p' is to p, the more semantically relevant the path is to the triple. For instance, if we were to ascertain whether Paris is the capital of France, facts about location of governmental headquarters might be more relevant than that about cultural heritage. To this end, we first look at a novel data-driven approach to measuring similarity between predicates. We then incorporate these similarities into Knowledge Linker.

We assume that two predicates are similar if they tend to co-occur in KG. The line graph $L(G)$ (Section 2.2) of the KG G allows us to study such co-occurrence between relations by considering their adjacency. However, since it includes duplicate nodes corresponding to duplicate relations in G, it is not suited to define a similarity metric on $\mathcal{R}$. We overcome this problem by contracting

duplicate nodes until there is exactly one node for each element of $\mathcal{R}$. A graph can be *contracted* by replacing two nodes with a new node whose set of neighbors is the union of their neighbors in $L(G)$. Rather than duplicating edges, the contracted graph is edge-weighted; the weight of a new edge reflects the number of old edges that are merged in the contraction. We thus start with G, then build $L(G)$ setting all edge weights to 1, and finally we iteratively contract pairs of nodes labeled with the same relation, until there are no duplicate labels. We call the resulting graph the *contracted line graph*, denoted by $L^*(G)$. See Figure 4.2 for an example of a small KG with four relations and five nodes.

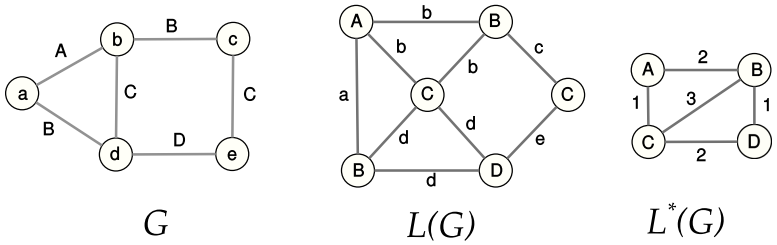


Figure 4.2: Example of the line graph $L(G)$ and the contracted line graph $L^*(G)$ of a simple knowledge graph G with four relations (denoted by uppercase letters) and five nodes (lowercase letters). The edge weights in $L^*(G)$ represent how often each relation is co-incident to its neighbors in G.

Let us denote with $C \in \mathbb{N}^{R \times R}$, where $R = |\mathcal{R}|$, the adjacency matrix of the contracted line graph. By construction, C is the co-occurrence matrix of $\mathcal{R}$, which could be used toward estimating similarity between relations, as is done in natural language processing for measuring relatedness between words (Barrière, 2016). We present below three measures of such similarity.

4.3.1 Cosine Similarity using a TF-IDF Representation

The raw co-occurrence counts in C are dominated by the most common relationships. Therefore, as customary in information retrieval, we apply TF-IDF weighting to C:

$$\begin{aligned} \text{TF}(r_i, r_j) &= \log(1 + C_{ij}), \\ \text{IDF}(r_j, \mathcal{R}) &= \log \frac{R}{|\{r_i | C_{ij} > 0\}|}, \\ C'(r_i, r_j, \mathcal{R}) &= \text{TF}(r_i, r_j) \cdot \text{IDF}(r_j, \mathcal{R}) \end{aligned} \tag{4.6}$$

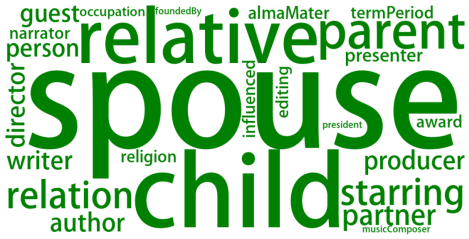
where C_{ij} is the co-occurrence count between $r_i \in \mathcal{R}$ and $r_j \in \mathcal{R}$. We define the *relational similarity* $u_{\cos}(r_i, r_j)$ as the cosine similarity between the i-th and j-th rows of C'. Figure 4.3 shows a few examples based on the cosine similarity $u_{\cos}$.

4.3.2 Pointwise Mutual Information

Another approach of measuring relational similarity is through the notion of Pointwise Mutual Information (PMI). Accordingly, the relational similarity between a pair of relations r_i and r_j is measured as:

$$u_{\text{pmi}}(r_i, r_j) = \log \left(1 + \frac{p(r_i, r_j)}{p(r_i)p(r_j)} \right) \tag{4.7}$$

where the numerator represents the observed joint probability of the two relations, and the denominator represents the expected probability of the relations to occur together if they are independent. The 1 inside logarithm ensures that the term is well-defined. Thus, the ratio measures how often than expected by chance would we expect the two relations to co-occur. The quantities can be estimated empirically as $p(r_i, r_j) = \frac{C_{ij}}{\sum_{i,j} C_{ij}}$ and $p(r_i) = \frac{\sum_j C_{ij}}{\sum_{i,j} C_{ij}}$. Figure 4.4 shows the top similar relations for the same predicates as in Figure 4.3 using this measure.

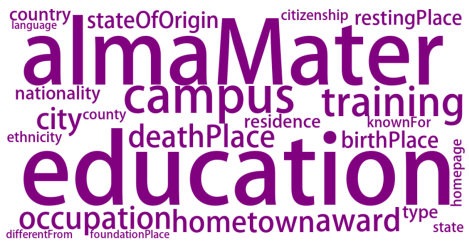


(a) spouse

(b) director

(c) capital

(d) battle

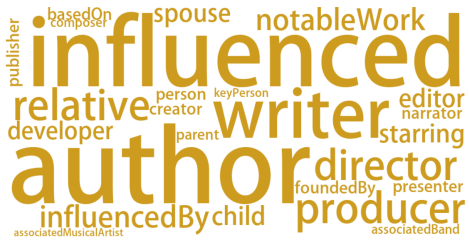


(e) education

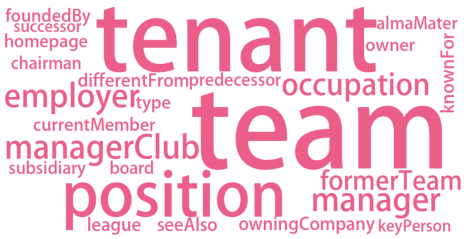
(f) vicePresident

(g) nationality

(h) keyPerson

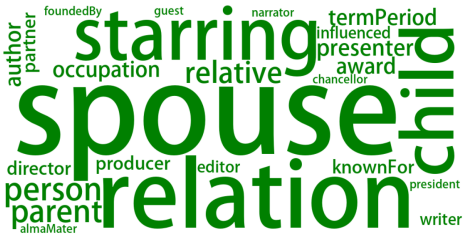


(i) author



(j) team

Figure 4.3: Top 20 most similar relations for a few predicates in DBpedia, using the cosine similarity measure. The font size is proportional to the relational similarity.



(a) spouse

(b) director

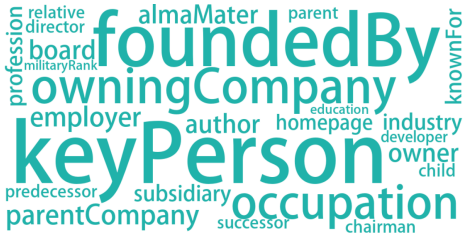
(c) capital

(d) battle

(e) education

(f) vicePresident

(g) nationality



(h) keyPerson

(i) author

(j) team

Figure 4.4: Top 20 most similar relations for a few predicates in DBpedia, using pointwise mutual information. The font size is proportional to the relational similarity.

4.3.3 Personalized PageRank

A third way to measure similarity between relations is through the notion of personalized PageRank (Haveliwala, 2002). The personalized PageRank x'_j of a relation r_j in a line graph represents the limiting probability of a random walk to arrive at r_j by starting at r_i. The personalized PageRank vector $\mathbf{x}'$ corresponding to all relations can be estimated in an iterative manner using the following equation:

$$\mathbf{x}' = \theta A\mathbf{x} + (1-\theta)\mathbf{e}/R \tag{4.8}$$

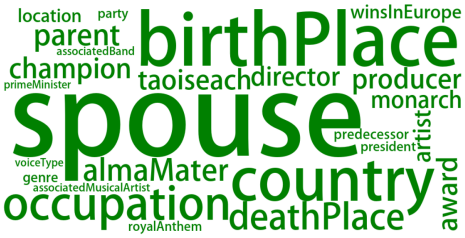
where θ is a chosen constant (usually between 0.8 and 0.9), A is a row-stochastic transition matrix obtained by normalizing the rows of C, $\mathbf{x}$ is the current estimate of PageRank vector, and $\mathbf{e}$ is a personalization vector. The first term in the above equation corresponds to the case where the random walker chooses one of the outgoing edges from its current node, and the second term corresponds to the probability that the walker randomly returns to a particular node.

A simple way of measuring the relational similarity between r_i and r_j is to take its value to be the PageRank x'_j of relation r_j by personalizing for r_i, i.e., we take $e_i = 1$, and 0 otherwise. However, since the PageRank of nodes in a directed graph is biased toward high-degree nodes such as `birthPlace`, as suggested in recent literature (Bánky et al., 2013; Shin et al., 2014), we adjust the PageRank of r_j by dividing by its indegree $k^{\text{in}}(r_j)$ to correct for such bias. Thus, the relational similarity between r_i and r_j is given by

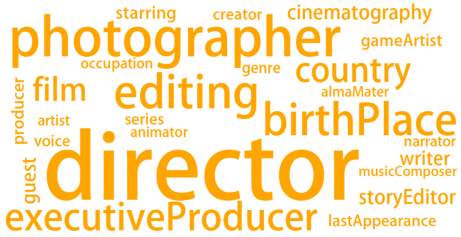
$$u_{\text{ppr}}(r_i, r_j) = p(r_j|r_i) = \frac{x'_j}{k^{\text{in}}(r_j)} \tag{4.9}$$

Figure 4.5 shows the top similar relations using this measure for the same predicates as in Figure 4.3 and Figure 4.4.

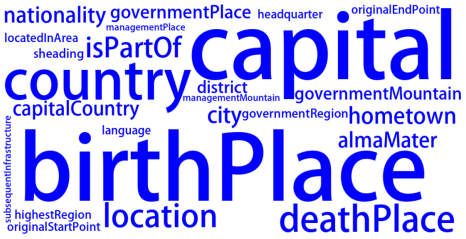
Comparing Figure 4.3, Figure 4.4 and Figure 4.5, one can see that for a given predicate, there is an overlap among the top similar relations across the three measures. However, they differ in their



(a) spouse

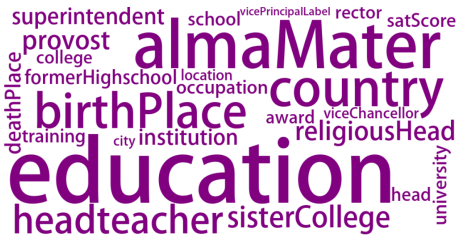


(b) director



(c) capital

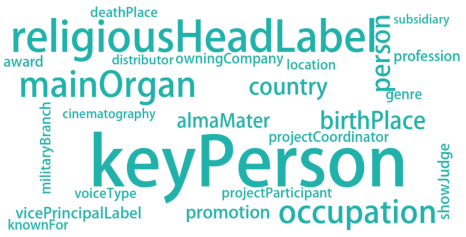
(d) battle



(e) education

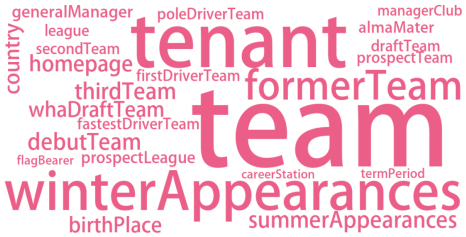
(f) vicePresident

(g) nationality



(h) keyPerson

(i) author



(j) team

Figure 4.5: Top 20 most similar relations for a few predicates in DBpedia, using personalized PageRank. The font size is proportional to the relational similarity.

rankings. We compare the effectiveness of the three measures on fact checking in Section 4.6.2.

With the similarity between any pair of predicates defined, we now extend Knowledge Linker to include relational similarity:

$$\mathcal{S}'(P_{s,p,o}) = \left[\sum_{i=2}^{n-1} \frac{\log k(v_i)}{u(r_{i-1}, p)} + \frac{1}{u(r_{n-1}, p)}\right]^{-1}. \tag{4.10}$$

This formulation maximizes the relational similarity between the predicate p' corresponding to an edge and p, in addition to the specificity of the intermediate nodes. The last term allows to include the relational similarity of the last edge without penalizing a path for o's generality. The truth score of the triple (s, p, o) is then same as before:

$$\tau^{\text{KL-REL}}(s, p, o) = \max_{P_{s,p,o} \in \mathcal{P}_{s,p,o}} \mathcal{S}'(P_{s,p,o}) \tag{4.11}$$

We call this extended approach the *Relational Knowledge Linker* (KL-REL). By incorporating relational similarity in its search, KL-REL is also a shortest path algorithm, and retains the computational efficiency of KL.

4.4 Calibration of Fact Checker

Given the choices of distinct path evaluation measures (Eqn. 4.2 and Eqn. 4.4) and directed vs. undirected representation of a KG, it is natural to ask which of these combinations should be used for fact checking. We determine this by making an empirical evaluation using DBpedia on a task of inferring the party affiliation of the 112'th U.S. Congress members, based on a set of ideologies.

We characterize each congress member (445 House members, 100 Senators) in terms of ideologies such as Socialism, Conservatism, Nationalism and Nazism. See Section A.2 of Appendix A for a complete list. We create a feature matrix $\mathcal{F}_{\text{tc}}$ whose rows represent the members, columns represent the ideologies as features, and entries represent truth values as computed by a specific combination of path evaluation measure and KG representation for KL. Figure 4.6 shows a subgraph

of the KG consisting of paths connecting the members and various ideologies. The high degree of polarization seen in this figure is consistent with insights from blogs (Adamic and Glance, 2005) and social media (Conover et al., 2011).

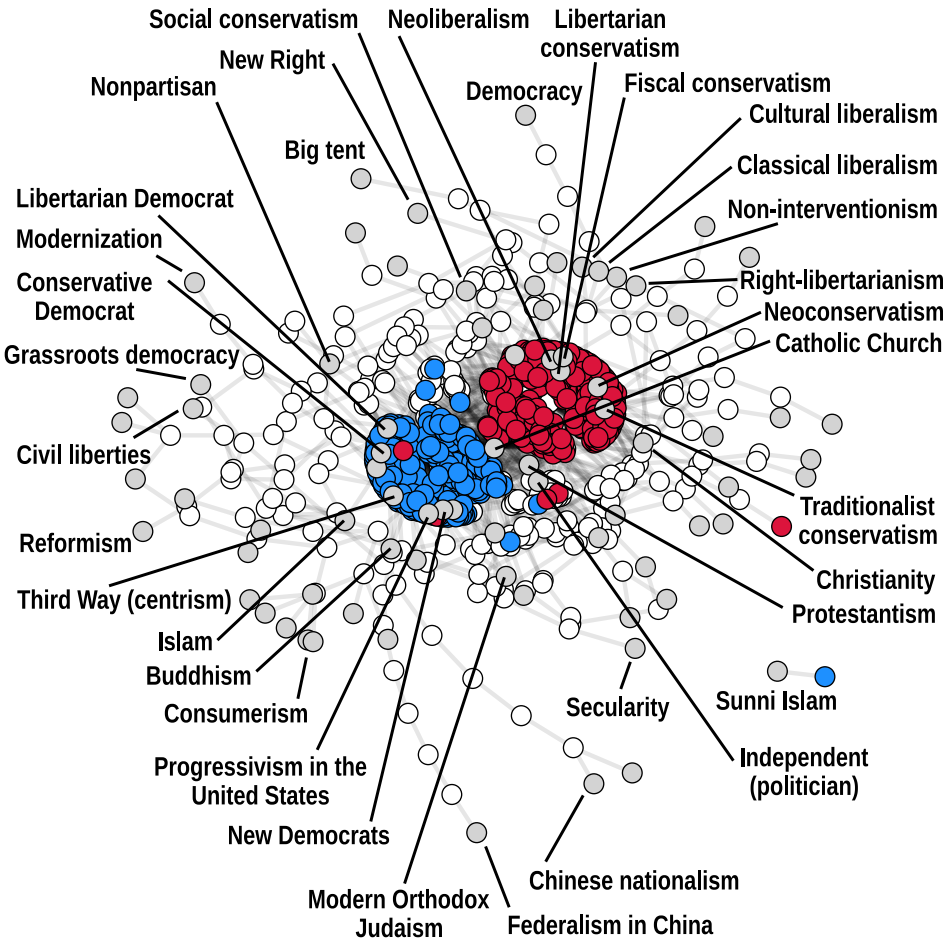


Figure 4.6: A network of U.S. Congress members and a set of ideologies from DBpedia. The nodes are arranged using a force-directed layout (Kamada and Kawai, 1989) to minimize the distance between nodes in proportion to the truth value assigned by KL using metric closure (Eqn. 4.3) and undirected KG. For clarity, only paths with edges having a truth value in top 1% of the values are shown. Red and blue nodes correspond to the members, gray nodes to the ideologies, and white nodes to other entities in the KG.

Using the feature matrix, we want to learn a model to predict the probability that "x is a member of y" given the truth values for the ideologies. Here x is a congress member and y is a political party, namely, Democratic or Republican. We employ off-the-shelf classifiers namely, Random Forest (RF) and k-Nearest Neighbor (k-NN) to predict this probability, and evaluate their performance

(and hence that of each combination) using the area under ROC curve (Section 2.5.2).

Table 4.1 shows the performance of these classifiers on both the data sets, one for House and another for Senate, using each combination. As seen, metric closure on an undirected KG results in superior performance; hence we consider Eqn. 4.3 as the *calibrated* model. We use metric closure and undirected KG in all validation tasks going forward.

Table 4.1: Transitive closure calibration. Performance by AUROC of two Random Forest and k-Nearest Neighbor classifiers on the ideology-based party classification task.

		Directed		**Undirected**	
		k-NN	RF	k-NN	RF
Metric	House	96	99	97	99
	Senate	70	100	96	100
Ultra-metric	House	56	57	53	57
	Senate	49	39	70	61

How does the calibrated model compare to the state of the art in political classification, DW-NOMINATE by Poole and Rosenthal (2007)? DW-NOMINATE analyzes legislative roll-call voting patterns and creates a two-dimensional projection of the data in which one of the axes is interpreted as a liberal-conservative scale. To answer the question, we compare the probabilities predicted by Random Forest trained on truth values obtained using the calibrated model, to scores derived from DW-NOMINATE. Figure 4.7 shows this comparison, clearly indicating that the truth values from our calibrated model hold the power to correctly segregate the two known groups.

4.5 Value of Indirect Paths

It is worth asking how much of the performance of KL is contributed by indirect paths, as opposed to direct edges? To investigate the role of indirect connections, we follow the same approach as outlined above, and compare the performance of two classifiers – one based on both direct and indirect connections, and another only based on direct connections.

To this end, we return to the political classification task, and build two classifiers based on the following feature matrices: (1) $\mathcal{F}_{\text{tc}}$ obtained using the calibrated model that uses both direct as

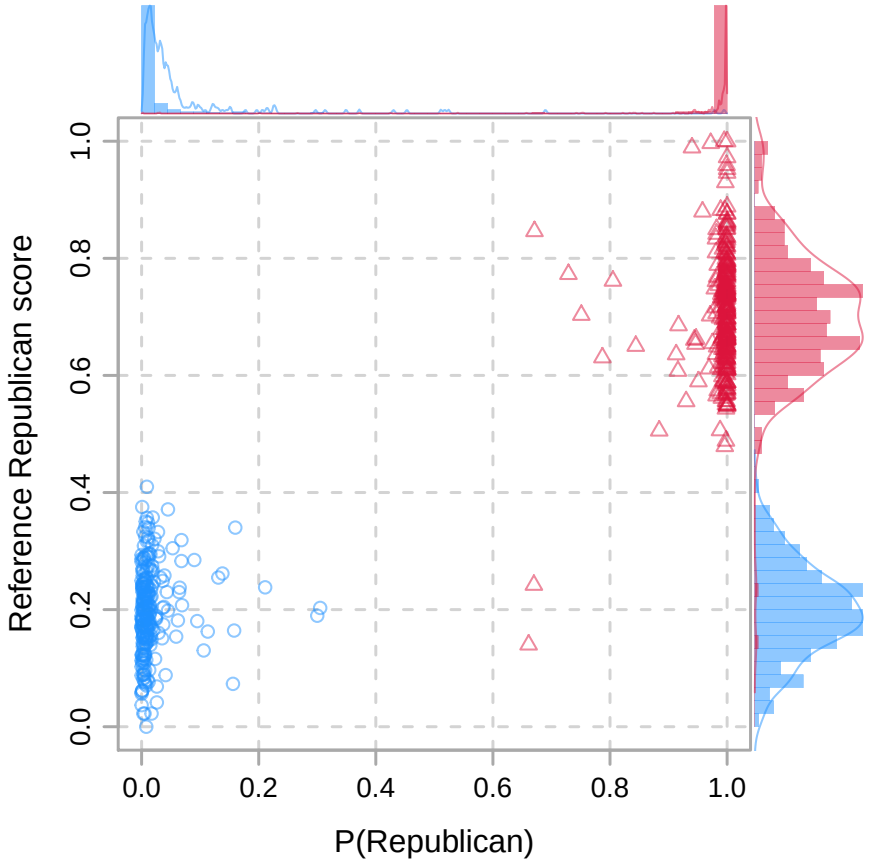


Figure 4.7: Comparison of KL performance on political classification task. The x-axis shows the party label probability given by Random Forest (calibrated model), and y-axis shows the reference score as derived from DW-NOMINATE. Red triangles are members affiliated to the Republican party and blue circles to the Democratic party. Histograms and density estimates are shown on the top and right axes, color-coded by actual affiliation.

well as indirect connections, and (2) $\mathcal{F}_{\text{b}}$ obtained in a similar manner, however using the direct connections, i.e., information available only in the Wikipedia articles' infoboxes. We again learn Random Forest and k-Nearest Neighbor classifiers using 10-fold cross-validation to compare the two scenarios.

Results of their performance in terms of AUROC and F-score is shown in Table 4.2. Notice that the classifiers based on direct connections are only marginally better than random. This confirms that the improvement in classifiers based on $\mathcal{F}_{\text{tc}}$ is mainly due to the set of indirect connections in KG.

Table 4.2: Performance by F-score and AUROC of Random Forest and k-Nearest Neighbor classifiers on political classification task.

			Infoboxes $\mathcal{F}_{\mathrm{b}}$	**Metric closure $\mathcal{F}_{\mathrm{tc}}$**
House	RF	F-score	0.54	0.99
		AUROC	0.66	1.00
	k-NN	F-score	0.68	0.90
		AUROC	0.54	0.97
Senate	RF	F-score	0.66	0.99
		AUROC	0.46	1.00
	k-NN	F-score	0.62	0.91
		AUROC	0.54	0.96

4.6 Checking Factual Statements

In this section, we evaluate fact-checking performance of Knowledge Linker and Relational Knowledge Linker on a number of test datasets pertaining to a variety of facts from entertainment, geography, sports, and more. Note that in practice, KL and KL-REL can be computed by Dijkstra's algorithm in $O(|E| \log_2 |V|)$ time using a binary heap implementation (Section 2.4.2).

4.6.1 Knowledge Graph, Datasets and Evaluation Metric

We continue to use DBpedia as the KG. Our version includes information from its ontology, instance-types and mapping-based properties; and has the following statistics: $|V|$ = 6M nodes, $|E|$ = 24M triples, and R = 663 relations.

Table 4.3 summarizes the data sets used in our evaluation. The data sets are categorized into two classes. The first class includes *synthetic* corpora that have been created by us and others, and contains *a priori* known mix of true and false triples. The false triples were created using local-closed world assumption (LCWA, Section 2.5.1); we provide additional details below. Datasets marked with an asterisk were first used in prior work (Shi and Weninger, 2016). We report the average number of facts per subject in the last column. A complete list of all the true triples in the synthetic datasets can be found in Appendix A.

Table 4.3: Summary of synthetic data sets used in the evaluation.

Dataset	Example Fact	True/Total	Facts/Subj.
	Synthetic		
NYT-Bestseller*	Flash Boys, author, M. Lewis	93/558	7.5
NBA-Team	K. Bryant, team, LA Lakers	41/164	9
Oscars	Gravity, director, A. Cuarón	78/4680	13
CEO*	Best Buy, keyPerson, H. Joly	201/1208	107
US War*	Siege of Corinth, battle, Henry Halleck	126/710	150
US-V. President*	B. Obama, vicePresident, J. Biden	47/274	169
FLOTUS	B. Obama, spouse, M. Obama	16/256	298
US-Capital #2*	Alabama, capital, Montgomery	50/300	4214
World-Capitals	Japan, capital, Tokyo	187/34969	NA
	Real-World		
GREC-Birthplace	D. Snow, birthPlace, Windermere, CA	273/1092	8
GREC-Deathplace	N. Tate, deathPlace, Southwark	126/504	8
GREC-Education	J. Warga, education, Bach. of Science	466/1861	9
GREC-Institution	A. Mirsky, almaMater, Harvard College	1546/6184	11
WSDM-Nationality	A. Einstein, nationality, Germany	50/200	97
WSDM-Profession	A. Sandler, profession, Comedian	110/440	220

The second class includes several *real-world* data sets. Some of these are derived from the Google Relational Extraction Corpora (GREC),[1] and contain information about birth place, death place, alma mater, and education degree of notable people. Two more data sets about professions and nationalities are derived from a corpus published during the WSDM Cup 2017 Triple Scoring Challenge.[2] The ground truth in both GREC and WSDM Cup corpora was obtained using crowd-sourcing. In the GREC, each triple was judged by five human raters. We use only triples whose rating was unanimous, i.e., either all true or all false. All triples in the WSDM Cup corpus were scored by seven raters, and represent true facts, by design. We consider only true triples with a unanimous score, and we generate false facts by randomly drawing from professions and nationalities that individuals are not known to hold, which also amounts to making a LCWA.

We evaluate both our fact checkers, KL and KL-REL, for their ability to discriminate true

[1] https://research.googleblog.com/2013/04/50000-lessons-on-how-to-read-relation.html
[2] http://www.wsdm-cup-2017.org/triple-scoring.html

statements of fact from false ones. To this end, we visualize the performance of both algorithms using a Receiver Operating Characteristic curve (ROC), and use the area under this curve (AUROC) as the evaluation metric. We also report the area under Precision-Recall curve (AUPR) for comparison. For a triple that is observed in the KG, we first remove its corresponding edge in G, and then perform fact checking. For computing relational similarity using the personalized PageRank, we use $\theta = 0.85$ and 1000 iterations to estimate the PageRank vector.

As mentioned above, for each dataset, the false triples were created using the local-closed world assumption (LCWA). This assumption is certainly valid for functional relations such as *bornIn*, however, for non-functional relations such as *profession* or *director*, the LCWA restricts the space of false triples by constraining the subject and predicate to be same as that of the true triple, and samples the object entity from the set of objects in the same relation. This implies that `(Jurassic Park, director, J. P. Dutta)` can serve as a valid false triple for the true triple `(Jurassic Park, director, Steven Spielberg)`. However, since Steven Spielberg and J. P. Dutta work in different film industries (Spielberg works in Hollywood, whereas Dutta works in Bollywood), random sampling from the set of objects may not be the optimal strategy for a challenging evaluation. Therefore, we restrict the set of false triples even further by only considering objects from the same dataset, e.g., all directors from Hollywood. Accordingly, a "better" valid false triple would be `(Jurassic Park, director, Alfonso Cuarón)`. It is possible that a true but unobserved triple such as `(Jurassic Park, director, Joe Johnston)` could be mistakenly sampled as a false triple in the sampling process. However, Dong et al. (2014) show empirically that LCWA serves well in practice, generating a plausible set of false triples.

4.6.2 Validation

Evaluation of KL

Figure 4.8 shows the results of KL as confusion matrices, on four datasets namely Oscars, FLOTUS, US-Capital #2, and World-Capitals, whose triples were created by a combination of all its subjects

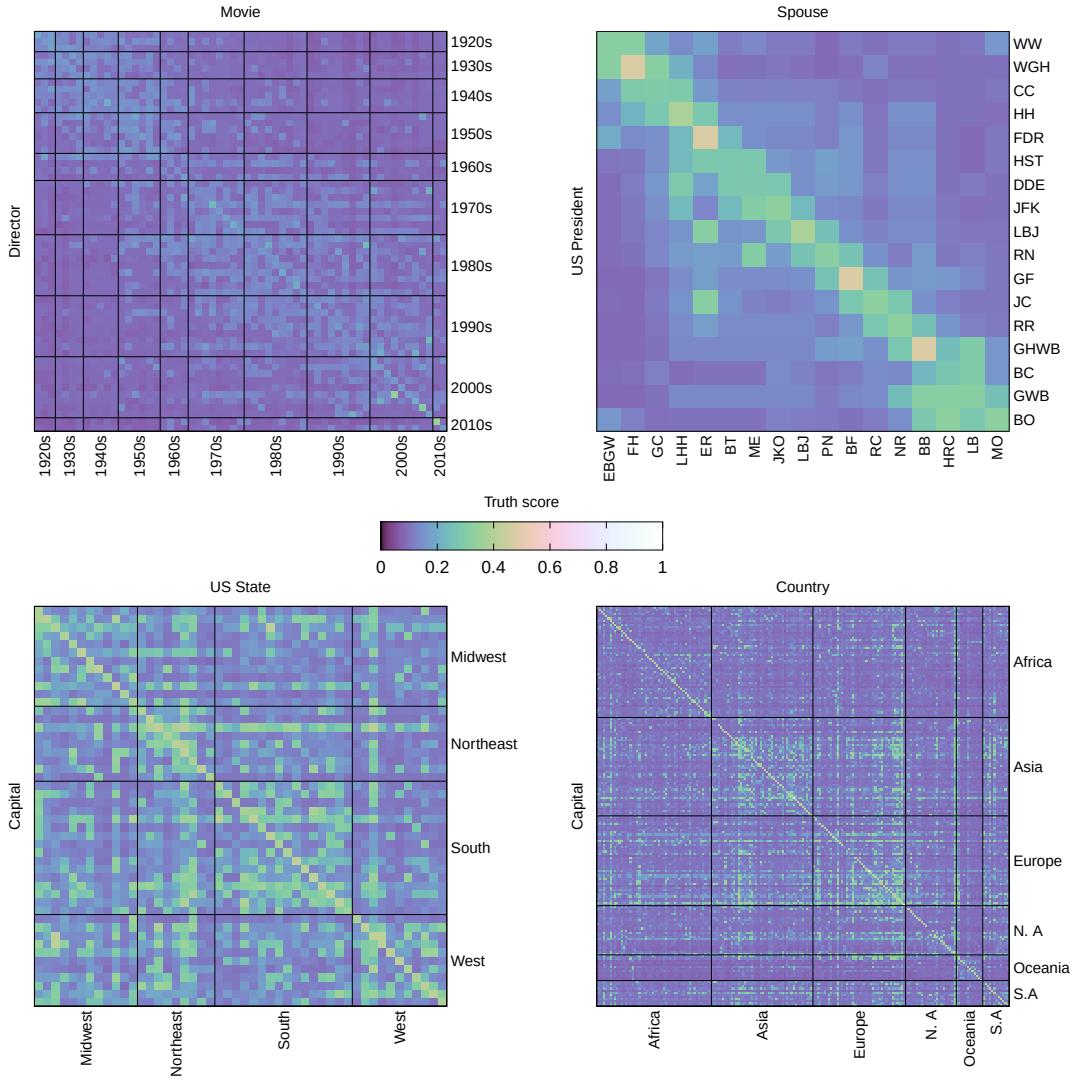


Figure 4.8: For each data set, the rows and columns represent subjects and objects respectively. The diagonals represent true statements. Higher truth values are mapped to colors of increasing intensity. (a) Oscars, (b) FLOTUS, (c) U.S. States and their capitals, grouped by U.S. Census Bureau-designated regions, and (d) World countries and their capitals, grouped by continent.

and objects. The bright spots along the diagonal indicate the ability of KL to correctly rank the true triples ahead of false ones. This is confirmed by their high AUROC scores of 98.59%, 97.56%, 100% and 95% respectively.

Evaluation of Relational Similarity Measures

Which measure of relational similarity among cosine similarity, pointwise mutual information (PMI), and personalized PageRank (PPR) should be used for the fact-checking task? Figure 4.9 shows a comparison of the performance by AUROC using the three measures over the datasets described in Table 4.3. Clearly, KL-REL using cosine similarity appears to perform better than the other two measures across all data sets as well as both synthetic and real subsets.

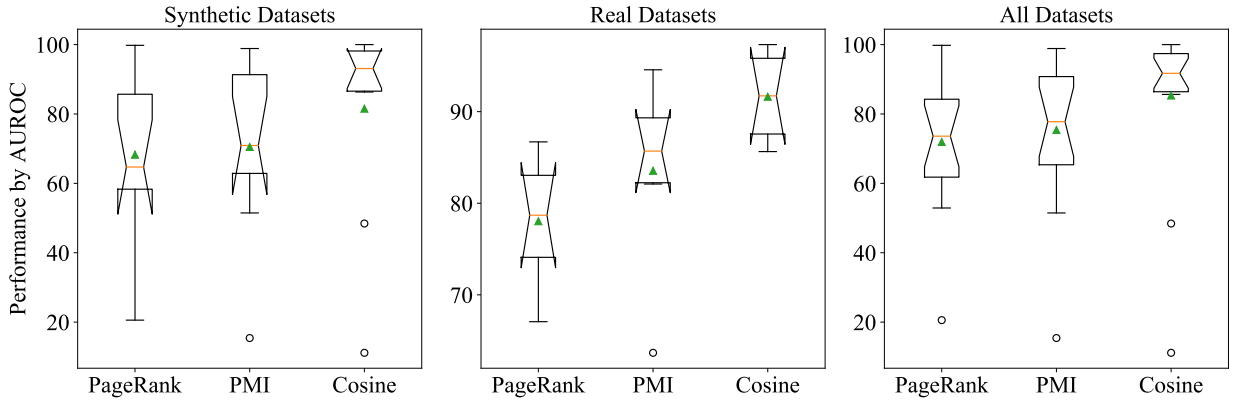


Figure 4.9: A comparison of performance (AUROC) of KL-REL on synthetic, real and all datasets using cosine similarity, PMI and personalized PageRank for measuring relational similarity. The orange line and green triangle represent the median and mean respectively.

To confirm, we perform the Nemenyi statistical test (Nemenyi, 1963) that tests the null hypothesis that there is no difference in the average AUROC performance of the three methods. The test reveals that the higher average performance based on cosine similarity is statistically significant than that based on PMI (p-value = 0.0075) and PPR (p-value = 0.0130) at 5% significance level. Moreover, there is no conclusive evidence that PMI and PPR are statistically different (p-value = 0.9829). We report all results going further using the cosine similarity measure since that works better on the fact-checking task.

Comparison between KL and KL-REL

Figure 4.10 and Figure 4.11 show the performance by ROC curve of KL and KL-REL on the synthetic and real datasets respectively. Table 4.4 shows the AUROC and AUPR of KL and KL-REL

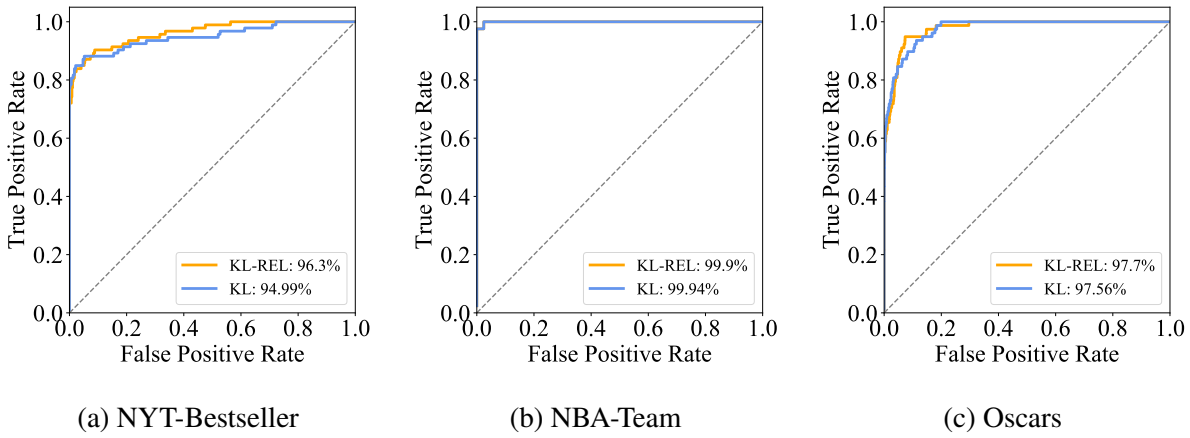
Table 4.4: Fact-checking performance.

	Dataset	AUROC		AUPR	
		KL	KL-REL	KL	KL-REL
Synthetic	NYT-Bestseller	94.99	**96.32**	91.23	**91.61**
	NBA-Team	**99.94**	**99.94**	**99.83**	**99.83**
	Oscars	97.56	**97.67**	**69.95**	67.86
	CEO	89.77	**89.88**	63.43	**76.11**
	US War	63.55	**86.34**	47.85	**71.79**
	US-V. President	74.62	**87.29**	63.97	**81.97**
	FLOTUS	**98.59**	98.32	**77.01**	66.96
	US-Capital #2	99.42	**100.00**	97.99	**100.00**
Real-World	GREC-Birthplace	92.10	**92.54**	84.14	**84.96**
	GREC-Deathplace	90.49	**90.91**	82.66	**83.19**
	GREC-Education	62.32	**86.44**	35.34	**69.18**
	GREC-Institution	**87.61**	85.64	**80.86**	76.29
	WSDM-Nationality	96.05	**96.92**	85.41	**91.86**
	WSDM-Profession	91.36	**97.32**	77.37	**92.74**
Average (Std. Error)		88.45 (3.4)	**93.25 (1.4)**	75.50 (4.9)	**82.45 (3.1)**

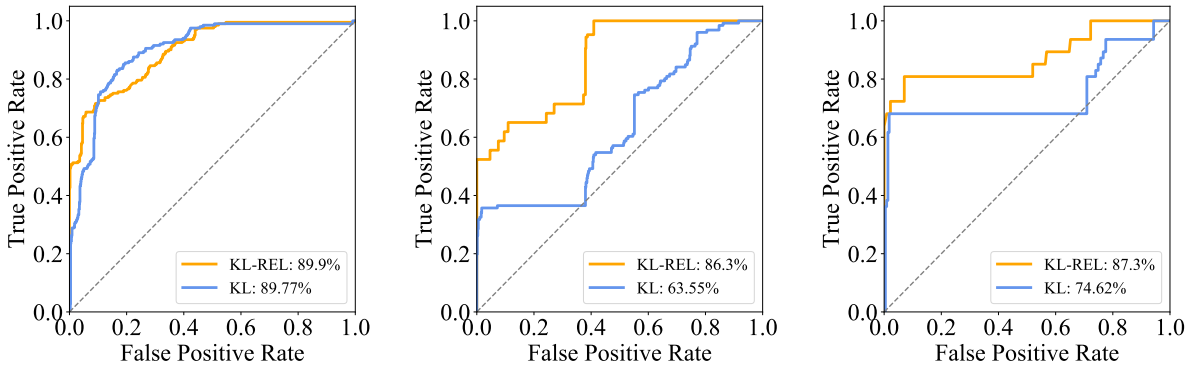
on the same. As we can observe, for most data sets, KL-REL offers better performance than KL per both metrics, which can be attributed to its inclusion of predicate semantics while assigning a truth score. The average difference in AUROC between KL and KL-REL taken across all data sets (including these two) is statistically significant per a Wilcoxon signed-rank test (Demšar, 2006), resulting in a p-value of 0.019. Thus, we confirm that KL-REL is superior to KL on average.

Interpretable Results

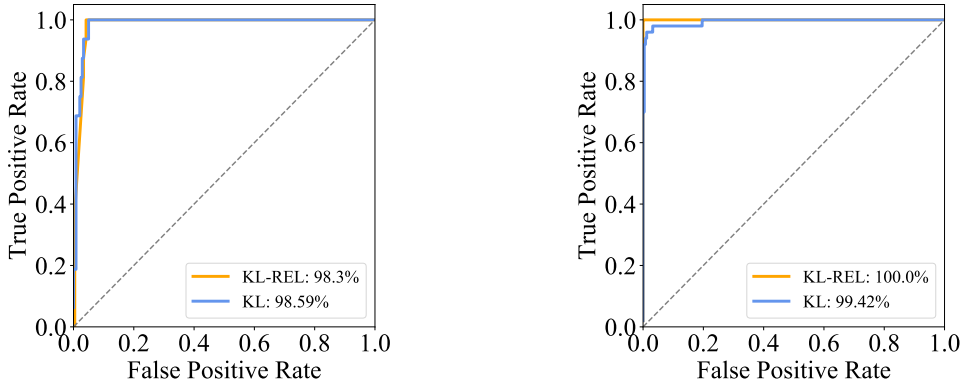
One advantage of KL and KL-REL is that they are very interpretable. Both models return a path for a triple, which can serve as an explanation to support or refute its corresponding claim. Table 4.5 shows example paths returned by KL-REL for a few true triples which also received a high score.



(a) NYT-Bestseller (b) NBA-Team (c) Oscars



(d) CEO (e) US-War (f) US-V.President

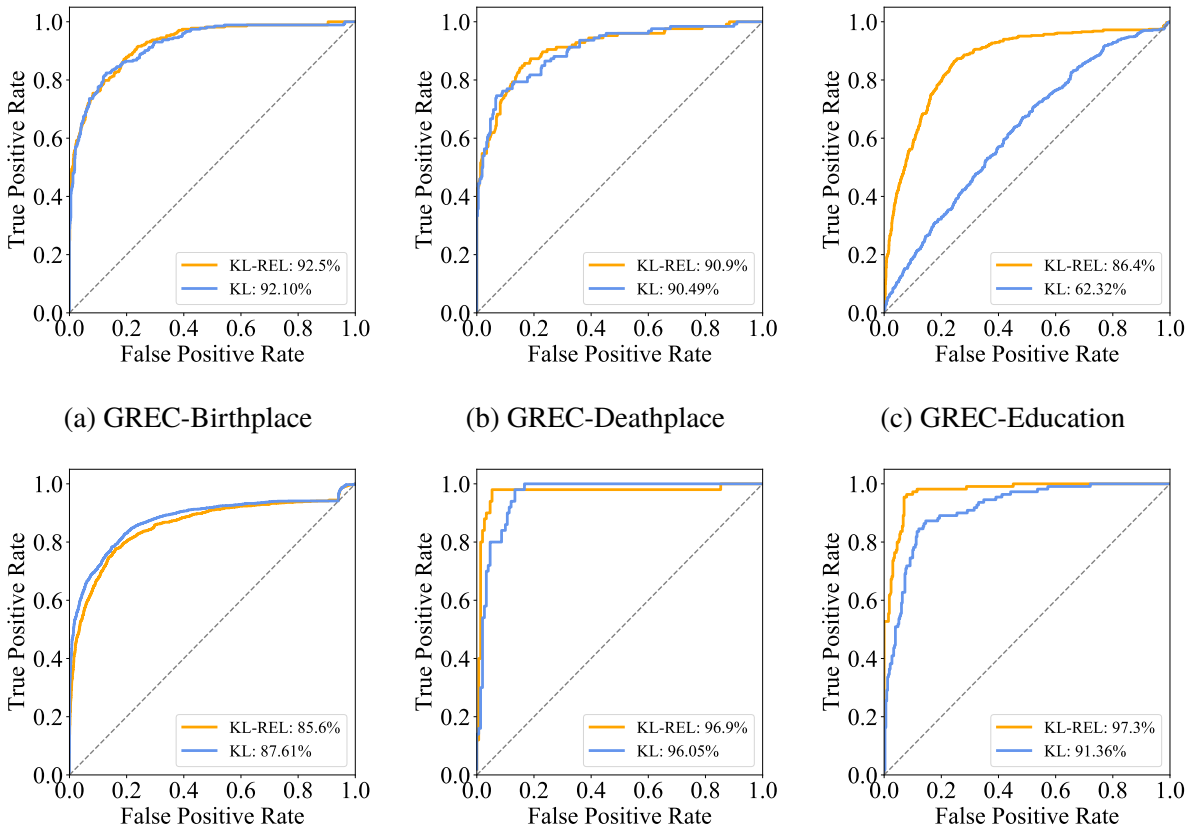


(g) FLOTUS (h) US-Capital #2

Figure 4.10: A visual comparison of the performance (ROC curve) of KL and KL-REL synthetic datasets. Number in the inset represents the area under the ROC curve.

4.7 Connection to Prior Work

We find a few interesting connections between KL/KL-REL and existing work.



(a) GREC-Birthplace (b) GREC-Deathplace (c) GREC-Education

(d) GREC-Institution (e) WSDM-Nationality (f) WSDM-Profession

Figure 4.11: A visual comparison of the performance (ROC curve) of KL and KL-REL real datasets. Number in the inset represents the area under the ROC curve.

The idea of using length of a path to assess information centrality of a node was first developed by Stephenson and Zelen (1989). There, the *information of a path* was defined as the inverse of its length, and paths between a pair of nodes were weighted based on their information. Longer paths thus received smaller weights. In KL and KL-REL, the specificity of a path precisely captures the idea of information of a path, except that the path length is now defined in terms of generality, instead of its traditional definition.

In the context of a folksonomy, Markines and Menczer (2009) proposed a measure called *maximum information path* (MIP) to effectively assess similarity between a pair of tags or resources. MIP is based on the idea of Shannon's information content of objects (tags or resources), and generalizes the traditional notion of shortest path. The authors find that a MIP for two objects passes

Table 4.5: Path returned by KL-REL.

True Fact	Path
Apple Inc., keyPerson, Tim Cook	Apple Inc. $\xrightarrow{\text{parentCompanyOf}}$ Apple Store $\xrightarrow{\text{keyPerson}}$ Tim Cook
Reggie Miller, team, Indiana Pacers	Reggie Miller $\xrightarrow{\text{draftTeam}}$ Indiana Pacers
Steve Jobs (book), author, Walter Isaacson	Steve Jobs (book) $\xrightarrow{\text{publisher}}$ Simon & Schuster $\xrightarrow{\text{publisherOf}}$ The Innovators: How a Group of Inventors, Hackers, Geniuses, and Geeks Created the Digital Revolution $\xrightarrow{\text{author}}$ Walter Isaacson
Washington, capital, Olympia, WA	Washington $\xrightarrow{\text{jurisdictionOf}}$ Washington State Liquor and Cannabis Board $\xrightarrow{\text{headquarter}}$ Olympia, WA
Dougie Vipond, birthPlace, Elderslie	Dougie Vipond $\xrightarrow{\text{birthPlace}}$ Scotland $\xrightarrow{\text{countryOf}}$ Elderslie

through the most specific object. This is similar to how KL and KL-REL assess truthfulness of a triple by finding the most specific path between its node pair.

PRA (Lao and Cohen, 2010b) and PredPath (Shi and Weninger, 2016) (also discussed in Section 3.4.1) rank relational paths (i.e., sequences of edge types or bodies of Horn clauses) based on their weight in a linear classifier. The top paths serve as evidence for their decisions. Similar to PRA and PredPath, KL and KL-REL also return a relational path, which serves as an explanation for the score. However, unlike the former two, KL and KL-REL only return a single best path.

4.8 Summary

Fact checking is an important activity to prevent dubious claims from spreading on online social platforms. In this chapter, we saw that we can make progress toward this daunting task by developing automatic fact checkers that rely on large structured repositories of knowledge. In particular, we saw two network theoretic approaches, namely, Knowledge Linker and Relational Knowledge Linker, that assign a truth value to a triple by simply solving a shortest path problem that exploits the implicit evidence of generality of concept nodes. Both the models offer some interpretability, which can

assist human fact checkers in their analysis of claims.

In pursuing both the approaches, we defined a new measure of path length based on the generality of intermediate nodes, that extends its traditional definition based on number of edges (or hops). We concluded that indirect connections between a concept node pair are more valuable for veracity assessment than direct connections. Evaluation on a range of test cases showed that KL-REL is more effective than KL, since it takes the predicate semantics into account besides path specificity.

Chapter 5

Fact Checking by Network Flow

In this chapter, we explore the possibility of using multiple paths between a node pair in assessing a triple's truthfulness. To this end, we introduce a network-flow based approach called *Knowledge Stream* (KS) that takes into account the broader structural and semantic context of a triple while assigning a truth value. We show that this approach is effective for fact checking on datasets previously considered, and is novel in its ability to automatically discover useful patterns and contextual facts pertaining to the claim triple. Before moving on, we strongly encourage the reader to review a few graph theory and network flow theory concepts explained in Chapter 2.

5.1 Motivation

We view a knowledge graph (KG) as a flow network, and knowledge as a fluid, abstract commodity that can flow over this network. Our interest lies in finding a set of paths between a triple's subject and object, which can explain the triple's truthfulness based on their ability to carry flow. We call this useful set of paths a "stream" of knowledge.

One such stream uncovered for a claim triple `(David and Goliath (book), author, Malcolm Gladwell)` is shown in Figure 5.1. Note that some paths give more evidence in support of the claim than others, as indicated by their relative higher width. For example, the fact that Malcolm Gladwell is author of the book *What the Dog Saw* that followed *David and Goliath*, is a stronger form of evidence than the fact that another book authored by Gladwell, *The Tipping Point*, was published by the same company (*Little, Brown and Company*) as *David and Goliath*. This situation is correctly captured by Knowledge Stream, which assigns a large flow to the former path than the latter.

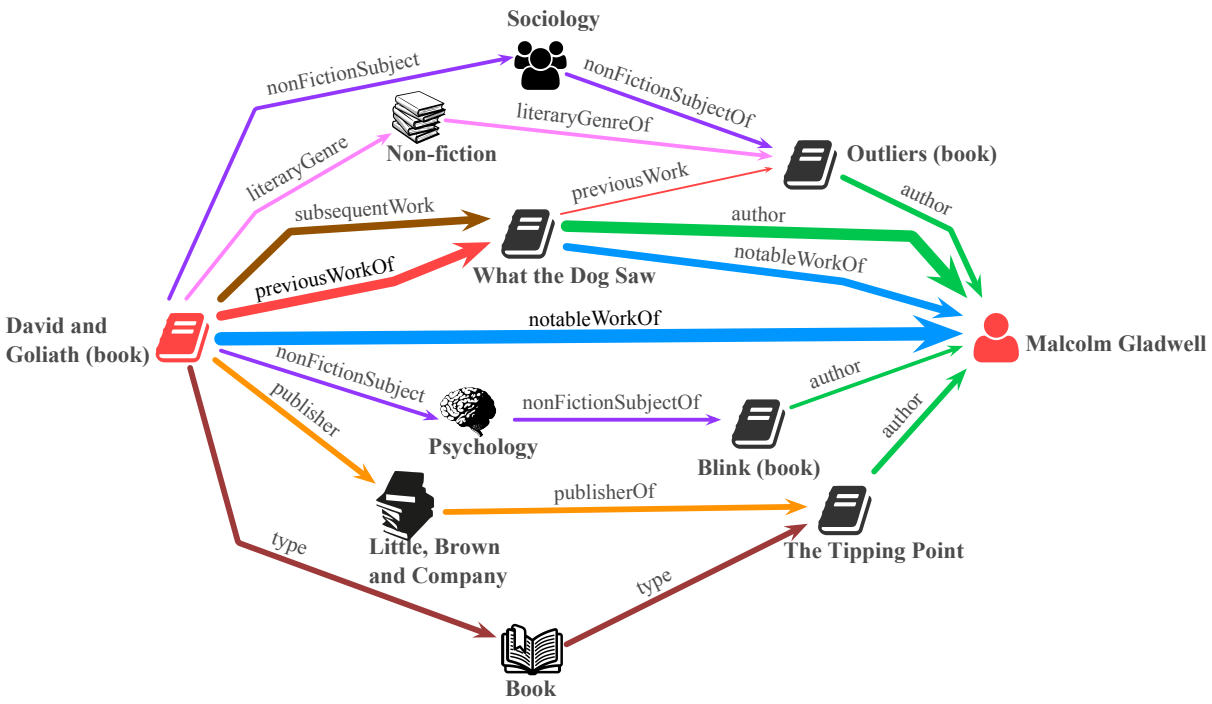


Figure 5.1: The best paths identified by Knowledge Stream for the triple `(David and Goliath (book), author, Malcolm Gladwell)`. The width of an edge is roughly proportional to the flow of knowledge through it.

The main motivations behind treating the KG as a flow network include:

1. **Use of multiple paths**: multiple paths between a node pair represent distinct high-level facts, and may provide greater semantic context than that provided by a single path,

2. **Reuse of edges**: edges participating in diverse overlapping chains of relationships can be reused by sending additional flow, and

3. **Efficient path search**: capacities of edges limit the number of paths in which they can participate, thereby constraining the path search space.

5.2 Fact Checking as a Minimum Cost Maximum Flow Problem

In Knowledge Stream, we model fact checking of a triple (s, p, o) as a problem of finding an optimal way to transfer knowledge from s to o across the network under two sets of constraints. These constraints depend on the structure of the KG as well as the triple under evaluation.

The first set of constraints relates to the edges, and dictates that the amount of flow that can be pushed across an edge $e = (v_i, r_m, v_j) \in E$ is bounded. We call them the *edge capacity constraints*. We take the lower bound on the flow to be zero, and the upper bound to be proportional to the relational similarity between r_m and p, the relation we are trying to check. We have seen how this similarity can be assessed in Section 4.3. Intuitively, the more semantically relevant r_m is to p, the higher this bound ought to be. For example, if we are to ascertain whether Malcolm Gladwell is the author of *David and Goliath*, facts about his birth place or nationality may be less pertinent than those about his previous works in non-fiction genre. We therefore define the upper bound or *capacity* of e with respect to (s, p, o) as

$$\mathcal{U}_{s,p,o}(e) = \frac{u(r_m, p)}{1 + \log k(v_j)}, \tag{5.1}$$

where we have multiplied the relational similarity $u(r_m, p)$ with the specificity of v_j. This choice of taking the product of the two quantities is only by design, and alternative ways could be explored.

The second set of constraints relates to the nodes, and states that, except for the nodes corresponding to subject s and object o, the amount of flow entering a node must equal that leaving the node. We call these the *node conservation constraints*. We associate with s (respectively o) a fixed *supply (demand)* of knowledge γ, equivalent to the maximum feasible flow between them. All other nodes do not have any supply or demand.

We associate with each edge $e = (v_i, r_m, v_j) \in E$ a *cost* $c(e)$ that denotes the cost per unit flow on that edge. In network flow problems, one typically aims to minimize such costs on edges. For example, in a transportation network, if the costs correspond to distances between cities, one seeks to find a least-cost way to transport goods. In the context of KG, as in Knowledge Linker, we penalize edges based on the generality of their incident nodes. Thus, we represent $c(e)$ proportional to the generality of v_j, i.e., $c(e) = \log k(v_j)$ (Eqn. 4.2). Since we use an undirected KG, we treat each edge as a two-way street and replace it by two directed edges, (v_i, r_m, v_j) and (v_j, r_m, v_i). Note that these edges may differ in their cost, depending on the generality of their incident nodes (v_j and v_i

respectively).

Having defined the constraints relating the nodes and edges of the KG, we solve a *minimum cost maximum flow* problem. The flow assignment to edges of the KG is a non-negative real-valued mapping $f : E \rightarrow \mathbb{R}^+$, that maximizes the total flow γ from s to o, while minimizing the total cost $\sum_{e \in E} c(e) f(e)$ subject to the edge capacity constraints:

$$0 \leq f(e) \leq \mathcal{U}_{s,p,o}(e)$$

and the node conservation constraints:

$$b(v) = \begin{cases} \gamma & v = s \\ -\gamma & v = o \\ 0 & \text{otherwise} \end{cases}$$

where $b(v_i) = \sum_{v_j \in V} f(v_i, v_j) - \sum_{v_j \in V} f(v_j, v_i)$ is the net flow emanating from node v_i.

We are interested in finding the set of paths along which the maximum flow γ is pushed from s to o. In practice, we solve the minimum cost maximum flow problem using an algorithm that computes such set of paths. We denote this set of paths the *stream of knowledge* $\mathcal{P}_{s,p,o}$. Each path in the stream carries knowledge at its full capacity. The maximum knowledge a path $P_{s,p,o}$ can carry is the minimum of the capacities of its edges, also called its *bottleneck* $\beta(P_{s,p,o})$. It can be shown that the maximum flow is the sum of the bottlenecks of the paths that are part of the stream:

$$\gamma = \sum_{P_{s,p,o} \in \mathcal{P}_{s,p,o}} \beta(P_{s,p,o}). \tag{5.2}$$

Having determined the maximum flow and knowing the exact contribution of each individual path in a stream, we now look at how to use the stream for fact checking (s, p, o): The flow through a path captures the relational similarity and specificity of its bottleneck, as well as the specificity of the intermediate nodes. However, long chains of specific relationships could lead us astray. Therefore,

Knowledge Stream should use a path's flow contribution proportional to its specificity $\mathcal{S}(P_{s,p,o})$ (Eqn. 4.2). We say that the net flow $\mathcal{W}(P_{s,p,o})$ in a path $P_{s,p,o}$ is the product of its bottleneck $\beta(P_{s,p,o})$ and specificity $\mathcal{S}(P_{s,p,o})$:

$$\mathcal{W}(P_{s,p,o}) = \beta(P_{s,p,o}) \cdot \mathcal{S}(P_{s,p,o}). \tag{5.3}$$

Fact checking a triple (s, p, o) then reduces to computing a *truth value* $\tau^{\text{KS}}(s, p, o)$ as the sum of the net flow over all paths in the stream:

$$\begin{aligned} \tau^{\text{KS}}(s, p, o) &= \sum_{P_{s,p,o} \in \mathcal{P}_{s,p,o}} \mathcal{W}(P_{s,p,o}) \\ &= \sum_{P_{s,p,o} \in \mathcal{P}_{s,p,o}} \beta(P_{s,p,o}) \cdot \mathcal{S}(P_{s,p,o}). \end{aligned} \tag{5.4}$$

5.3 Computing Knowledge Stream

We now discuss how to solve our optimization problem and compute the truth value of a triple in practice. A well-known algorithm called Successive Shortest Path (SSP) provides a solution to the optimization problem and also returns the sequence of paths. The idea is to push the maximum flow γ from s to o by iteratively finding a shortest path in a residual network, along which we can push some flow. The *residual network* $G(f)$ of G w.r.t flow f has the same set of nodes V as G, but has two kinds of edges: (1) *forward edges* with some "leftover capacity" over which one can push additional flow, and (2) *backward edges* that represents edges already allocated, over which one can push reverse flow in order to undo flow in forward edges. At each step in the iteration we compute the bottleneck of the shortest path, given by

$$\beta(P_{s,p,o}) = \min \{x_e | e \in P_{s,p,o}\}, \tag{5.5}$$

where $x_e \leq \mathcal{U}_e$ represents the residual capacity of edge e in the residual network. Our extended version of SSP to compute the stream of knowledge and the truth value $\tau^{\text{KS}}(s, p, o)$ is shown in

Algorithm 1.

Algorithm 1 Knowledge Stream Algorithm

1: **procedure** KNOWLEDGESTREAM(G, s, p, o)
2: $\tau \leftarrow 0, \mathcal{P} \leftarrow \emptyset, f \leftarrow 0$
3: $\pi \leftarrow 0$
4: $c_{v_i,r_m,v_j} = \log(v_j), \forall (v_i, r_m, v_j) \in E$
5: $c^{\pi}_{v_i,r_m,v_j} = c_{v_i,r_m,v_j} - \pi(v_i) + \pi(v_j)$
6: $d \leftarrow$ compute shortest path distances from s to all other nodes in $G(f)$ w. r. t. c^{π}
7: $P \leftarrow$ a shortest path from s to o in $G(f)$
8: **while** P exists **do**
9: $\mathcal{P} \leftarrow \mathcal{P} \cup \{P\}$
10: $\pi \leftarrow \pi - d$
11: $\beta(P) \leftarrow \min \left\{ x_{v_i,r_m,v_j} \middle| (v_i, r_m, v_j) \in P \right\}$
12: Push $\beta(P)$ units of flow along P
13: $\mathcal{S}(P) \leftarrow \frac{1}{1+\sum_{i=2}^{n-1} \log k(v_i)}$ for $v_i \in P$
14: $\mathcal{W}(P) \leftarrow \beta(P) \cdot \mathcal{S}(P)$
15: $\tau \leftarrow \tau + \mathcal{W}(P)$
16: update $f, G(f)$ and reduced edge lengths c^{π}
17: $d \leftarrow$ compute shortest path distances from s to all other nodes in $G(f)$ w. r. t. c^{π}
18: $P \leftarrow$ a shortest path from s to o in $G(f)$
19: **end while**
20: **return** $\tau, \mathcal{P}$
21: **end procedure**

We associate a real number $\pi(v_i)$ (Line 3) with each node $v_i \in V$, called its *node potential*. A vector of such node potentials π serves two important purposes: (1) it allows us to keep track of the *reduced cost* c^{π} (Line 5) of an edge at each step of the algorithm, which makes successive path-finding efficient; and (2) it serves as an ingredient of the *reduced cost optimality conditions* that ensure the achievement of maximum flow upon termination. For a more detailed introduction of these terms and their role, we refer the reader to the classic introductory text on network flow by Ahuja et al. (1993, Ch. 9).

The complexity bounds for the SSP algorithm assume that all edge weights are integral, which does not hold for our capacities, since $\mathcal{U}_{s,p,o}(e) \in [0, 1]$. This is not a problem however, since capacities are rational numbers and can therefore be easily converted to integers. If the maximum flow γ is an integer, the Knowledge Stream algorithm takes at most γ iterations. Since each shortest path computation can be performed in $O(|E| \log |V|)$ time using Dijkstra's algorithm (Dijkstra,

1959) with a binary heap implementation, the overall complexity of the algorithm is $O(\gamma|E|\log|V|)$. In practice, γ is not an integer, and is computed by the algorithm; this makes Knowledge Stream a pseudo-polynomial time algorithm. In practice we find that the performance of our implementation differs widely across datasets; Figure 5.2 shows the average time taken for datasets described in Table 4.3. The black bars indicate the standard deviation. As seen, there is a large variability in the time taken within and across datasets. Nevertheless, we find that overall, KS gives an acceptable average of 356 seconds per triple on a laptop.

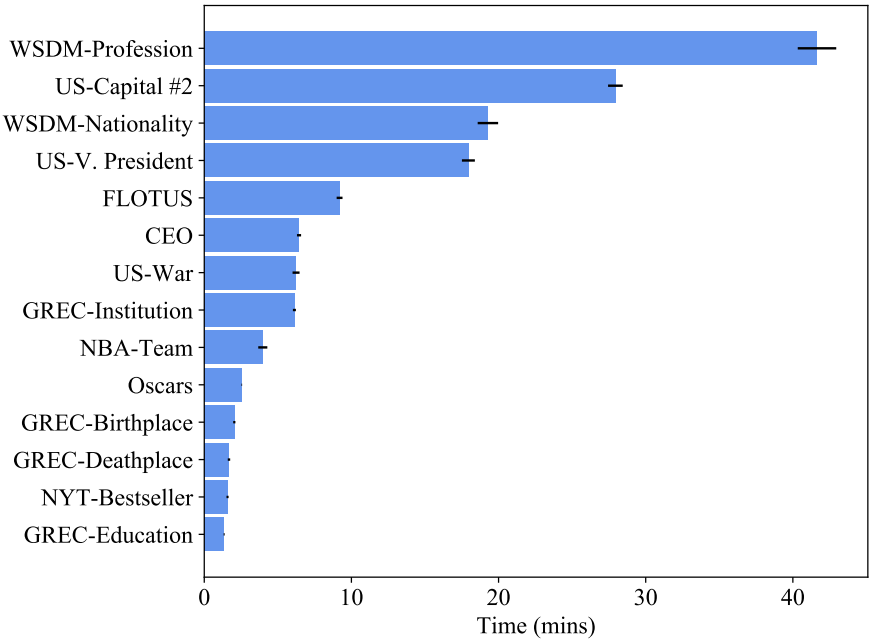


Figure 5.2: Average time taken by Knowledge Stream in minutes. The bars represent standard deviation of the datasets.

5.4 Checking Factual Statements

Knowledge Graph, Benchmark and Evaluation Metric

We evaluate Knowledge Stream using the same version of DBpedia and on same datasets as considered for evaluating KL and KL-REL, i.e., Table 4.3. We use information from its ontology, instance-types and mapping-based properties; and the resultant KG has following statistics: $|V|$ =

6M nodes, $|E|$ = 24M triples, and R = 663 relations. The data set World-Capitals is computationally expensive due to a large search space for each triple, so we exclude it from evaluation.

We compare all our algorithms (KL, KL-REL and KS) with each other, and to many existing algorithms applicable for fact checking. The existing algorithms include —

- six link prediction algorithms namely Adamic & Adar (Adamic and Adar, 2003), Jaccard coefficient (Liben-Nowell and Kleinberg, 2007), Degree Product (Shi and Weninger, 2016)), Katz (Katz, 1953), Path Entropy (labeled as PathEnt henceforth) (Xu et al., 2016b), and SimRank (Jeh and Widom, 2002); the first three use neighborhood of a node pair, whereas the rest three use the ensemble of paths,
- one algorithm for knowledge graph completion namely TransE by Bordes et al. (2013), and
- two algorithms designed for fact checking, namely PredPath by Shi and Weninger (2016) and PRA by Lao and Cohen (2010b)

For Katz, PathEnt, PRA and PredPath, we use up to 200 paths for each value of path length $l = 2, 3$. In case of Katz, we set the attenuation factor to 0.05 as suggested by Liben-Nowell and Kleinberg (2007). To scale SimRank to the size of DBpedia, we implemented the approximation based on random walks proposed by Kusumoto et al. (2014), and use the following values for its parameters: decay factor $c = 0.8$, number of terms $T = 50$, and number of random walks $R = 1000$. In the case of TransE, we create 100-dimensional embeddings using a margin of one and a learning rate of 0.01 for 1,000 epochs. We use the area under the Receiver Operating Characteristic curve (AUROC) to evaluate all algorithms, since it allows us to compare the accuracy across datasets with different ratios of true and false facts.

Validation

Table 5.1 and Table 5.2 give a comparison of fact-checking performance between our approaches and other algorithms on the synthetic and real-world datasets. We report average performance and standard error across datasets for each algorithm in the last column. Although statistical significance

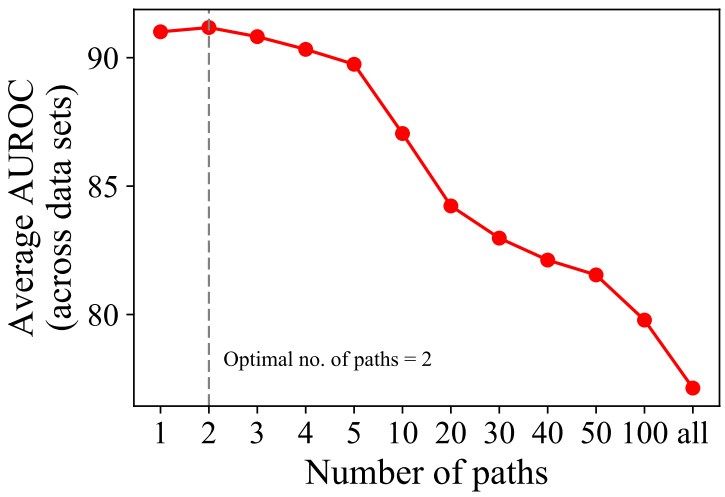


Figure 5.3: Average performance of Knowledge Stream across datasets as a function of the number of paths used in the stream.

tests do not reveal a clear overall winner, we make a few observations. KL-REL performs better than TransE and all link prediction algorithms. In fact, it outperforms all other algorithms on real-world datasets and has comparable performance to PredPath on synthetic data.

KS lags behind KL. A possible explanation could be that the extra signal provided by the additional paths found by KS may not always be beneficial. To shed more light into this issue, we analyzed the average performance as a function of the number of paths in the stream. Figure 5.3 shows this behavior, indicating that the overall optimum is attained when exactly two paths are considered. On the one hand, this confirms the value of considering multiple paths. On the other hand, this suggests that too many paths hinder performance, and thus the number of paths should be tuned.

Based on this insight, we include in our evaluation two variants of Knowledge Stream. KS-AVG uses the number of paths (two) resulting in the best performance on average. KS-CV uses cross-validation to tune the optimal number of paths for each dataset; this makes KS-CV a supervised approach. As we see from both tables, KS-AVG and KS-CV have a better performance on average than KS, and even better than KL-REL on synthetic datasets. This confirms our intuition of focusing only on a few paths in a stream.

Table 5.1: Fact-checking performance (AUROC) on synthetic data. Best scores for each dataset are shown in bold.

Method	NYT-Bestseller	NBA-Team	Oscars	CEO	US-War	US-V. President	FLOTUS	Capital #2	Avg. (S.E.)
KS	89.72	**99.96**	95.00	81.19	72.11	77.80	98.05	**100.00**	89.23 (3.9)
KS-AVG	91.95	99.01	98.13	80.96	**99.98**	**99.53**	99.09	99.76	96.05 (2.3)
KS-CV	93.63	99.29	97.72	80.52	**99.98**	99.47	99.27	99.28	96.14 (2.3)
KL-REL	96.32	99.94	97.67	**89.88**	86.34	87.29	98.32	**100.00**	94.47 (2.0)
KL	94.99	99.94	97.56	89.77	63.55	74.62	98.59	99.42	89.80 (4.8)
PredPath (Shi and Weninger, 2016)	**99.80**	92.31	**99.97**	88.67	99.51	94.40	**100.00**	99.68	**96.79** (1.6)
PRA (Lao and Cohen, 2010b)	96.24	91.26	99.54	87.73	99.96	50.00	60.48	98.88	85.51 (6.8)
TransE (Bordes et al., 2013)	80.99	56.71	82.66	82.68	53.22	72.50	84.82	85.31	74.86 (4.6)
Katz (Katz, 1953)	96.52	98.50	98.98	87.53	57.80	72.92	97.42	99.97	88.70 (5.5)
PathEnt (Xu et al., 2016b)	97.53	97.00	98.99	92.96	93.65	90.60	**100.00**	**100.00**	91.76 (3.5)
SimRank (Jeh and Widom, 2002)	88.36	97.74	51.47	80.83	50.03	65.78	91.85	99.50	73.15 (6.6)
Adamic & Adar (Adamic and Adar, 2003)	95.84	99.73	56.54	84.97	54.98	81.06	99.40	**100.00**	84.06 (6.7)
Jaccard (Liben-Nowell and Kleinberg, 2007)	92.64	99.42	53.35	78.74	49.68	70.79	97.89	**100.00**	80.31 (7.3)
Degree Product (Shi and Weninger, 2016)	56.52	53.21	54.42	49.17	64.08	49.55	50.00	52.10	53.63 (1.7)

Table 5.2: Fact-checking performance (AUROC) on real test datasets. Best scores for each dataset are shown in bold.

Method	GREC Birthplace	GREC Death-place	GREC Education	GREC Institution	WSDM National-ity	WSDM Profession	Avg. (S. E.)
KS	72.92	80.02	89.03	78.62	97.92	98.66	86.20 (4.4)
KS-AVG	81.38	83.58	75.46	81.31	93.37	92.93	84.67 (2.9)
KS-CV	82.28	82.57	75.23	81.33	94.20	95.84	85.24 (3.3)
KL-REL	**92.54**	**90.91**	86.44	85.64	96.92	97.32	**91.63** (2.0)
KL	92.10	90.49	62.32	**87.61**	96.05	91.36	86.65 (5.0)
PredPath (Shi and Weninger, 2016)	84.64	76.54	83.21	80.14	95.20	92.71	85.41 (2.9)
PRA (Lao and Cohen, 2010b)	74.34	75.58	70.51	63.95	83.87	50.00	69.71 (4.8)
TransE (Bordes et al., 2013)	54.88	56.47	66.32	44.99	77.09	82.91	63.78 (5.9)
Katz (Katz, 1953)	88.46	84.07	89.55	82.99	**99.23**	**98.84**	90.52 (2.9)
PathEnt (Xu et al., 2016b)	84.02	79.2	83.48	82.26	55.31	48.13	72.07 (6.5)
SimRank (Jeh and Widom, 2002)	86.91	85.48	67.60	72.92	89.39	92.27	82.43 (4.0)
Adamic & Adar (Adamic and Adar, 2003)	82.79	79.13	50.00	74.58	97.21	95.07	79.80 (7.0)
Jaccard (Liben-Nowell and Kleinberg, 2007)	80.39	75.99	49.95	69.88	95.93	90.01	77.02 (6.6)
Degree Product (Shi and Weninger, 2016)	52.82	50.86	**91.51**	64.56	84.38	86.36	71.75 (7.3)

We observe that KL-REL, KS, KS-AVG and KS-CV algorithms often outperform existing fact-checking methods (PredPath and PRA). We emphasize that KL, KL-REL and KS-AVG are purely unsupervised algorithms, whereas PredPath and PRA require supervision for both feature selection and model training.

Finally, link prediction algorithms (Adamic & Adar, Jaccard coefficient, Degree Product, and SimRank) tend to perform poorly. Katz is an exception in this category. On real-world datasets, its performance is comparable to that of KL-REL. However, KL, KL-REL and KS are computationally efficient compared to Katz. In case of KL and KL-REL this is because of its focus on a single path.

As for KS, it uses multiple paths and penalizes longer paths just like Katz, but is more efficient thanks to the capacity constraints.

5.5 Discovery of Relational Path Patterns

For each triple, the paths discovered by KS, KL-REL, KL, PRA, and PredPath, can be seen as the evidence used by the algorithms in deciding whether the fact is true. By pooling together evidence from many triples, we can discover data-driven patterns that define a relation, based on the prior knowledge in the KG.

It is natural to ask whether the patterns discovered by our methods (especially KS and KL-REL) conform to common-sense understanding of these relations. To do so, we perform the following simple exercise. For each relation, we define two sets A and B of all paths discovered from either true or false triples, respectively. We then rank the paths in decreasing order of their frequency of occurrence in the set difference $A - B$.

Table 5.3 shows a few top paths discovered by KS for a few relations. Clearly, they represent patterns that are highly relevant. This characteristic of KS has a wider applicability — with only a few true and false examples, these patterns can be discovered in an unsupervised fashion, and used either as a seed set of rules in information extraction projects, or as features for learning other concepts.

5.6 Surfacing Facts Relevant to a Claim

The workflow of a human fact checker begins by gathering facts that are relevant to a claim being checked. Possible sources are background information, interview transcripts, closed captions of debates, etc. See for example a book on fact checking by Borel (2016). We find that KS can assist in this task by identifying the general context of a triple. As an illustration, Figure 5.4 shows the set of most relevant facts (indicated by the paths) for the triple `(Berkshire Hathaway, keyPerson, Warren Buffett)`. The width of edges are roughly proportional to their net flow $\mathcal{W}(P_{s,p,o})$. See Figure 5.1 for another example. Notice the diversity in the set of facts that support

Table 5.3: Relational patterns discovered by Knowledge Stream.

Relation	Pattern	Frequency	Example
FLOTUS	(child, childOf)	34	J. F. Kennedy $\xrightarrow{\text{child}}$ Patrick Kennedy $\xrightarrow{\text{childOf}}$ Jacqueline Kennedy Onassis
	(parentOf, parent)	20	J. F. Kennedy $\xrightarrow{\text{parentOf}}$ Patrick Kennedy $\xrightarrow{\text{parent}}$ Jacqueline Kennedy Onassis
	(child, parent)	19	J. F. Kennedy $\xrightarrow{\text{child}}$ Patrick Kennedy $\xrightarrow{\text{parent}}$ Jacqueline Kennedy Onassis
	(predecessor, spouse, predecessorOf)	6	R. Reagan $\xrightarrow{\text{predecessor}}$ P. Brown $\xrightarrow{\text{spouse}}$ B. Brown $\xrightarrow{\text{predecessorOf}}$ N. Reagan
CEO	(parentCompanyOf, keyPerson)	32	News Corporation $\xrightarrow{\text{parentCompanyOf}}$ Sky TV plc $\xrightarrow{\text{keyPerson}}$ Rupert Murdoch
	(employerOf)	24	Twitter $\xrightarrow{\text{employerOf}}$ Dick Costolo
	(foundedBy)	24	Foxconn $\xrightarrow{\text{foundedBy}}$ Terry Gou
	(subsidiary, keyPerson)	20	Samsung $\xrightarrow{\text{subsidiary}}$ Samsung Electronics $\xrightarrow{\text{keyPerson}}$ Lee Kun-hee
US Capital #2	(deathPlaceOf, deathPlace)	491	Delaware $\xrightarrow{\text{deathPlaceOf}}$ Nathaniel B. Smithers $\xrightarrow{\text{deathPlace}}$ Dover, Delaware
	(part, isPartOf)	123	Delaware $\xrightarrow{\text{part}}$ Delaware Valley $\xrightarrow{\text{isPartOf}}$ Dover, Delaware
	(headquarterOf, location)	112	Kansas $\xrightarrow{\text{headquarterOf}}$ State Library of Kansas $\xrightarrow{\text{location}}$ Topeka, Kansas
	(jurisdictionOf, location)	104	Kansas $\xrightarrow{\text{jurisdictionOf}}$ Kansas Department of Revenue $\xrightarrow{\text{location}}$ Topeka, Kansas
NYT-Bestseller	(previousWork, author)	77	The Burning Room $\xrightarrow{\text{previousWork}}$ The Black Box (novel) $\xrightarrow{\text{author}}$ Michael Connelly
	(subsequentWorkOf , previousWork, author)	47	Worst Case $\xrightarrow{\text{subsequentWorkOf}}$ Run for Your Life (novel) $\xrightarrow{\text{previousWork}}$ Step on a Crack $\xrightarrow{\text{author}}$ James Patterson
	(subsequentWorkOf, author)	43	The Burning Room $\xrightarrow{\text{subsequentWorkOf}}$ The Black Box (novel) $\xrightarrow{\text{author}}$ Michael Connelly
	(seriesOf, author)	26	In Death $\xrightarrow{\text{seriesOf}}$ Treachery in Death $\xrightarrow{\text{author}}$ Nora Roberts
GREC Institution	(highschool)	31	Casey Clausen $\xrightarrow{\text{highschool}}$ Bishop Alemany High School
	(careerStation, team)	23	Eric Loyd $\xrightarrow{\text{careerStation}}$ Eric Loyd $\xrightarrow{\text{team}}$ University of Central Florida
	(education)	22	Stuart Dybek $\xrightarrow{\text{education}}$ St. Rita of Cascia High School
	(team)	20	Eric Loyd $\xrightarrow{\text{team}}$ University of Central Florida

these triples. Also note how Knowledge Stream is able to "bubble up" the most intuitively relevant facts by channeling a large flow through their corresponding paths (indicated by their wider edges). While other approaches either rely on availability of path patterns curated by knowledge engineers,

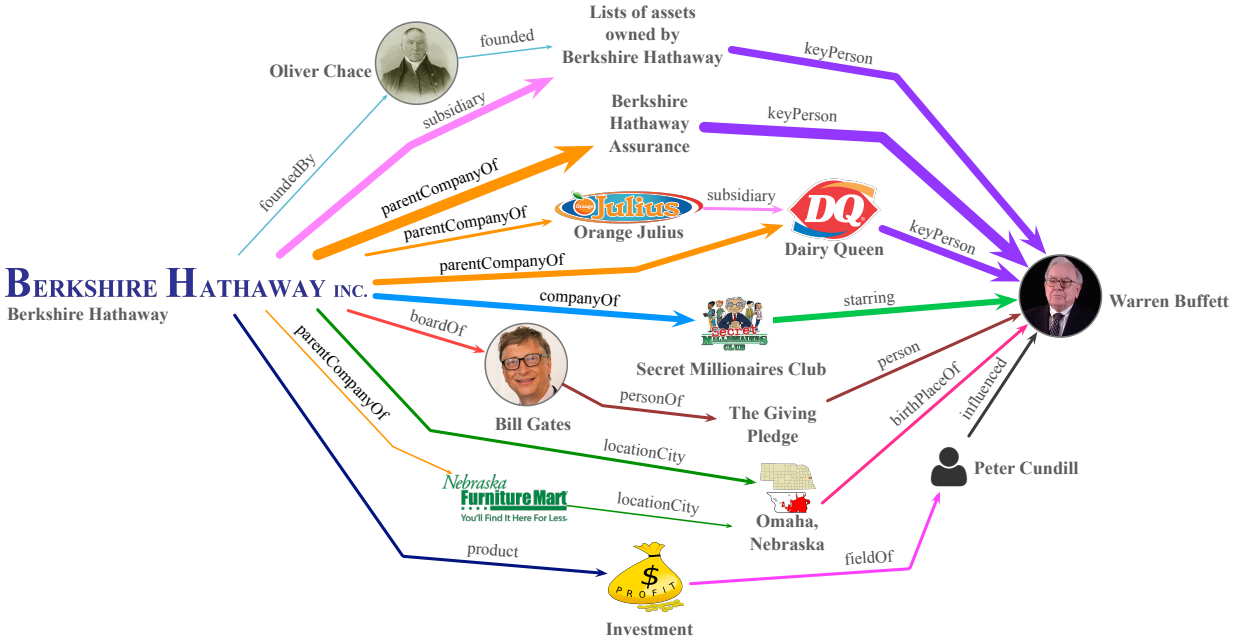


Figure 5.4: Relevant facts about a target claim as surfaced by Knowledge Stream.

or mine them using a large number of labeled examples, KS automatically surfaces relevant ground facts in an unsupervised way. We believe that it is the first computational fact-checking approach featuring such an expressive power.

5.7 Connection to Prior Work

We find a few interesting connections between KS and prior work.

The idea of using maximum flow to measure node similarity in citation graphs has been explored by Lu et al. (2001). There, the capacity of an edge was designed to be proportional to $1/d^k$ where k is related to the shortest distance of the endpoints of the edge to the node pair, and d is an empirically chosen parameter to control the weight of paths of different length. This choice ensures that longer paths contribute less to the overall flow, and thus the similarity. In contrast, KS penalizes flow on longer paths by dividing the flow by length of the path (Eqn. 5.3).

Communicability in a complex network is understood as the degree to which "traffic" can be transferred or communicated between a pair of nodes in the network. Its traditional definition only

considers the shortest path between the nodes. However, since not all flow processes take the shortest path (Borgatti, 2005), Estrada and Hatano (2008) introduced a generalized measure that also takes into account longer paths and walks besides the shortest path. The measure gives less weight to the contribution by long walks than short ones. This strategy is in line with that of KS, which lowers the flow contribution of paths based on their length. However, KS currently only considers simple paths, unlike their measure that also includes walks, meaning same node is allowed to be visited multiple times.

Xu et al. (2016a) introduced Path Entropy (PE) index, a similarity index for node pair based on entropies of simple paths connecting the nodes. The entropy of a path is approximated by the sum of entropies of its links, and the entropy of a direct link is measured based on the probability of its occurrence in the network given that there is no degree correlation among nodes. PE index is similar to KS in two respects: (1) Both KS and PE index evaluate the heterogeneity of paths, although in different ways. A path in KS contributes by its capacity to carry flow, whereas a path in PE index contributes based on its entropy, and (2) Both approaches favor short paths over long ones. KS uses specificity to identify short paths, whereas PE index penalizes long paths using a factor equal to the reciprocal of their length.

5.8 Summary

Network flow theory (Ahuja et al., 1993) has guided the design of many applications in engineering, logistics, manufacturing, and so on. In this chapter, we have shown that it can also serve as a useful toolbox for reasoning about facts, and for fact checking in particular.

We presented a novel, network-flow based approach called Knowledge Stream (KS) to assess and explain the truthfulness of a statement of fact by leveraging its greater semantic context in a knowledge graph. KS finds a stream of short paths between a triple's subject and object nodes to transfer maximum knowledge between them, and uses the flow on these paths to assign a truth value to the triple. Evaluation of KS on a diverse set of real-world and synthetic test cases shows that its performance is on par with the state of the art. Moreover, in many cases, multiple paths found by

KS can provide additional evidence to support fact checking. Besides good performance, KS can also automatically uncover useful path patterns and relevant facts about a claim.

Chapter 6

Assessing Relevance of Type-Like Triples

In this chapter, we look at a special case of fact checking, which involves making relevance assessments for triples that specify the type of an entity.

6.1 Triple Scoring Problem

The triple `(Wolfgang Amadeus Mozart, profession, Composer)` specifies that Mozart was a composer by associating the type entity Composer with the subject entity Wolfgang Amadeus Mozart under the predicate *profession*. Triples such as these that relate subject entities to their types from a predefined ontology are called *type-like triples*, and their predicate relations such as *profession* or *nationality* are called *type-like relations* (TLRs).

For person entity queries like "american actors," or "professions of Albert Einstein", search engines need to rank relevant type-like triples ahead of irrelevant ones. For example, for the query "american actors," a candidate set may consist of individuals such as Johnny Depp, Leonardo DiCaprio, George Bush, Hillary Clinton, and Lady Gaga as their subject entities, since all of them have appeared in movies. However, only Johnny Depp and Leonardo DiCaprio are popularly known as actors, while George Bush and Hillary Clinton are known as politicians, and Lady Gaga as a musician. In case of the second query "professions of Albert Einstein," the candidate set consists of Einstein's professions, namely Theoretical Physicist, Philosopher, Mathematician and Teacher. However, the fact that Einstein was a theoretical physicist is intuitively more relevant than the fact that he was a philosopher, mathematician or teacher. Thus, a quantitative relevance assessment is needed for such type-like triples.

A solution to the problem can cover both the use cases: (1) ranking candidate entities based on a

query type (e.g., ranking actors for the query "american actors"), and (2) ranking multiple types of the same query entity (e.g., ranking all aforementioned professions of Albert Einstein).

However, measuring the degree to which an entity belongs to a particular type is a non-trivial task, just like the problem of gauging relevance of documents for a search engine query. The main issue here is the subjective and ill-defined notion of "degree," which can be hard to quantify. Moreover, opinions often differ about an entity's type. Measures such as popularity, i.e., number of mentions of an entity in a knowledge base or text corpus, do not adequately correctly capture it either. Nevertheless, one of the insights shared by Bast et al. (2015) from their crowd-sourcing effort to obtain human relevance judgments is that there is still a broad consensus in general. Given the need for relevance assessment of triples, and the fact that humans do have a general agreement about them, assigning a *relevance score* to a triple is an important task, and has been termed as the *triple scoring* problem in literature. Such a score quantifies the degree of relevance to which its entity belongs to a particular type. The relevance scores are often simply referred to as *triple scores*.

The problem was first introduced by Bast et al. (2015), who showed that random guessing is ineffective, and also presented a number of algorithms leveraging Freebase and Wikipedia articles' text. The problem was also framed as a challenge at the Web Search and Data Mining Cup 2017 (WSDM Cup 2017) (Heindorf et al., 2017).[1]

Formally, the triple scoring problem is stated as follows:

> Given a triple from a type-like relation, compute a score (in the range 0 through 7) that measures the relevance of a statement expressed by the triple compared to other triples from the same relation.

Here, a triple with a score of 7 is considered as the most relevant, whereas that with a score of 0 is the least relevant. Bast et al. (2015) created two datasets of human-judged triples for evaluating this problem, one each for two TLRs, namely *profession* and *nationality*. Seven raters were asked to judge each triple as primary (label 1) or secondary (label 0), and the score of a triple represents the number of these raters who considered the profession/nationality to be primary for the person in the

[1] http://www.wsdm-cup-2017.org/triple-scoring.html

triple. A score of 0 for a triple indicates all the raters considered the profession/nationality to be secondary. Note that secondary does not imply a false fact here; all triples are known to be true. Table 6.1 shows a few triples judged by humans for the two relations.

Table 6.1: Relevance scores assigned by humans. A score of 7 means the triple is highly relevant, whereas a score of 0 indicates least relevant.

Type-Like Relation	Example Triple	Relevance Score
profession	Barack Obama, profession, Politician	7
	Barack Obama, profession, Law professor	1
	Barack Obama, profession, Lawyer	0
	Barack Obama, profession, Author	0
	Albert Einstein, profession, Theoretical Physicist	7
	Albert Einstein, profession, Philosopher	4
	Albert Einstein, profession, Mathematician	4
	Albert Einstein, profession, Teacher	0
nationality	Walter Scott, nationality, Scotland	7
	Walter Scott, nationality, United Kingdom	3
	Albert Einstein, nationality, Germany	7
	Albert Einstein, nationality, Switzerland	4
	Albert Einstein, nationality, United States of America	4

We present a supervised learning approach called *RelSifter*[2] for the triple scoring problem, which leverages facts from a knowledge graph (KG) to assign a triple score. We show that it is effective at the task for the two type-like relations, and its performance is only slightly behind that from the state of the art.

6.2 RelSifter

We hypothesize that a type (object of a type-like triple) such as *Actor* or *Scientist*, can be described by a set of most pertinent "activities" that describe people known to have that type. For example,

[2] **Rel**evance **Sifter**

actors are known to star in movies, and scientists are known for their academic affiliations and scholarly works. In a KG, such activities are simply the predicates associated with outgoing edges of people known to have a certain profession/nationality. In the following, we motivate the intuition behind our approach using the *profession* type-like relation, however, the same intuition applies to any other TLR.

In RelSifter, we first identify the set of all person entities s of the KG who are subject of some type-like triple, say, `(`s`, profession, Actor)` — the set of all actors. We then define the set of *all* activities as the set of outgoing predicates of any triple in which the subject entity is an actor. Of course, not all activities are equally pertinent; this set would contain several predicates that are not informative of being an actor. A good example of an uninformative predicate is *bornIn* – everybody is born somewhere after all, and hence has a triple in the KG indicating this fact. Some other predicates, however, are more pertinent to being an actor. For example, actors are known to star in movies or plays. We approach the problem of identifying the most pertinent activities (predicates) for actors, or any profession, by means of TF-IDF, and pick only the top k predicates as *the* activities describing that profession.

A pooled set of these top k activities across all professions in the KG forms the feature set for our learning problem. For each triple in a training set corresponding to *profession* TLR, we measure the extent of "overlap" between observed activities of the person in the triple and the defining activities for the profession (the specific type, e.g., *Actor*) as identified earlier. To this end, we create a sparse vector representation for each triple in terms of the activities as features. A non-zero entry for an activity in this vector represents the TF-IDF value for the activity, indicating that the subject of the triple is engaged in the defining activity. The resultant sparse matrix forms our input for learning algorithms that create a model for predicting relevance scores for future type-like triples.

In the following, we first look at how to derive the feature set, followed by the details of learning the model and its evaluation.

6.3 Features from Knowledge Graph

We measure the pertinence of an activity to a type o by combining two dimensions of its informativeness: *popularity* and *focus*.

Pertinence by popularity: Popularity aims to capture the intuition that an activity is representative of an object profession (or nationality) if it is commonly found among people known to have that profession (or nationality). As previously mentioned, activity information is generally found on a properly defined subset of the second-degree neighbors of the type entity in the KG. See Figure 6.1 for an example from DBpedia. One of the top activities for profession *Actor* according to this measure is *starring*.

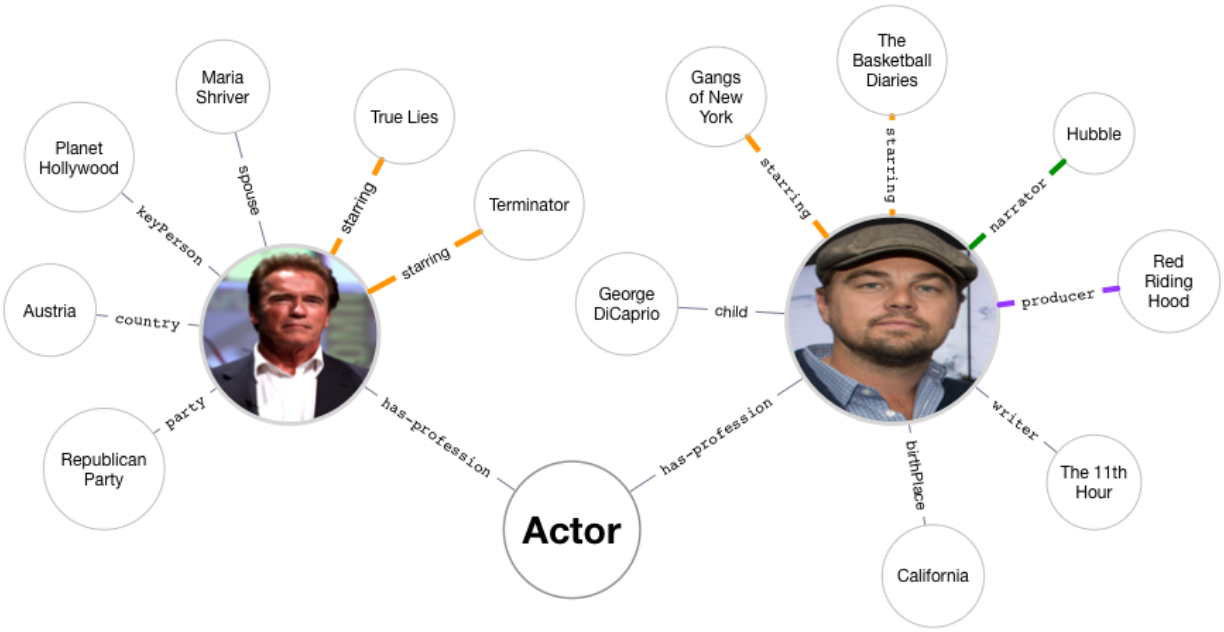


Figure 6.1: Few activities indicated by the predicates of triples associated with actors, *Arnold Schwarzenegger* and *Leonardo DiCaprio*. Pertinent activities are highlighted in color. Distinct colors correspond to distinct activities in DBpedia.

Formally, let U represent the universe of people in a KG, S_o the set of people with profession o, and R_o the set of relations associated with people in S_o. Let us consider a predicate $r \in R_o$ and denote by $g_{S_o}(r)$ the number of person entities in S_o with activity r, and by $g_U(r)$ the number of person entities in U with activity r.

The pertinence of r to o by popularity P using TF-IDF weighting is:

$$P_o(r) = \log\left(1 + g_{S_o}(r)\right) \cdot \log\left(\frac{|U|}{g_U(r)}\right).$$

Pertinence by focus: Another form of pertinence of an activity is its role in defining the "focus" of a person. An activity may be representative of an object profession (or nationality) if it is relatively more frequent among people having that profession (or nationality) than among those who do not. For example, the top activity for *Actor* according to this measure is *cast member*, which is different from the top activity by popularity.

Let us denote by $f_{S_o}(r)$ the frequency of r for persons in S_o. Then the pertinence of r to o by focus F again using TF-IDF weighting is defined as:

$$F_o(r) = \log\left(1 + f_{S_o}(r)\right) \cdot \log\left(\frac{\sum_{r' \in R_o} \left(1 + f_U(r') - f_{S_o}(r')\right)}{1 + f_U(r) - f_{S_o}(r)}\right)$$

where the second term reflects the inverse relative frequency of relations in persons *not* having the profession o. The number 1 inside logarithm in above formulae ensures that the term is well-defined.

Combined pertinence of an activity: We finally combine the measures of pertinence by popularity and focus to define the *combined pertinence* C as their product

$$\mathrm{C}_o(r) = \mathrm{P}_o(r) \cdot \mathrm{F}_o(r),$$

which we use to identify the most informative activities of a TLR.

Table 6.2 and Table 6.3 show a few examples of the top k (with $k = 5$) activities for a few professions and nationalities, in decreasing order of their combined pertinence as derived from Wikidata. The top k activities of all professions (or nationalities) when pooled together forms the set of features for our learning algorithm. As one might expect, the set of activities of related professions do overlap. For example, the activity *lyrics by* is among the top k activities of both *Pianist* and

Lyricist. However, its position differs in the two rankings.

Table 6.2: Top 5 activities for a few professions in Wikidata per combined pertinence.

Rank	Pianist	Lyricist	Architect	Talk show host	Mathematician
1.	composer	lyrics by	architect	presenter	proved by
2.	follower of	score by	architectural style	production company	solved by
3.	instrument	librettist	structural engineer	narrator	doctoral student
4.	lyrics by	performer	main building contractor	executive producer	doctoral advisor
5.	record label	composer	notable work	influenced by	field of work

Table 6.3: Top 5 activities for a few nationalities in Wikidata per combined pertinence.

Rank	Italy	Australia	India	Soviet Union
1.	production designer	spoken text audio	pronunciation audio	backup or reserve team or crew
2.	feed URL	solved by	solved by	astronaut mission
3.	party chief repr.	cuisine	audio	academic degree
4.	executive body	handedness	filmography	corporate officer
5.	powerplant	bowling style	bowling style	contributing factor of

6.4 Triple Score Learning

Given a type-like triple, we wish to learn a model that predicts its relevance score in the discrete, ordinal range from 0 to 7. In this section, we describe how we solve this relevance scoring problem using a supervised learning algorithm. Our idea is to measure the overlap between the activities that describe the person in the triple, and those characterizing the type as identified in previous section.

We have two training sets — one consisting of 515 triples from the *profession* TLR, and other of 162 triples from the *nationality* TLR. We construct a feature vector for each of these triples in the following way: an entry in the feature vector takes the range-normalized combined pertinence value of the feature if the person (subject) of the triple has the feature (activity), and zero otherwise. The resultant sparse feature matrix forms our input for the learning algorithms.

We experiment using three learning algorithms: Ordinal Logistic Regression (OLR) (Rennie and Srebro, 2005), and two ensemble learners, namely, Random Forest (Breiman, 2001) and Adaboost (Freund and Schapire, 1997) with decision trees as base learners. OLR is a generalization of logistic

regression for ordinal data, and is designed to take advantage of the ordinal structure in the class labels (i.e., the human-judged scores). Since all mistakes are not equal in case of ordinal data, we choose an *all-thresholds loss* that penalizes predictions made farther away from the expected output. See (Rennie and Srebro, 2005) for more details about this loss function and its implication. We apply Random Forest and Adaboost by treating the problem as a multi-class classification.

To control overfitting, we perform 10-fold cross-validation. For OLR, we use a package called *mord* by Pedregosa-Izquierdo (2015) that implements the approaches introduced by Rennie and Srebro (2005). We experiment with the following penalty values for its parameter α: 1, 5, 10, 15, 20, 50, 75, 100, 250, 500, 1000, whereas for Random Forest and AdaBoost, we try bags of 10, 50, 100, 250 and 500 base estimators. For every algorithm, we select the best model by a grid search over the range of their respective parameters.

6.5 Evaluation

We evaluate RelSifter using two KGs: DBpedia and Wikidata. We supplement the data in these KGs with additional set of triples provided as part of the WSDM Cup 2017 Triple Scoring challenge. Specifically, we incorporate 499,244 `(person, profession)` and 318,779 `(person, nationality)` pairs that feature 200 professions, 100 nationalities, and 385,426 persons. As before, we use DBpedia with information about its ontology, instances-types and mapping-based properties. In case of Wikidata, we use its dataset of *simple statements* that includes its taxonomy, class hierarchy of Wikidata properties, class membership information, and simple statements that excludes triples related to contextual and provenance information. Table 6.4 gives a summary of our undirected KGs used to evaluate RelSifter. In general, our experience suggests that Wikidata has comparatively more information per entity than that in DBpedia. This is also evident from Figure 6.2.

For each of the two KGs, we extract the top k activities for every object profession and nationality by their combined pertinence, build a feature matrix corresponding to these features, and learn a model to predict unseen triples. We compare performance of models built by the three learning

Table 6.4: Statistics of Knowledge Graphs.

KG	Nodes	Relations	Triples
DBpedia	6.1M	663	48M
Wikidata	29.4M	839	234M

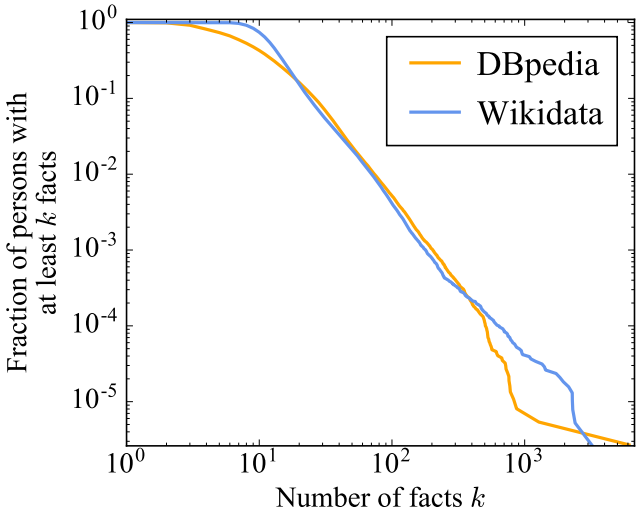


Figure 6.2: Complementary cumulative distribution of the number of facts per person in the two KGs.

algorithms using accuracy with a maximum deviation by $\delta = 2$ as the evaluation metric. See Section 2.5.2 for the definition of this metric.

Figure 6.3 shows the results obtained by experimenting with different values of k. For both KGs, Random Forest works better in case of *profession* TLR, whereas Ordinal Logistic Regression is the outright winner for the *nationality* TLR. In general, Adaboost delivers relatively poor performance.

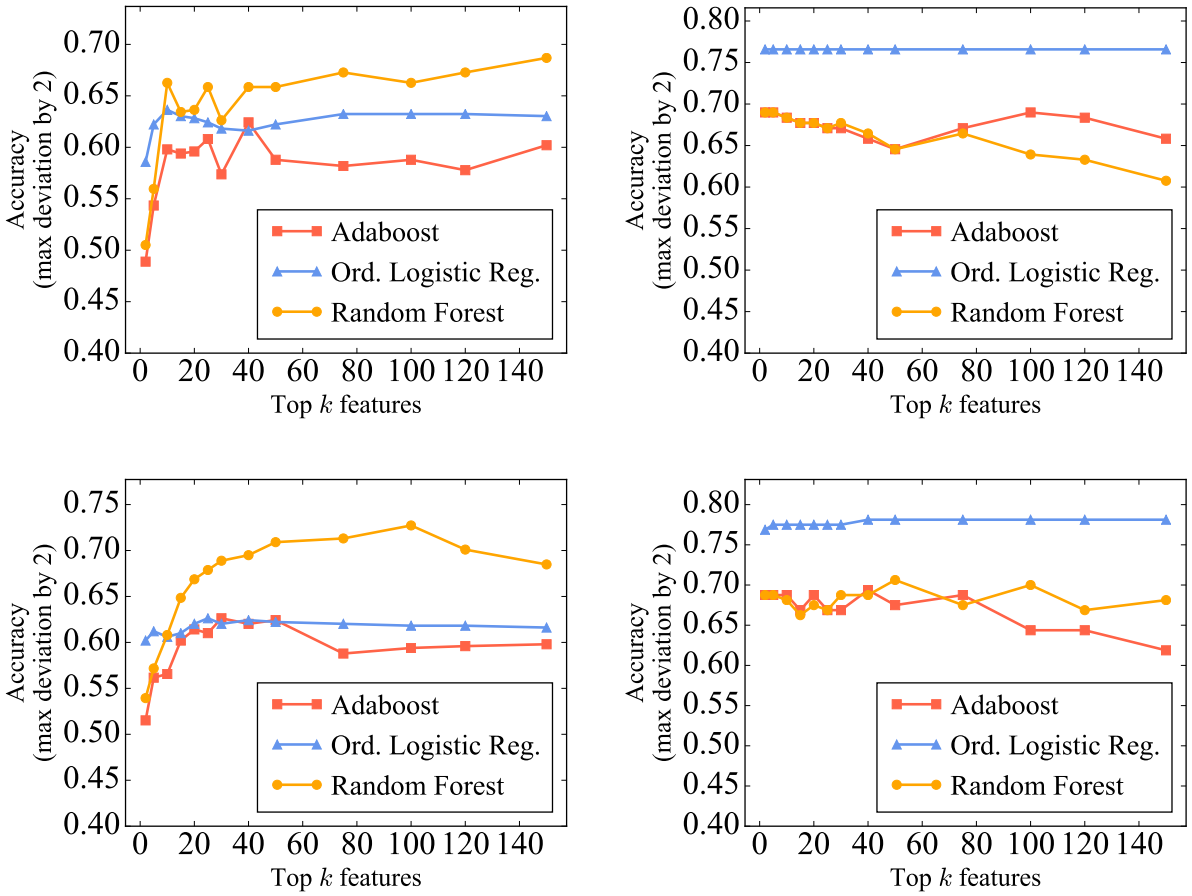


Figure 6.3: Performance by combined pertinence. Top: DBpedia; Bottom: Wikidata. Left: Profession; Right: Nationality.

The performance of RelSifter for both TLRs is comparable or lags by a small margin behind the best performing algorithms by Bast et al. (2015), which we consider as our baseline. Our best models result in 73% accuracy for the *profession* TLR and 78% for the *nationality* TLR. An overview of results from better approaches that surpass this baseline can be found in (Bast et al., 2017).

6.6 Connection to Prior Work

As mentioned earlier, the triple scoring problem was first defined by Bast et al. (2015), who also introduced a variety of algorithms that use text from Wikipedia articles and Freebase as a knowledge base. In one approach called *Words Counting*, a logistic regression classifier is learned to predict whether a triple is primary or secondary. The features for this model consist of words in the text of a person's Wikipedia article, while the weights were determined in the following manner. For each profession (and likewise for nationality), words describing persons' profession in positive examples were extracted, their TF-IDF values computed, and the inverse rank of words based on these TF-IDF values were taken as the weight for logistic regression classifier. Here, the positive examples consisted of triples in which the person had a single profession (e.g., `(Humphrey Bogart, profession, Actor)`), whereas negative examples consisted of triples with the person not known to have a certain profession (e.g., `(Barack Obama, profession, Actor)`. For a new triple, the sum of the weights of indicator mentions (i.e., words that are observed in article text for the person in the triple) is used to make a prediction.

This approach bears similarity to RelSifter. Like words in Words Counting, RelSifter first finds a set of predicates as features using a TF-IDF weighting scheme, and later uses them in a classifier. However, RelSifter differs in two respects: (1) it only uses the top k features as opposed to all words from the text, and uses range-normalized TF-IDF values as features instead of inverse rank of words; and (2) it learns a single ordinal or multi-class classifier for each type-like relation such as *profession*, unlike Words Counting which creates a distinct model for every single profession.

6.7 Summary

In this chapter, we presented a novel supervised approach called RelSifter for the triple scoring problem. The task can be seen as a special case of fact checking triples, since like any ordinary triple, the goal is to compute a numeric score, but for type-like triples. RelSifter assigns relevance score to a triple by measuring the overlap between activities that describe its subject entity and those

characterizing the type (object) of the triple. When scoring triples from *profession* and *nationality* type-like relations, RelSifter results in an accuracy of 73% and 78% respectively. This suggests that it can be an effective approach for evaluating facts, despite the skewness in number of facts per individual available in KGs.

Chapter 7

Conclusion and Future Work

Fact checking is an important activity to prevent dubious claims and unverified rumors from spreading on online social platforms. Through this research, we have introduced novel ways of performing automated fact checking of elementary claims that are expressible as triples. We now summarize some of the strengths and limitations of our approaches – Knowledge Linker (KL), Relational Knowledge Linker (KL-REL), and Knowledge Stream (KS). The limitations also suggest ways of improving and extending the approaches. We conclude with a discussion on a few research directions for the future.

7.1 Strengths of KL, KL-REL and KS

1. **Accurate predictions**: All the approaches deliver comparable performance to the state-of-the-art algorithms for fact checking, as shown by experiments on several hand-crafted and real-world datasets.

2. **Unsupervised algorithms**: None of our algorithms require labeled data to make predictions. Supervised algorithms on the other hand require many labeled examples for reliable estimation of the model parameters.

3. **Ability to discover relational patterns**: We have seen the ability of KS to discover relational paths (i.e., sequences of edge types) in a data-driven way (Section 5.5). This feature has implications for supporting relation extraction in machine reading projects where typically one needs to provide a seed set of rules in the form of such paths.

4. **Ability to combine multiple evidence and surface relevant facts**: By employing concepts

in network flow theory, KS is able to incorporate evidence based on multiple paths, and thereby improve predictive performance. These paths (single path by KL and KL-REL, and multiple by KS) not only offer transparency to the prediction, but also represent relevant facts (Section 5.6) that may bring to light interesting relationships, which can otherwise be hard to search for from a deluge of information sources (Cohen et al., 2011a).

7.2 Limitations, Extensions and Future Work

1. **Evaluation of relational similarity**: The success of KL-REL and KS hinges on the appropriate measurement of relational similarity (Section 4.3). The development and evaluation of effective relational similarity metrics is an important avenue of future work.

   One way to evaluate our approach based on the line graph of a KG is to compare them to human judgments sought by, say, crowd-sourcing the task. Such judgments would also help understand the problem better.

   Another possible direction could be to learn the relational similarity in a supervised way, and compare the learned similarities to that from current approach. A possible way to learn could be to start with a random state of values for entries in the relational similarity matrix (whose entries represent pairwise similarity between relations), and refine them while optimizing for AUROC using KL-REL or KS on a given set of labeled examples. Assumptions can be made to model the nature of correlation between relations, which may also guide the refinement process.

2. **Design of KS capacities**: In KS, we defined the capacity of an edge as a product of its relational similarity to the target predicate, and the specificity of node to which it is incident. This choice is only a matter of design, and alternatives could be explored. The capacities could incorporate metadata from the KG itself, for example confidence scores from information extraction phase (see, e.g., YAGO2).

3. **Improving KS implementation for better run-time**: Our version of KS relies on successive

path-finding, which can be slow for triples involving subjects with a large search space. Our implementation takes a few minutes to check each triple with DBpedia, and the run-time varies widely across different datasets. See Figure 5.2. Other approaches could be explored in the future. For example, the relaxation algorithm or the network simplex algorithm discussed by Ahuja et al. (1993, Ch. 9, 11) has better theoretical and run-time behavior.

4. **Measures of node generality**: We have used the degree of a node as a crude measure of its generality. Alternatively, one could use the more sophisticated notion of *betweenness centrality* (Freeman, 1977) of a node as a measure of its generality. A comparative evaluation of the two measures used toward fact-checking task could be an interesting exercise.

5. **Transitive closures**: A central hypothesis underlying our algorithms has been that nodes of a true triple are close to each other, and this closeness can be measured by distinct ways of computing the transitive closure of a weighted graph (Simas and Rocha, 2015). We explored two such closures for KL (Section 4.4), namely metric and ultra-metric closure, both of which use a single path. In the future, alternatives such as k-shortest paths and *diffusion closure* (Simas and Rocha, 2015) could be explored. Like KS, diffusion closure uses multiple paths, but unlike KS that uses the idea of maximum flow, it assigns a similarity score to a node pair based on the harmonic mean of the number of paths connecting them. A comparison of the three approaches, KS, k-shortest paths and diffusion closure, is another avenue to explore.

6. **Modeling time and location**: Many KGs such as YAGO2 and Wikidata contain facts augmented with spatio-temporal details. Checking facts that may be true only during a certain time frame or at a particular location is another challenge. One way to extend KL-REL or KS to handle such facts could be to bias the search toward those areas of the KG that may contain facts valid during that interval or near that place.

7. **Ranking surfaced facts**: In KS, to surface facts relevant to a triple, we ranked the set of paths in a stream based on their flow values. Devising alternative ways to rank such facts, reflecting their novelty, diversity, or serendipity is another interesting thread of future research.

Literature evaluating recommender systems (Gunawardana and Shani, 2015) can help in this regard. This idea is similar to the idea of ranking news leads in FactWatcher (Hassan et al., 2014)[1] based on their recency, popularity, or interestingness.

8. **Understanding the difficulty of checking a fact**: The results in this thesis show that the same algorithm can perform with different levels of accuracy when applied to different datasets. This suggests that some datasets contain facts that are more challenging to check than others. How to gauge the difficulty of checking a fact is an open question for future research. Progress in this direction will also enable the design of datasets with desired level of difficulty for evaluating fact-checking algorithms.

9. **Positive-Unlabeled Learning**: In a KG, all triples are considered to be true facts, whereas the status of unobserved triples is unknown, i.e., they include a mix of false triples and undiscovered true triples. A paradigm in the machine learning literature known as Positive-Unlabeled Learning (PU-Learning) (du Plessis et al., 2014; Elkan and Noto, 2008; Jain et al., 2017) aims to improve the performance of classifiers in such situations where one has a set of positive examples, and a large set of unlabeled examples that contains a small set of positive examples and an abundance of negative ones. The approaches underlying this paradigm make optimal use of available data through appropriate assumptions about the class distributions. Viewing fact checking as a problem of learning a probabilistic classifier, the PU-Learning paradigm could lead to novel fact-checking methods.

10. **Robustness of fact checking**: In this thesis, we have not explored the robustness of our algorithms with respect to inaccuracies or incompleteness of the KG. This is an important direction for future work. One way to explore robustness could be to sample different subsets of the KG and/or inject random noise, such as deletion of known facts or addition of spurious ones.

[1] http://idir.uta.edu/factwatcher/nba.php

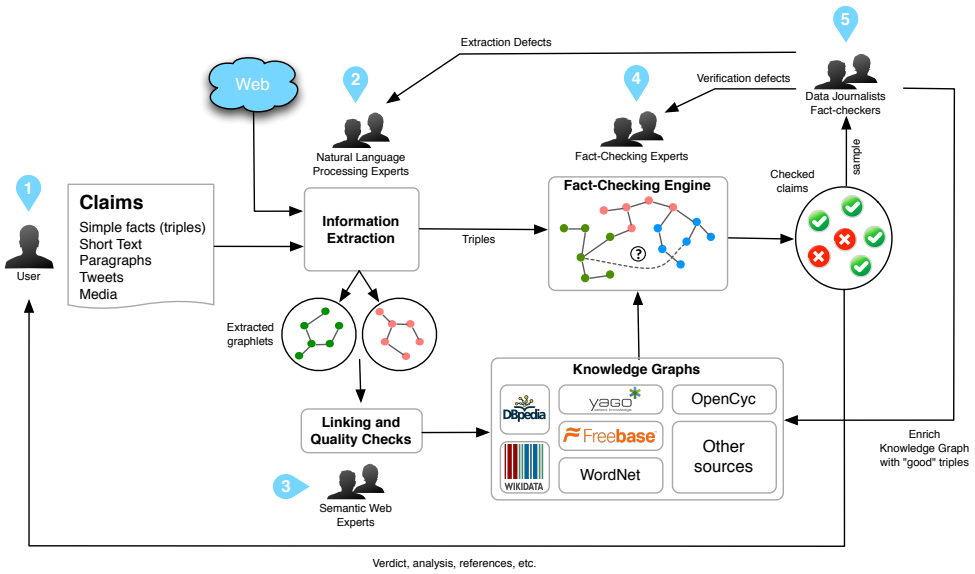


Figure 7.1: A sketch of fact-checking pipeline. The numbers in blue indicate different expert groups engaged to make this workflow functional.

7.3 Fact Checking - A Pragmatic Workflow

Based on the methods introduced in this dissertation, we sketch in Fig. 7.1 a high-level workflow for fact checking, taking into consideration the current state of KGs, the state of the art in fact checking, and availability of off-the-shelf technology for NLP and IE. This workflow requires collaboration of experts from diverse areas such as journalism, linguistics, social science, NLP, machine learning, databases, and semantic web, to collectively push the frontier.

Here, an interface may accept claims from an end user (#1), which could be as simple as a triple, or more complex like a paragraph or text in natural language. This input can then be parsed using IE techniques to extract one or more check-worthy claim triples, with their entities and relations linked to corresponding information in a KG. These triples are then fed to an automated fact-checker such as ours, which makes a judgment on their veracity. The checked claims (also commonly called *fact-checks*) can be presented back to the user, supplemented with evidence and explanation for the verdict.

In the short term, a fact-checking workflow such as this can assist journalists and human fact checkers (#5) in quickly detemining veracity of simple claims made in articles. Moreover, they could

also actively participate in the knowledge acquisition and fact-checking process. For example, they could investigate computationally checked claims, and work with fact-checking experts (#4) (or NLP experts (#2)) to solve any verification (or extraction) issues. Moreover, since a computational fact checker's ability depends on the type of knowledge available in KG, they could contribute towards identifying knowledge that may be missing or needs to be incorporated. Experts on Semantic Web or knowledge engineers concerning a given domain (#3) can help in this respect, besides continuing to create structured facts from other existing unstructured sources.

Such communication among experts cutting across disciplines can throw light on new problems, and foster future research in their individual fields. Shared schema and standards around data and sharing mechanisms can be defined to ensure reliability of data and interoperability between different journalistic organizations. The workflow outlined above may indicate a paradigm shift in the way contemporary human fact checkers investigate claims. However, with timely access to vast repositories of knowledge, and their hands on efficient tools to aid their analysis, the impact on their fact-checking productivity could be phenomenal. This direction could ultimately help mitigate the risks of misinformation spread in our society.

7.4 Concluding Remarks

The thesis of this research has been that the set of short paths connecting the subject and object concept entities of a claim triple holds the explanatory power to assign a truth value to the triple, representing its degree of truthfulness. Although our algorithms have shown competitive performance on fact-checking datasets, we are still a long way from the dream of having a completely automated fact checker. Much remains to be done on both fronts, on the knowledge acquisition as well as on the natural language understanding front. We believe the methods introduced in this dissertation provide a foundation for future research on computational fact checking, and for the development of useful tools to assist human fact checkers, investigative journalists and ordinary citizens.

Appendix A

Data Sets

A.1 Evaluation Data Sets

The datasets used in this thesis can be downloaded from http://carl.cs.indiana.edu/data/fact-checking/.

Below I list only the true triples from each dataset.

A.1.1 FLOTUS

Table A.1: US President vs. Spouse. Number inside parenthesis beside subject/object represents the number of facts about them available in DBpedia 2016.

President		First Lady	
WGH	Warren G. Harding (56)	FH	Florence Harding (9)
CC	Calvin Coolidge (80)	GC	Grace Coolidge (17)
HH	Herbert Hoover (76)	LHH	Lou Henry Hoover (18)
FDR	Franklin D. Roosevelt (178)	ER	Eleanor Roosevelt (40)
HST	Harry S. Truman (152)	BT	Bess Truman (16)
DDE	Dwight D. Eisenhower (176)	ME	Mamie Eisenhower (20)
JFK	John F. Kennedy (160)	JKO	Jacqueline Kennedy Onassis (35)
LBJ	Lyndon B. Johnson (170)	LBJ	Lady Bird Johnson (21)
RN	Richard Nixon (230)	PN	Pat Nixon (29)
GF	Gerald Ford (160)	BF	Betty Ford (26)
JC	Jimmy Carter (217)	RC	Rosalynn Carter (20)

Table A.1 – continued from previous page

President		First Lady	
RR	Ronald Reagan (382)	NR	Nancy Reagan (28)
GHWB	George H. W. Bush (269)	BB	Barbara Bush (25)
BC	Bill Clinton (509)	HC	Hillary Clinton (56)
GWB	George W. Bush (697)	LB	Laura Bush (23)
BO	Barack Obama (1263)	MO	Michelle Obama (20)

A.1.2 Oscars

Table A.2: Best Picture vs. Director

Year	Movie	Director
1927-28	7th Heaven (1927 film) (15)	Frank Borzage (76)
1927-28	Two Arabian Knights (14)	Lewis Milestone (63)
1928-29	The Divine Lady (13)	Frank Lloyd (83)
1929-30	All Quiet on the Western Front (1930 film) (11)	Lewis Milestone (63)
1932-33	Cavalcade (1933 film) (14)	Frank Lloyd (83)
1934	It Happened One Night (12)	Frank Capra (78)
1935	The Informer (1935 film) (14)	John Ford (160)
1936	Mr. Deeds Goes to Town (11)	Frank Capra (78)
1937	The Awful Truth (9)	Leo McCarey (64)
1939	Gone with the Wind (film) (18)	Victor Fleming (67)
1940	The Grapes of Wrath (film) (14)	John Ford (160)
1941	How Green Was My Valley (film) (16)	John Ford (160)
1942	Mrs. Miniver (film) (14)	William Wyler (58)
1943	Casablanca (film) (8)	Michael Curtiz (186)
1944	Going My Way (13)	Leo McCarey (64)
1945	The Lost Weekend (film) (9)	Billy Wilder (76)
1946	The Best Years of Our Lives (15)	William Wyler (58)
1948	The Treasure of the Sierra Madre (film) (11)	John Huston (100)
1949	A Letter to Three Wives (14)	Joseph L. Mankiewicz (81)
1950	All About Eve (13)	Joseph L. Mankiewicz (81)
1951	A Place in the Sun (film) (12)	George Stevens (78)

Table A.2 – continued from previous page

Year	Movie	Director
1952	The Quiet Man (15)	John Ford (160)
1953	From Here to Eternity (16)	Fred Zinnemann (40)
1954	On the Waterfront (9)	Elia Kazan (44)
1955	Marty (film) (13)	Delbert Mann (52)
1956	Giant (1956 film) (7)	George Stevens (78)
1957	The Bridge on the River Kwai (18)	David Lean (53)
1958	Gigi (1958 film) (16)	Vincente Minnelli (45)
1959	Ben-Hur (1959 film) (18)	William Wyler (58)
1960	The Apartment (12)	Billy Wilder (76)
1961	Jerome Robbins (11)	Robert Wise (64)
1962	Lawrence of Arabia (film) (7)	David Lean (53)
1964	My Fair Lady (film) (14)	George Cukor (67)
1965	The Sound of Music (film) (11)	Robert Wise (64)
1966	A Man for All Seasons (1966 film) (13)	Fred Zinnemann (40)
1967	The Graduate (14)	Mike Nichols (51)
1968	Oliver! (film) (17)	Carol Reed (42)
1969	Midnight Cowboy (13)	John Schlesinger (36)
1970	Patton (film) (15)	Franklin J. Schaffner (35)
1971	The French Connection (film) (15)	William Friedkin (40)
1973	The Sting (13)	George Roy Hill (29)
1974	The Godfather Part II (11)	Francis Ford Coppola (91)
1976	Rocky (15)	John G. Avildsen (34)
1977	Annie Hall (7)	Woody Allen (122)

Table A.2 – continued from previous page

Year	Movie	Director
1978	The Deer Hunter (17)	Michael Cimino (21)
1979	Kramer vs. Kramer (15)	Robert Benton (24)
1980	Ordinary People (11)	Robert Redford (73)
1981	Reds (film) (15)	Warren Beatty (35)
1982	Gandhi (film) (9)	Richard Attenborough (80)
1983	Terms of Endearment (11)	James L. Brooks (51)
1984	Amadeus (film) (15)	Miloš Forman (28)
1985	Out of Africa (film) (14)	Sydney Pollack (57)
1986	Platoon (film) (8)	Oliver Stone (75)
1987	The Last Emperor (17)	Bernardo Bertolucci (31)
1988	Rain Man (13)	Barry Levinson (65)
1989	Born on the Fourth of July (film) (15)	Oliver Stone (75)
1990	Dances with Wolves (7)	Kevin Costner (55)
1991	The Silence of the Lambs (film) (16)	Jonathan Demme (51)
1992	Unforgiven (7)	Clint Eastwood (111)
1994	Forrest Gump (8)	Robert Zemeckis (49)
1995	Braveheart (16)	Mel Gibson (71)
1996	The English Patient (film) (12)	Anthony Minghella (30)
1997	Titanic (1997 film) (22)	James Cameron (58)
1998	Saving Private Ryan (15)	Steven Spielberg (125)
1999	American Beauty (1999 film) (18)	Sam Mendes (25)
2000	Traffic (2000 film) (14)	Steven Soderbergh (62)
2001	A Beautiful Mind (film) (19)	Ron Howard (92)

Table A.2 – continued from previous page

Year	Movie	Director
2002	The Pianist (2002 film) (15)	Roman Polanski (50)
2003	The Lord of the Rings: The Return of the King (9)	Peter Jackson (43)
2004	Million Dollar Baby (11)	Clint Eastwood (111)
2005	Brokeback Mountain (14)	Ang Lee (29)
2006	The Departed (20)	Martin Scorsese (96)
2007	No Country for Old Men (film) (8)	Coen brothers (43)
2008	Slumdog Millionaire (17)	Danny Boyle (28)
2009	The Hurt Locker (18)	Kathryn Bigelow (27)
2011	The Artist (film) (11)	Michel Hazanavicius (23)
2012	Life of Pi (film) (15)	Ang Lee (29)
2013	Gravity (film) (13)	Alfonso Cuarón (34)

A.1.3 US-Capital

Table A.3: US State vs. Capital. Number inside parenthesis beside subject/object represents the number of facts about them available in DBpedia 2016.

Region	State	Capital
Midwest	Illinois (10194)	Springfield, Illinois (327)
	Indiana (5337)	Indianapolis (1333)
	Iowa (3020)	Des Moines, Iowa (567)
	Kansas (2887)	Topeka, Kansas (302)
	Michigan (6136)	Lansing, Michigan (369)
	Minnesota (6368)	Saint Paul, Minnesota (904)
	Missouri (3830)	Jefferson City, Missouri (121)
	Nebraska (1934)	Lincoln, Nebraska (382)
	North Dakota (1100)	Bismarck, North Dakota (161)
	Ohio (8279)	Columbus, Ohio (1345)
	South Dakota (1080)	Pierre, South Dakota (64)
	Wisconsin (5687)	Madison, Wisconsin (768)
Northeast	Connecticut (2375)	Hartford, Connecticut (719)
	Maine (1880)	Augusta, Maine (118)
	Massachusetts (5527)	Boston (4491)
	New Hampshire (1307)	Concord, New Hampshire (182)
	New Jersey (5530)	Trenton, New Jersey (438)
	New York (19733)	Albany, New York (619)
	Pennsylvania (11841)	Harrisburg, Pennsylvania (467)
	Rhode Island (905)	Providence, Rhode Island (1177)

Table A.3 – continued from previous page

Region	State	Capital
	Vermont (1129)	Montpelier, Vermont (85)
South	Alabama (3019)	Montgomery, Alabama (445)
	Arkansas (2625)	Little Rock, Arkansas (789)
	Delaware (850)	Dover, Delaware (131)
	Florida (7179)	Tallahassee, Florida (473)
	Georgia (U.S. state) (4088)	Atlanta (3112)
	Kentucky (4161)	Frankfort, Kentucky (139)
	Louisiana (3988)	Baton Rouge, Louisiana (711)
	Maryland (3663)	Annapolis, Maryland (296)
	Mississippi (1979)	Jackson, Mississippi (488)
	North Carolina (3652)	Raleigh, North Carolina (685)
	Oklahoma (2539)	Oklahoma City (905)
	South Carolina (1924)	Columbia, South Carolina (582)
	Tennessee (3112)	Nashville, Tennessee (2836)
	Texas (9817)	Austin, Texas (1669)
	Virginia (5243)	Richmond, Virginia (1223)
	West Virginia (4547)	Charleston, West Virginia (317)
West	Alaska (1448)	Juneau, Alaska (157)
	Arizona (2407)	Phoenix, Arizona (1212)
	California (24454)	Sacramento, California (1081)
	Colorado (2949)	Denver (1566)
	Hawaii (1194)	Honolulu (1197)
	Idaho (1589)	Boise, Idaho (406)

Table A.3 – continued from previous page

Region	State	Capital
	Montana (1328)	Helena, Montana (146)
	Nevada (1797)	Carson City, Nevada (145)
	New Mexico (1586)	Santa Fe, New Mexico (301)
	Oregon (3843)	Salem, Oregon (339)
	Utah (2032)	Salt Lake City (1216)
	Washington (state) (4000)	Olympia, Washington (207)
	Wyoming (1028)	Cheyenne, Wyoming (221)

A.1.4 World-Capital

Table A.4: World Capital vs. Country

Region	Capital	State
	Angola	Luanda
	Angola	Luanda
	Benin	Porto-Novo
	Botswana	Gaborone
	Burkina Faso	Ouagadougou
	Burundi	Bujumbura
	Cameroon	Yaoundé
	Cape Verde	Praia
	Central African Republic	Bangui
	Comoros	Moroni Comoros
	Democratic Republic of the Congo	Kinshasa
	Republic of the Congo	Brazzaville
	Djibouti	Djibouti
	Egypt	Cairo
	Equatorial Guinea	Malabo
	Eritrea	Asmara
	Ethiopia	Addis Ababa
	Gabon	Libreville
	The Gambia	Banjul
	Ghana	Accra
	Guinea	Conakry
	Guinea Bissau	Bissau
	Kenya	Nairobi

Table A.4 – continued from previous page

Region	Capital	State
	Lesotho	Maseru
	Liberia	Monrovia
	Libya	Tripoli
	Madagascar	Antananarivo
	Malawi	Lilongwe
	Mali	Bamako
	Mauritania	Nouakchott
	Mauritius	Port Louis
	Morocco	Rabat
	Mozambique	Maputo
	Namibia	Windhoek
	Niger	Niamey
	Nigeria	Abuja
	Rwanda	Kigali
	São Tomé and Príncipe	São Tomé
	Senegal	Dakar
	Seychelles	Victoria Seychelles
	Sierra Leone	Freetown
	Somalia	Mogadishu
	South Africa	Pretoria
	South Sudan	Juba
	Sudan	Khartoum
	Swaziland	Mbabane
	Tanzania	Dodoma
	Togo	Lomé

Table A.4 – continued from previous page

Region	Capital	State
	Tunisia	Tunis
	Uganda	Kampala
	Zambia	Lusaka
	Zimbabwe	Harare
	Afghanistan	Kabul
	Armenia	Yerevan
	Azerbaijan	Baku
	Bahrain	Manama
	Bangladesh	Dhaka
	Bhutan	Thimphu
	Brunei	Bandar Seri Begawan
	Burma	Naypyidaw
	Cambodia	Phnom Penh
	China	Beijing
	Cyprus	Nicosia
	East Timor	Dili
	Georgia country	Tbilisi
	India	New Delhi
	Indonesia	Jakarta
	Iran	Tehran
	Iraq	Baghdad
	Israel	Jerusalem
	Japan	Tokyo
	Jordan	Amman
	Kazakhstan	Astana

Asia

Table A.4 – continued from previous page

Region	Capital	State
	North Korea	Pyongyang
	South Korea	Seoul
	Kuwait	Kuwait City
	Kyrgyzstan	Bishkek
	Laos	Vientiane
	Lebanon	Beirut
	Malaysia	Kuala Lumpur
	Maldives	Malé
	Mongolia	Ulan Bator
	Nepal	Kathmandu
	Oman	Muscat Oman
	Pakistan	Islamabad
	State of Palestine	Jerusalem
	Philippines	Manila
	Qatar	Doha
	Saudi Arabia	Riyadh
	Singapore	Singapore
	Syria	Damascus
	Tajikistan	Dushanbe
	Thailand	Bangkok
	Turkey	Ankara
	Turkmenistan	Ashgabat
	United Arab Emirates	Abu Dhabi
	Uzbekistan	Tashkent
	Vietnam	Hanoi

Table A.4 – continued from previous page

Region	Capital	State
Europe	Albania	Tirana
	Andorra	Andorra la Vella
	Austria	Vienna
	Belarus	Minsk
	Belgium	Brussels
	Bosnia and Herzegovina	Sarajevo
	Bulgaria	Sofia
	Croatia	Zagreb
	Czech Republic	Prague
	Denmark	Copenhagen
	Estonia	Tallinn
	Finland	Helsinki
	France	Paris
	Germany	Berlin
	Greece	Athens
	Hungary	Budapest
	Iceland	Reykjavík
	Republic of Ireland	Dublin
	Italy	Rome
	Latvia	Riga
	Liechtenstein	Vaduz
	Lithuania	Vilnius
	Luxembourg	Luxembourg
	Republic of Macedonia	Skopje
	Malta	Valletta

Table A.4 – continued from previous page

Region	Capital	State
	Monaco	Monaco
	Montenegro	Podgorica
	Norway	Oslo
	Poland	Warsaw
	Portugal	Lisbon
	Romania	Bucharest
	Russia	Moscow
	San Marino	City of San Marino
	Serbia	Belgrade
	Slovakia	Bratislava
	Slovenia	Ljubljana
	Spain	Madrid
	Sweden	Stockholm
	Switzerland	Bern
	Ukraine	Kiev
	United Kingdom	London
	Vatican City	Vatican City
	Antigua and Barbuda	St.Johns
	Bahamas	Nassau Bahamas
	Barbados	Bridgetown
	Belize	Belmopan
	Canada	Ottawa
	Costa Rica	San José Costa Rica
	Cuba	Havana
	Dominica	Roseau

Table A.4 – continued from previous page

Region	Capital	State
	Dominican Republic	Santo Domingo
	El Salvador	San Salvador
	Guatemala	Guatemala City
	Haiti	Port-au-Prince
	Honduras	Tegucigalpa
	Jamaica	Kingston Jamaica
	Mexico	Mexico City
	Kingdom of the Netherlands	Amsterdam
	Nicaragua	Managua
	Panama	Panama City
	Saint Kitts and Nevis	Basseterre
	Saint Lucia	Castries
	Saint Vincent and the Grenadines	Kingstown
	Trinidad and Tobago	Port of Spain
	United States	Washington D.C.
Oceania	Australia	Canberra
	Fiji	Suva
	Kiribati	South Tarawa
	Marshall Islands	Majuro
	Federated States of Micronesia	Palikir
	New Zealand	Wellington
	Papua New Guinea	Port Moresby
	Samoa	Apia
	Solomon Islands	Honiara
	Tonga	Nuku‘alofa

Table A.4 – continued from previous page

Region	Capital	State
	Tuvalu	Funafuti
	Vanuatu	Port Vila
South America	Argentina	Buenos Aires
	Bolivia	Sucre
	Brazil	Brasília
	Chile	Santiago
	Colombia	Bogotá
	Ecuador	Quito
	Guyana	Georgetown Guyana
	Paraguay	Asunción
	Peru	Lima
	Suriname	Paramaribo
	Uruguay	Montevideo
	Venezuela	Caracas

A.1.5 NYT-Bestseller

Table A.5: New York Times Bestseller vs. Author

	Book	Author
1	Killing Lincoln (4)	Bill O'Reilly (political commentator) (33)
2	Killing Kennedy (5)	Bill O'Reilly (political commentator) (33)
3	Obama's Wars (7)	Bob Woodward (24)
4	Words of Radiance (7)	Brandon Sanderson (33)
5	Humans of New York (4)	Brandon Stanton (8)
6	Dead in the Family (10)	Charlaine Harris (31)
7	Dead Reckoning (novel) (13)	Charlaine Harris (31)
8	The Southern Vampire Mysteries (19)	Charlaine Harris (31)
9	Chelsea Chelsea Bang Bang (7)	Chelsea Handler (29)
10	Lies That Chelsea Handler Told Me (9)	Chelsea Handler (29)
11	Wild: From Lost to Found on the Pacific Crest Trail (6)	Cheryl Strayed (8)
12	The Lost Symbol (11)	Dan Brown (20)
13	Inferno (Brown novel) (8)	Dan Brown (20)
14	The Social Animal (Brooks book) (8)	David Brooks (journalist) (17)
15	The Greater Journey (6)	David McCullough (29)
16	Let's Explore Diabetes with Owls (5)	David Sedaris (23)
17	What the Night Knows (9)	Dean Koontz (119)
18	In My Time: A Personal and Political Memoir (5)	Dick Cheney (57)

Table A.5 – continued from previous page

	Book	Author
19	Known and Unknown: A Memoir (8)	Donald Rumsfeld (45)
20	Fifty Shades of Grey (11)	E. L. James (18)
21	Committed: A Skeptic Makes Peace with Marriage (5)	Elizabeth Gilbert (16)
22	In the Garden of Beasts (5)	Erik Larson (author) (8)
23	A Dance with Dragons (11)	George R. R. Martin (96)
24	Decision Points (6)	George W. Bush (697)
25	Gone Girl (novel) (4)	Gillian Flynn (18)
26	The Overton Window (4)	Glenn Beck (45)
27	Live Wire (novel) (1)	Harlan Coben (30)
28	The Casual Vacancy (5)	J. K. Rowling (45)
29	The Cuckoo's Calling (5)	J. K. Rowling (45)
30	Alex Cross, Run (10)	James Patterson (91)
31	Worst Case (1)	James Patterson (91)
32	The Land of Painted Caves (10)	Jean M. Auel (16)
33	House Rules (novel) (6)	Jodi Picoult (28)
34	Paterno (book) (4)	Joe Posnanski (10)
35	The Confession (novel) (7)	John Grisham (65)
36	The Racketeer (novel) (7)	John Grisham (65)
37	The Litigators (13)	John Grisham (65)
38	Sycamore Row (6)	John Grisham (65)
39	Game Change (11)	John Heilemann (9)
40	Earth (The Book) (18)	Jon Stewart (52)

Table A.5 – continued from previous page

	Book	Author
41	Imagine: How Creativity Works (5)	Jonah Lehrer (7)
42	Freedom (Franzen novel) (4)	Jonathan Franzen (21)
43	Extremely Loud and Incredibly Close (7)	Jonathan Safran Foer (14)
44	The Help (5)	Kathryn Stockett (7)
45	Life (Keith Richards) (6)	Keith Richards (87)
46	Fall of Giants (8)	Ken Follett (30)
47	Winter of the World (8)	Ken Follett (30)
48	Edge of Eternity (8)	Ken Follett (30)
49	Spoken from the Heart (4)	Laura Bush (23)
50	Unbroken: A World War II Story of Survival, Resilience, and Redemption (5)	Laura Hillenbrand (8)
51	The Obama Diaries (4)	Laura Ingraham (18)
52	61 Hours (8)	Lee Child (37)
53	Personal (novel) (8)	Lee Child (37)
54	Worth Dying For (novel) (8)	Lee Child (37)
55	Never Go Back (Child novel) (8)	Lee Child (37)
56	A Wanted Man (8)	Lee Child (37)
57	David and Goliath (book) (7)	Malcolm Gladwell (14)
58	The Liberty Amendments (5)	Mark Levin (12)
59	No Easy Day (10)	Mark Owen (46)
60	12th of Never (novel) (10)	Maxine Paetro (8)
61	The Reversal (9)	Michael Connelly (46)
62	The Burning Room (10)	Michael Connelly (46)

Table A.5 – continued from previous page

	Book	Author
63	The Black Box (novel) (11)	Michael Connelly (46)
64	The Fifth Witness (5)	Michael Connelly (46)
65	The Gods of Guilt (5)	Michael Connelly (46)
66	Flash Boys (6)	Michael Lewis (19)
67	The Big Short (6)	Michael Lewis (19)
68	No Apology (6)	Mitt Romney (38)
69	The Longest Ride (9)	Nicholas Sparks (31)
70	In Death (12)	Nora Roberts (30)
71	The Bone Bed (8)	Patricia Cornwell (31)
72	The Wise Man's Fear (9)	Patrick Rothfuss (11)
73	Drift: The Unmooring of American Military Power (3)	Rachel Maddow (24)
74	Zealot: The Life and Times of Jesus of Nazareth (3)	Reza Aslan (20)
75	Duty: Memoirs of a Secretary at War (8)	Robert Gates (40)
76	Towers of Midnight (10)	Robert Jordan (36)
77	A Memory of Light (11)	Robert Jordan (36)
78	Water for Elephants (3)	Sara Gruen (9)
79	Going Rogue (13)	Sarah Palin (39)
80	Lean In (2)	Sheryl Sandberg (18)
81	Si-cology 1 (11)	Si Robertson (18)
82	My Beloved World (8)	Sonia Sotomayor (17)
83	The Grand Design (book) (8)	Stephen Hawking (71)

Table A.5 – continued from previous page

	Book	Author
84	Mr. Mercedes (5)	Stephen King (360)
85	Doctor Sleep (novel) (7)	Stephen King (360)
86	11/22/63 (6)	Stephen King (360)
87	The Girl Who Kicked the Hornets' Nest (12)	Stieg Larsson (20)
88	The Girl with the Dragon Tattoo (14)	Stieg Larsson (20)
89	Capital in the Twenty-First Century (9)	Thomas Piketty (17)
90	Bossypants (4)	Tina Fey (55)
91	Dead or Alive (novel) (8)	Tom Clancy (45)
92	Threat Vector (11)	Tom Clancy (45)
93	Steve Jobs (book) (9)	Walter Isaacson (13)

A.1.6 NBA-Team

Table A.6: NBA Player vs. Team

	Player	Team
1	Bob Pettit (7)	Atlanta Hawks
2	Satch Sanders (8)	Boston Celtics
3	Bill Russell (8)	Boston Celtics
4	Larry Bird (10)	Boston Celtics
5	John Havlicek (7)	Boston Celtics
6	Kevin McHale (basketball) (8)	Boston Celtics
7	Sam Jones (basketball) (7)	Boston Celtics
8	Tom Boerwinkle (7)	Chicago Bulls
9	Dirk Nowitzki (11)	Dallas Mavericks
10	Joe Dumars (7)	Detroit Pistons
11	Isiah Thomas (8)	Detroit Pistons
12	Al Attles (7)	Golden State Warriors
13	Paul Arizin (7)	Golden State Warriors
14	Rudy Tomjanovich (8)	Houston Rockets
15	Calvin Murphy (7)	Houston Rockets
16	Allen Leavell (8)	Houston Rockets
17	Jeff Foster (basketball) (8)	Indiana Pacers
18	Rik Smits (7)	Indiana Pacers
19	Reggie Miller (10)	Indiana Pacers
20	James Worthy (10)	Los Angeles Lakers
21	Michael Cooper (14)	Los Angeles Lakers

Table A.6 – continued from previous page

	Player	Team
22	Kobe Bryant (14)	Los Angeles Lakers
23	Magic Johnson (19)	Los Angeles Lakers
24	Elgin Baylor (8)	Los Angeles Lakers
25	Jerry West (10)	Los Angeles Lakers
26	Udonis Haslem (11)	Miami Heat
27	Bill Bradley (16)	New York Knicks
28	Willis Reed (7)	New York Knicks
29	Julius Erving (12)	Philadelphia 76ers
30	Alvan Adams (9)	Phoenix Suns
31	Jack Twyman (5)	Sacramento Kings
32	Tony Parker (18)	San Antonio Spurs
33	David Robinson (basketball) (6)	San Antonio Spurs
34	Tim Duncan (11)	San Antonio Spurs
35	Nick Collison (11)	Seattle SuperSonics
36	Fred Brown (basketball) (9)	Seattle SuperSonics
37	Nate McMillan (11)	Seattle SuperSonics
38	Mark Eaton (basketball) (6)	Utah Jazz
39	John Stockton (7)	Utah Jazz
40	Darrell Griffith (7)	Utah Jazz
41	Wes Unseld (8)	Washington Wizards

A.1.7 CEO

Table A.7: Company vs. CEO. Number inside parenthesis beside subject/object represents the number of facts about them available in DBpedia 2016.

	Company	CEO
1	ACE Limited (10)	Evan G. Greenberg (10)
2	AOL (82)	Tim Armstrong (executive) (6)
3	AT&T (67)	Randall L. Stephenson (12)
4	Accenture (23)	Pierre Nanterme (6)
5	Aditya Birla Group (34)	Kumar Mangalam Birla (21)
6	Adobe Systems (114)	Shantanu Narayen (13)
7	Advanced Micro Devices (40)	Rory Read (6)
8	Agenus (4)	Garo H. Armen (4)
9	Ahold (23)	John Rishton (3)
10	Airbus Group (45)	Tom Enders (8)
11	Alcatel-Lucent (23)	Michel Combes (14)
12	Alcoa (25)	Klaus Kleinfeld (14)
13	Alliance Boots (20)	Stefano Pessina (14)
14	Amazon.com (93)	Jeff Bezos (10)
15	American Airlines Group (15)	Doug Parker (5)
16	American Express (22)	Kenneth Chenault (8)
17	Apple Inc. (459)	Tim Cook (14)
18	ArcelorMittal (16)	Lakshmi Mittal (27)
19	BAE Systems (70)	Ian King (BAE Systems) (8)
20	BHP Billiton (40)	Andrew Mackenzie (businessman) (9)

Table A.7 – continued from previous page

	Company	CEO
21	BMW (457)	Norbert Reithofer (4)
22	BP (37)	Bob Dudley (6)
23	Banco Bilbao Vizcaya Argentaria (21)	Francisco González (banker) (4)
24	Bank of America (50)	Brian Moynihan (8)
25	Barclays (32)	Antony Jenkins (8)
26	Berkshire Hathaway (92)	Warren Buffett (26)
27	Best Buy (10)	Hubert Joly (8)
28	Bharti Enterprises (16)	Sunil Mittal (16)
29	Boston Consulting Group (12)	Rich Lesser (9)
30	CBS Corporation (93)	Leslie Moonves (16)
31	Campbell Soup Company (20)	Denise Morrison (6)
32	Canonical (company) (39)	Jane Silber (3)
33	Capital One (28)	Richard Fairbank (3)
34	Caterpillar Inc. (51)	Douglas R. Oberhelman (19)
35	Cavium (6)	Syed B. Ali (2)
36	Cisco Systems (84)	John T. Chambers (12)
37	Citigroup (46)	Michael Corbat (9)
38	Cognizant (11)	Francisco D'Souza (10)
39	Comcast (87)	Brian L. Roberts (12)
40	ConAgra Foods (10)	Gary Rodkin (3)
41	Crown Worldwide Group (13)	James E. Thompson (5)
42	DLF (company) (16)	Kushal Pal Singh (12)
43	Daimler AG (109)	Dieter Zetsche (15)

Table A.7 – continued from previous page

	Company	CEO
44	Dell (48)	Michael Dell (13)
45	Deutsche Bank (44)	Juergen Fitschen (4)
46	Deutsche Bank (44)	Anshu Jain (12)
47	Deutsche Post (26)	Frank Appel (7)
48	DineEquity (9)	Julia Stewart (businesswoman) (6)
49	EBay (30)	John Donahoe (12)
50	Electronic Arts (858)	Andrew Wilson (businessman) (7)
51	Ericsson (35)	Hans Vestberg (7)
52	ExxonMobil (34)	Rex Tillerson (11)
53	FUBU (5)	Daymond John (16)
54	Facebook (110)	Mark Zuckerberg (17)
55	FedEx (17)	Frederick W. Smith (10)
56	Fidelity Investments (28)	Abigail Johnson (12)
57	Ford Motor Company (1196)	Mark Fields (businessman) (12)
58	Foxconn (58)	Terry Gou (5)
59	GMA Network (615)	Felipe Gozon (22)
60	GMR Group (8)	Grandhi Mallikarjuna Rao (8)
61	GVK (conglomerate) (7)	Gunupati Venkata Krishna Reddy (9)
62	General Dynamics (70)	Phebe Novakovic (5)
63	General Electric (127)	Jeffrey R. Immelt (13)
64	Gerdau (11)	André Bier Gerdau Johannpeter (8)
65	GlaxoSmithKline (24)	Andrew Witty (8)
66	Global Telecom Holding (13)	Naguib Sawiris (9)

Table A.7 – continued from previous page

	Company	CEO
67	Goldman Sachs (38)	Lloyd Blankfein (12)
68	Google (333)	Larry Page (17)
69	Graham Holdings Company (23)	Donald E. Graham (12)
70	HCL Technologies (17)	Anant Gupta (11)
71	HSBC (57)	Stuart Gulliver (12)
72	Halliburton (11)	David J. Lesar (16)
73	Hewlett-Packard (89)	Meg Whitman (14)
74	Hindalco Industries (10)	Kumar Mangalam Birla (21)
75	Honeywell (26)	David M. Cote (7)
76	IGATE (8)	Ashok Vemuri (4)
77	IKEA (14)	Mikael Ohlsson (3)
78	Infosys (26)	Vishal Sikka (7)
79	Intel (111)	Brian Krzanich (7)
80	InterCall (7)	Scott Etzler (9)
81	J.Crew (19)	Mickey Drexler (2)
82	Juniper Networks (12)	Kevin Johnson (executive) (4)
83	Kaplan, Inc. (15)	Andrew S. Rosen (7)
84	Kingdom Holding Company (11)	Al-Waleed bin Talal (19)
85	Kodak (25)	Jeff Clarke (businessman) (7)
86	Lanco Infratech (13)	Madhusudhan Rao Lagadapati (5)
87	Land Securities (18)	Robert Noel (businessman) (7)
88	Las Vegas Sands (15)	Sheldon Adelson (13)
89	Lockheed Martin (97)	Marillyn Hewson (5)

Table A.7 – continued from previous page

	Company	CEO
90	MGA Entertainment (11)	Isaac Larian (15)
91	Manchester United F.C. (2505)	David Gill (executive) (14)
92	Mercedes-Benz (316)	Dieter Zetsche (15)
93	Merrill Lynch (29)	John Thain (11)
94	MetLife (15)	C. Robert Henrikson (3)
95	Microsoft (612)	Satya Nadella (13)
96	Mile High Comics (8)	Chuck Rozanski (8)
97	Mindtree (13)	Krishnakumar Natarajan (6)
98	Morgan Stanley (37)	James P. Gorman (12)
99	Mossy Oak (6)	Toxey Haas (9)
100	Motorola (69)	Greg Brown (businessman) (5)
101	Mozilla (25)	Mitchell Baker (10)
102	NBCUniversal (178)	Jeff Zucker (14)
103	NIIT (8)	Vijay K. Thadani (7)
104	Nation Media Group (14)	Wilfred Kiboro (6)
105	National Amusements (42)	Sumner Redstone (17)
106	News Corporation (60)	Rupert Murdoch (33)
107	Nike, Inc. (29)	Mark Parker (3)
108	Nintendo (870)	Satoru Iwata (27)
109	Nokia (314)	Rajeev Suri (12)
110	Nortel (20)	Mike S. Zafirovski (10)
111	Novartis (21)	Daniel Vasella (8)
112	Oracle Corporation (119)	Larry Ellison (22)

Table A.7 – continued from previous page

	Company	CEO
113	Outback Steakhouse (23)	Chris T. Sullivan (7)
114	PepsiCo (64)	Indra Nooyi (16)
115	Pizza Hut (13)	David C. Novak (3)
116	Playboy Enterprises (33)	Christie Hefner (12)
117	Popular, Inc. (18)	Richard Carrión (9)
118	Prudential Financial (13)	John Strangfeld (3)
119	Qantas (31)	Alan Joyce (executive) (9)
120	Qwest (28)	Richard Notebaert (11)
121	Ralph Lauren Corporation (15)	Ralph Lauren (15)
122	Reliance Anil Dhirubhai Ambani Group (22)	Anil Ambani (26)
123	Reliance Industries (29)	Mukesh Ambani (27)
124	Renault (324)	Carlos Ghosn (17)
125	Rocawear (6)	Jay Z (289)
126	Royal Bank of Canada (26)	Gordon Nixon (7)
127	Royal Dutch Shell (65)	Ben van Beurden (6)
128	SAP SE (33)	Bill McDermott (9)
129	SAS Institute (9)	James Goodnight (12)
130	Saab Automobile (37)	Jan-Åke Jonsson (12)
131	Samsung (126)	Lee Kun-hee (12)
132	SandRidge Energy (11)	Tom L. Ward (4)
133	Sbarro (15)	James J. Greco (16)
134	Scottrade (16)	Rodger O. Riney (2)

Table A.7 – continued from previous page

	Company	CEO
135	Seagate Technology (15)	Stephen J. Luczo (11)
136	Simon Property Group (167)	David Simon (CEO) (8)
137	Singtel (36)	Chua Sock Koong (4)
138	Sirius Satellite Radio (35)	Mel Karmazin (10)
139	Sleep Country Canada (6)	Stephen K. Gunn (4)
140	SoftBank (23)	Masayoshi Son (15)
141	Sonic Drive-In (11)	J. Clifford Hudson (10)
142	Sony (530)	Kazuo Hirai (10)
143	Sony Computer Entertainment (586)	Andrew House (11)
144	Starbucks (22)	Howard Schultz (14)
145	Statoil (24)	Helge Lund (7)
146	Sulekha (9)	Satya Prabhakar (12)
147	SunTrust Banks (12)	James M. Wells, III (3)
148	Sun Hung Kai Properties (22)	Walter Kwok (8)
149	Sun Microsystems (69)	Jonathan I. Schwartz (5)
150	Sun Pharmaceutical (12)	Dilip Shanghvi (10)
151	SuperValu (United States) (22)	Jeff Noddle (5)
152	Syntel (13)	Bharat Desai (9)
153	TJX Companies (16)	Carol Meyrowitz (3)
154	Target Corporation (20)	Gregg Steinhafel (4)
155	Tata Consultancy Services (21)	Natarajan Chandrasekaran (7)
156	Telefónica (44)	César Alierta (6)
157	Telmex (15)	Carlos Slim (17)

Table A.7 – continued from previous page

	Company	CEO
158	Telstra (26)	David Thodey (9)
159	Temasek Holdings (26)	Ho Ching (9)
160	Tesco (33)	Philip Clarke (businessman) (8)
161	The Blackstone Group (55)	Stephen A. Schwarzman (11)
162	The Coca-Cola Company (111)	Muhtar Kent (16)
163	The Royal Bank of Scotland (21)	Ross McEwan (10)
164	The Siegel Group (7)	Steve Siegel (6)
165	The Travelers Companies (14)	Jay S. Fishman (4)
166	The Trump Organization (27)	Donald Trump (54)
167	The Walt Disney Company (300)	Bob Iger (15)
168	Tim Hortons (13)	Paul D. House (4)
169	Time Warner (154)	Jeff Bewkes (8)
170	Tod's (10)	Diego Della Valle (9)
171	Toyota (768)	Hiroshi Okuda (2)
172	Toys for Bob (25)	Paul Reiche III (16)
173	Tracinda (2)	Kirk Kerkorian (8)
174	TradeKing (7)	Donato A. Montanaro (2)
175	Turner Broadcasting System (86)	Philip I. Kent (3)
176	Twitter (41)	Dick Costolo (5)
177	U.S. Century Bank (18)	Octavio Hernández (1)
178	U.S. Steel (28)	John P. Surma (8)
179	UAL Corporation (12)	Glenn Tilton (2)
180	UBS (37)	Sergio Ermotti (8)

Table A.7 – continued from previous page

	Company	CEO
181	UGI Corporation (9)	Lon R. Greenberg (2)
182	United Parcel Service (19)	Scott Davis (businessman) (4)
183	Valero Energy (10)	William R. Klesse (2)
184	Verizon Communications (53)	Lowell McAdam (10)
185	Viacom (132)	Philippe Dauman (8)
186	Viking Range (6)	Fred Carl, Jr. (5)
187	Vodafone (43)	Vittorio Colao (8)
188	Volkswagen (296)	Martin Winterkorn (9)
189	Vulcan Inc. (10)	Paul Allen (45)
190	WWE (426)	Vince McMahon (63)
191	Walgreens (18)	Gregory Wasson (10)
192	Walmart (41)	Mike Duke (8)
193	Warner Bros. (3746)	Barry Meyer (10)
194	Wells Fargo (43)	John Stumpf (15)
195	Whole Foods Market (10)	John Mackey (businessman) (9)
196	Williams-Sonoma (12)	Laura J. Alber (2)
197	Winn-Dixie (8)	Peter Lynch (9)
198	Wipro (20)	T K Kurien (5)
199	YG Entertainment (131)	Yang Hyun-suk (40)
200	Yahoo! (106)	Marissa Mayer (18)
201	Yum! Brands (20)	David C. Novak (3)

A.1.8 US War

Table A.8: Civil War vs. Commander. Number inside parenthesis beside subject/object represents the number of facts about them available in DBpedia 2016.

	War	Commander
1	Battle of Antietam (345)	George B. McClellan (43)
2	Battle of Antietam (345)	Robert E. Lee (64)
3	Battle of Appomattox Court House (101)	Robert E. Lee (64)
4	Battle of Appomattox Court House (101)	Ulysses S. Grant (103)
5	Battle of Bentonville (185)	Joseph E. Johnston (47)
6	Battle of Bentonville (185)	William Tecumseh Sherman (47)
7	Battle of Cedar Creek (101)	Jubal Early (34)
8	Battle of Cedar Creek (101)	Philip Sheridan (32)
9	Battle of Champion Hill (100)	John C. Pemberton (9)
10	Battle of Champion Hill (100)	Ulysses S. Grant (103)
11	Battle of Chancellorsville (306)	Stonewall Jackson (39)
12	Battle of Chancellorsville (306)	Robert E. Lee (64)
13	Battle of Chancellorsville (306)	Joseph Hooker (28)
14	Battle of Chickamauga (314)	William Rosecrans (36)
15	Battle of Chickamauga (314)	Braxton Bragg (25)
16	Battle of Cold Harbor (249)	George Meade (33)
17	Battle of Cold Harbor (249)	Robert E. Lee (64)
18	Battle of Cold Harbor (249)	Ulysses S. Grant (103)
19	Battle of Five Forks (110)	George Pickett (16)

Table A.8 – continued from previous page

	War	Commander
20	Battle of Five Forks (110)	Philip Sheridan (32)
21	Battle of Fort Blakely (53)	St. John Richardson Liddell (8)
22	Battle of Fort Blakely (53)	Edward Canby (23)
23	Battle of Fort Donelson (77)	Andrew Hull Foote (15)
24	Battle of Fort Donelson (77)	Simon Bolivar Buckner (30)
25	Battle of Fort Donelson (77)	John B. Floyd (22)
26	Battle of Fort Donelson (77)	Ulysses S. Grant (103)
27	Battle of Fort Donelson (77)	Gideon Johnson Pillow (11)
28	Battle of Fort Stedman (31)	John Parke (9)
29	Battle of Fort Stedman (31)	John Brown Gordon (33)
30	Battle of Fort Sumter (15)	P. G. T. Beauregard (35)
31	Battle of Fort Sumter (15)	Robert Anderson (Civil War) (12)
32	Battle of Forts Jackson and St. Philip (34)	David Farragut (21)
33	Battle of Forts Jackson and St. Philip (34)	Johnson K. Duncan (4)
34	Battle of Franklin (1864) (151)	John Schofield (30)
35	Battle of Franklin (1864) (151)	John Bell Hood (33)
36	Battle of Fredericksburg (368)	Robert E. Lee (64)
37	Battle of Fredericksburg (368)	Ambrose Burnside (46)
38	Battle of Gaines's Mill (58)	George B. McClellan (43)
39	Battle of Gaines's Mill (58)	Fitz John Porter (20)
40	Battle of Gaines's Mill (58)	Robert E. Lee (64)

Table A.8 – continued from previous page

	War	Commander
41	Battle of Gettysburg (436)	John F. Reynolds (15)
42	Battle of Gettysburg (436)	George Meade (33)
43	Battle of Gettysburg (436)	Robert E. Lee (64)
44	Battle of Glorieta Pass (16)	William Read Scurry (9)
45	Battle of Glorieta Pass (16)	John P. Slough (6)
46	Battle of Glorieta Pass (16)	Charles L. Pyron (6)
47	Battle of Glorieta Pass (16)	John Chivington (12)
48	Battle of Island Number Ten (45)	William W. Mackall (7)
49	Battle of Island Number Ten (45)	John Pope (military officer) (16)
50	Battle of Island Number Ten (45)	John P. McCown (6)
51	Battle of Island Number Ten (45)	Andrew Hull Foote (15)
52	Battle of Jonesborough (210)	Oliver O. Howard (32)
53	Battle of Jonesborough (210)	George Henry Thomas (25)
54	Battle of Jonesborough (210)	William Tecumseh Sherman (47)
55	Battle of Jonesborough (210)	William J. Hardee (20)
56	Battle of Jonesborough (210)	John Bell Hood (33)
57	Battle of Malvern Hill (78)	Abraham Lincoln (96)
58	Battle of Malvern Hill (78)	Robert E. Lee (64)
59	Battle of Malvern Hill (78)	Jefferson Davis (71)
60	Battle of Malvern Hill (78)	Fitz John Porter (20)
61	Battle of Malvern Hill (78)	George B. McClellan (43)
62	Battle of Mansfield (32)	Richard Taylor (general) (22)
63	Battle of Mansfield (32)	Nathaniel P. Banks (56)

Table A.8 – continued from previous page

	War	Commander
64	Battle of Mobile Bay (107)	David Farragut (21)
65	Battle of Mobile Bay (107)	Gordon Granger (22)
66	Battle of Mobile Bay (107)	Franklin Buchanan (7)
67	Battle of Mobile Bay (107)	Richard Lucian Page (8)
68	Battle of Nashville (210)	John Bell Hood (33)
69	Battle of Nashville (210)	George Henry Thomas (25)
70	Battle of Pea Ridge (84)	Samuel Ryan Curtis (11)
71	Battle of Pea Ridge (84)	Earl Van Dorn (23)
72	Battle of Perryville (203)	Don Carlos Buell (14)
73	Battle of Perryville (203)	Alexander McDowell McCook (12)
74	Battle of Perryville (203)	Braxton Bragg (25)
75	Battle of Perryville (203)	Leonidas Polk (15)
76	Battle of Shiloh (304)	Albert Sidney Johnston (21)
77	Battle of Shiloh (304)	P. G. T. Beauregard (35)
78	Battle of Shiloh (304)	Don Carlos Buell (14)
79	Battle of Shiloh (304)	Ulysses S. Grant (103)
80	Battle of Spotsylvania Court House (222)	Ulysses S. Grant (103)
81	Battle of Spotsylvania Court House (222)	George Meade (33)
82	Battle of Spotsylvania Court House (222)	Robert E. Lee (64)
83	Battle of Stones River (247)	William Rosecrans (36)

Table A.8 – continued from previous page

	War	Commander
84	Battle of Stones River (247)	Braxton Bragg (25)
85	Battle of Westport (68)	Sterling Price (49)
86	Battle of Westport (68)	Samuel Ryan Curtis (11)
87	Battle of Wilson's Creek (60)	Nicholas Bartlett Pearce (5)
88	Battle of Wilson's Creek (60)	Benjamin McCulloch (8)
89	Battle of Wilson's Creek (60)	Sterling Price (49)
90	Battle of Wilson's Creek (60)	Samuel D. Sturgis (20)
91	Battle of Wilson's Creek (60)	Nathaniel Lyon (13)
92	Battle of the Crater (62)	Ambrose Burnside (46)
93	Battle of the Crater (62)	Robert E. Lee (64)
94	Battle of the Crater (62)	William Mahone (28)
95	Battle of the Wilderness (237)	Robert E. Lee (64)
96	Battle of the Wilderness (237)	George Meade (33)
97	Battle of the Wilderness (237)	Ulysses S. Grant (103)
98	First Battle of Bull Run (161)	Irvin McDowell (17)
99	First Battle of Bull Run (161)	Robert Patterson (11)
100	First Battle of Bull Run (161)	P. G. T. Beauregard (35)
101	First Battle of Bull Run (161)	Joseph E. Johnston (47)
102	First Battle of Winchester (27)	Nathaniel P. Banks (56)
103	First Battle of Winchester (27)	Stonewall Jackson (39)
104	Second Battle of Bull Run (238)	John Pope (military officer) (16)
105	Second Battle of Bull Run (238)	Robert E. Lee (64)
106	Second Battle of Fort Fisher (92)	Alfred Terry (19)

Table A.8 – continued from previous page

	War	Commander
107	Second Battle of Fort Fisher (92)	David Dixon Porter (17)
108	Second Battle of Fort Fisher (92)	Braxton Bragg (25)
109	Second Battle of Fort Fisher (92)	William H.C. Whiting (8)
110	Second Battle of Fort Fisher (92)	Robert Hoke (17)
111	Second Battle of Fort Fisher (92)	William Lamb (Confederate States Army officer) (3)
112	Second Battle of Petersburg (68)	Robert E. Lee (64)
113	Second Battle of Petersburg (68)	Ulysses S. Grant (103)
114	Second Battle of Petersburg (68)	George Meade (33)
115	Second Battle of Petersburg (68)	P. G. T. Beauregard (35)
116	Siege of Corinth (220)	P. G. T. Beauregard (35)
117	Siege of Corinth (220)	Henry Halleck (13)
118	Siege of Port Hudson (77)	Nathaniel P. Banks (56)
119	Siege of Port Hudson (77)	Franklin Gardner (17)
120	Siege of Vicksburg (275)	Ulysses S. Grant (103)
121	Siege of Vicksburg (275)	John C. Pemberton (9)
122	Third Battle of Petersburg (69)	Ulysses S. Grant (103)
123	Third Battle of Petersburg (69)	George Meade (33)
124	Third Battle of Petersburg (69)	Edward Ord (15)
125	Third Battle of Petersburg (69)	Robert E. Lee (64)
126	Third Battle of Petersburg (69)	A. P. Hill (17)

A.1.9 US-V. President

Table A.9: US. President vs. US Vice-President. Number inside parenthesis beside subject/object represents the number of facts about them available in DBpedia 2016.

	President	Vice President
1	George Washington (115)	John Adams (73)
2	John Adams (73)	Thomas Jefferson (112)
3	Thomas Jefferson (112)	Aaron Burr (28)
4	Thomas Jefferson (112)	George Clinton (vice president) (35)
5	James Madison (71)	Elbridge Gerry (21)
6	James Monroe (66)	Daniel D. Tompkins (25)
7	John Quincy Adams (64)	John C. Calhoun (45)
8	Andrew Jackson (104)	Martin Van Buren (51)
9	Martin Van Buren (51)	Richard Mentor Johnson (41)
10	William Henry Harrison (69)	John Tyler (77)
11	James K. Polk (58)	George M. Dallas (34)
12	Zachary Taylor (56)	Millard Fillmore (55)
13	Franklin Pierce (57)	William R. King (26)
14	James Buchanan (75)	John C. Breckinridge (50)
15	Abraham Lincoln (96)	Hannibal Hamlin (33)
16	Abraham Lincoln (96)	Andrew Johnson (67)
17	Ulysses S. Grant (103)	Schuyler Colfax (27)
18	Ulysses S. Grant (103)	Henry Wilson (28)
19	Rutherford B. Hayes (67)	William A. Wheeler (15)
20	James A. Garfield (56)	Chester A. Arthur (61)

Table A.9 – continued from previous page

	President	Vice President
21	Grover Cleveland (85)	Thomas A. Hendricks (30)
22	Benjamin Harrison (60)	Levi P. Morton (21)
23	Grover Cleveland (85)	Adlai Stevenson I (21)
24	William McKinley (83)	Garret Hobart (14)
25	William McKinley (83)	Theodore Roosevelt (118)
26	Theodore Roosevelt (118)	Charles W. Fairbanks (17)
27	William Howard Taft (81)	James S. Sherman (15)
28	Woodrow Wilson (92)	Thomas R. Marshall (21)
29	Warren G. Harding (56)	Calvin Coolidge (80)
30	Calvin Coolidge (80)	Charles G. Dawes (32)
31	Herbert Hoover (76)	Charles Curtis (31)
32	Franklin D. Roosevelt (178)	John Nance Garner (28)
33	Franklin D. Roosevelt (178)	Henry A. Wallace (22)
34	Franklin D. Roosevelt (178)	Harry S. Truman (152)
35	Harry S. Truman (152)	Alben W. Barkley (38)
36	Dwight D. Eisenhower (176)	Richard Nixon (230)
37	John F. Kennedy (160)	Lyndon B. Johnson (170)
38	Lyndon B. Johnson (170)	Hubert Humphrey (33)
39	Richard Nixon (230)	Spiro Agnew (24)
40	Richard Nixon (230)	Gerald Ford (160)
41	Gerald Ford (160)	Nelson Rockefeller (33)
42	Jimmy Carter (217)	Walter Mondale (33)
43	Ronald Reagan (382)	George H. W. Bush (269)

Table A.9 – continued from previous page

	President	Vice President
44	George H. W. Bush (269)	Dan Quayle (29)
45	Bill Clinton (509)	Al Gore (51)
46	George W. Bush (697)	Dick Cheney (57)
47	Barack Obama (1263)	Joe Biden (53)

A.2 List of Ideologies

1981 Irish hunger strike; 9/11 Truth movement; Abertzale; Abolished monarchy; African nationalism; African socialism; Afrikaner nationalism; Agorism; Agrarianism; Agrarian socialism; Ahlus Sunnah wal Jamaah (organisation); Albanians in the Republic of Macedonia; Alexander Lukashenko; Algeria; Algerianism; Alsace; Alter-globalization; American nationalism; Anarchism; Anarchist communism; Anarcho-capitalism; Anarcho-syndicalism; Andalusian nationalism; Anglo-Irish Treaty; Anglophile; Animal rights; Animal welfare; Anti; Anti-Americanism; Anti-capitalism; Anti-Catholicism in the United Kingdom; Anti-clericalism; Anti-communism; Anti Communism; Anti-corporate activism; Anti-Corruption; Anti-Esotericism; Anti-fascism; Anti-Federalism; Antifeminism; Anti-globalization movement; Anti-imperialism; Anti-Islamism; Anti-Judaism; Anti-Leninism; Anti-LGBT; Anti-liberal; Antimilitarism; Anti-nationalism; Anti-Polish sentiment; Anti-Revisionism; Antisemitism; Anti-Sovietism; Anti-Stalinist left; Anti-statism; Anti-taxation; Anti-war movement; Antiziganism; Anti-Zionism; António Ramalho Eanes; Apartheid in South Africa; Arab citizens of Israel; Arab nationalism; Arab socialism; Aragon; Armenian nationalism; Assyrian nationalism; Austrofascism; Authoritarianism; Autonomism; Awoism; Azerbaijani nationalism; Ba'athism; Balanced budget; Baloch nationalism; Bangladeshi nationalism; Basque nationalism; Bavaria; Bavarian independence; Bavarian Regionalism; Beauty; Beijing; Belarus; Bengali nationalism; Berberism; Berber people; Beta Israel; Big tent; Black Consciousness Movement; Black nationalism; Black supremacy; Bolivarianism; Bolivarian Revolution; Bolivia; Bosniaks; Bosnianism; Brahmin; Bre-

ton nationalism; Breton people; British Empire; British Fascism; British nationalism; Buddhism; Buddhist socialism; Bulgaria; Burmese Way to Socialism; Business; Caliphate; Cambodia; Canadian nationalism; Canarian nationalism; Cantonalism; Capitalism; Carinthian Slovenes; Castilian nationalism; Castroism; Catalan nationalism; Catalan separatism; Catholic Church; Catholic social teaching; Centralisation; Central Powers; Centre-left; Centre-right; Centrism; Chaldean Christians; Cham issue; Cham people (Asia); Chardal; Chavismo; Chinese nationalism; Chinese reunification; Chinese socialism; Christian communism; Christian democracy; Christian ethics; Christian humanism; Christianity; Christian left; Christian right; Christian socialism; Citizenship; Civil and political rights; Civil liberties; Classical liberalism; Classical Marxism; Clerical fascism; Clericalism; Colonialism; Communism; Communist Party of China; Communitarianism; Community politics; Conflict management; Congress of Verona (1943); Conservatism; Conservatism in Australia; Conservatism in Canada; Conservatism in Germany; Conservatism in the United States; Conservative Democrat; Conservative liberalism; Constitutionalism; Constitutional monarchy; Consumerism; Consumer protection; Cooperative; Co-operative economics; Cooperative federalism; Copyright; Cornerstone Group; Cornish Assembly; Cornish Autonomy; Cornish nationalism; Corporatism; Corsican nationalism; Cosmopolitanism; Côte d'Ivoire; Council communism; Creationism; Criticism of Islam; Croatian nationalism; Croats of Boka Kotorska; Croats of Vojvodina; Cultural conservatism; Cultural liberalism; Danish Realm; Decentralization; Decolonization; Degar; Degrowth; Demarchy; Democracy; Democratic liberalism; Democratic security; Democratic socialism; Democratic Struggle; Democratization; Deng Xiaoping Theory; Departments of Bolivia; Developmentalism; Development criticism; Devizes; Devolution; Devolved English parliament; Dictatorship of the proletariat; Direct democracy; Direct rule; Disputed status of Gibraltar; Distributism; Doi Moi; Dominant minority; Dominionism; Drug policy reform; Druze; Early Malay nationalism; Eastern Orthodox Church; Ecology; Ecology movement; Economic liberalism; Economic nationalism; Economic rationalism; Eco-socialism; Eco-sociality; E-democracy; Education; Egalitarianism; Egyptian nationalism; Éire Nua; Electoral reform; Electoral reform in New Zealand; Elitism; English independence; English nationalism; Environmentalism; Equal justice under law; Equal opportunity;

Equal rights; Eritrea; Especifismo; Ethnic-minority; Ethnic nationalism; Ethnocentrism; Ethnopluralism; Eurasianism; Euro; Eurocommunism; European integration; European People's Party; Euroscepticism; Expansionism; Experimental projects; Factions in the Democratic Party (United States); Fair trade; Falangism; Family values; Far-left politics; Faroe Islands; Far right in the United Kingdom; Far-right politics; Fascism; Fathers' rights movement; Federalism; Federalism in China; Federalist; Federation; Feminism; Feudalism; Filipino nationalism; Finland; Fiscal conservatism; Fiscal federalism; Flemish Movement; Fourth International Posadist; Francisco de Sá Carneiro; Francoist Spain; Francophile; Francophone; Franjo Tuđman; Freedom of information; Freedom of speech; Free love; Free market; Free trade; French Community of Belgium; French First Republic; French nationalism; Friesland; Frivolous political party; Fujimorism; Fundamentalism; Gaels; Gag rule; Galicianism (Galicia); Galician nationalism; Gaullism; Georgism; Gerald Götting; German Emperor; German language; German nationalism; Good governance; Goulash Communism; Grassroots; Grassroots democracy; Greater Armenia (political concept); Greater Israel; Greater Somalia; Greek nationalism; Green anarchism; Green conservatism; Greenland; Green liberalism; Green libertarianism; Green party; Green politics; Gross national happiness; Guerrilla warfare; Guevarism; Gwynedd; Halakha; Haredi Judaism; Harm reduction; Hazara people; Hindutva; Ho Chi Minh Thought; Holism; Honduras; House of Grimaldi; Hoxhaism; Hugo Chávez; Humanism; Humanist Movement; Human rights; Hungarian nationalism; Hungarians in Romania; Hungarians in Slovakia; Husakism; Hutu Power; Idealism; Imperial Preference; Impossibilism; Independence; Independent (politician); Indian nationalism; Indigenism; Indigenous rights; Individualism; Industrialisation; Integral humanism; Integralism; Intellectual property; Internal resistance to South African apartheid; Internationalism; Internationalism (politics); International Socialist Tendency; Internet censorship; Iranian nationalism; Iranian reform movement; Iraqi nationalism; Iraqi Turkmens; Irish nationalism; Irish republicanism; Iron Guard; Islam; Islamic democracy; Islamic fundamentalism; Islamic republic; Islamism; Islamophobia; Isolationism; Israeli–Palestinian conflict; Istria; Italian nationalism; Italy–Malta relations; Ivoirité; Jadid; James Madison; Japanese militarism; Japanese nationalism; J. B. Danquah; Jeffersonian democracy; Juche; Justice; Justicialist Party; Kahanism; Katarismo;

Kemalist ideology; Ketuanan Melayu; Keynesian economics; Kham; Khatim an-Nabuwwah; Khmer people; Kidderminster; Kirchnerism; Korean nationalism; Kuomintang; Kurdish nationalism; Kurdish people; Kwame Nkrumah; Kyrgyz nationalism; Laborer; Labor rights; Labor Zionism; Labour movement; Laïcité; Laissez-faire; Land of Israel; Latvia; Latvian people; Law and order (politics); Lebanese nationalism; Left centrism; Left communism; Left-libertarianism; Left–right politics; Left-wing nationalism; Left-wing politics; Legality of cannabis; Leninism; Liberal conservatism; Liberal democracy; Liberalism; Liberalism in Australia; Liberalism in Colombia; Liberalism in South Korea; Liberalism in the United States; Liberal movements within Islam; Liberal nationalism; Liberal socialism; Liberation theology; Liberism; Libertarian conservatism; Libertarian Democrat; Libertarianism; Libertarian Marxism; Libertarian socialism; Lieberman Plan; Lists of active separatist movements; Lithuania; Localism; Localism (politics); Luxemburgism; Macedonian nationalism; Malta–United Kingdom relations; Maltese; Maoism; Market liberalism; Market socialism; Martinique; Marxism; Marxism–Leninism; Masculism; Mass politics; Megali Idea; Metaxism; Microeconomic reform; Militant; Militarism; Millî Görüş; Minarchism; Minority rights; Miraism; Mixed economy; Mizrahi Jews; Mobutism; Moderate; Moderate Islamism; Moderation; Modernization; Modern Orthodox Judaism; Monarchism; Monarchy; Monarchy of Australia; Monetary reform; Montenegrin nationalism; Morality; Moravism; Morocco; Movement for the unification of Romania and Moldova; Multiculturalism; Multiethnic society; Muslim; Nacionalismo (Argentine political movement); Naga Nationalism; Nasserism; National Bolshevism; National Catholicism; National communism; National conservatism; Nationalism; Nationalization; National liberalism; National Liberation (historical); National Reconciliation; National syndicalism; Nativism (politics); Natural Capitalism: Creating the Next Industrial Revolution; Nazism; Neocolonialism; Neoconservatism; Neoconservatism (disambiguation); Neo-fascism; Neoliberalism; Neo-Nazism; Netherlands; Network neutrality; Neutrality (international relations); Nevis; New Democrats; New Left; New Nationalism; New Right; None of the above; Non-interventionism; Nonpartisan; Nonsectarian; Nonviolence; Nonviolent resistance; Non-voting; Nordic agrarian parties; Norse religion; Norwegian romantic nationalism; Objectivism (Ayn Rand); One country, two systems; One na-

tion conservatism; Open government; Opposition (politics); Opposition to immigration; Oromo people; Pacifism; Pakistani nationalism; Paleoconservatism; Paleolibertarianism; Palestinian nationalism; Palestinian sovereignty; Pan-Africanism; Panama; Pan-Americanism; Pan-Arabism; Pañcasīla; Pancasila (politics); Pan-European identity; Pan-European nationalism; Pan-Germanism; Pan-Iranism; Pan-Islamic awakening; Pan-Islamism; Pan-Latin Americanism; Pan-Slavism; Pan-Turkism; Paraguay; Parliamentary system; Participatory democracy; Participatory politics; Partition of Belgium; Pashtun people; Patent; Patriotism; Peace; Peace movement; Pensioner; Pensioners' Party; People of Ethiopia; Peronism; Personalism; Peru; Peter Kropotkin; Pim Fortuyn; Pim Fortuyn List; Platformism; Pochvennichestvo; Poles in Lithuania; Political corruption; Political freedom; Political parties of minorities; Political positions of David Cameron; Political radicalism; Political satire; Politics of Israel; Popolarismo; Popular front; Popular Socialism; Populism; Portuguese nationalism; Portuguese people; Pragmatism; Presentation program; President of the United States; President of Ukraine; Privacy; Pro-Europeanism; Progressive Christianity; Progressive Democrats of America; Progressivism; Progressivism in the United States; Protectionism; Protest; Protestantism; Protest vote; Proto-fascism; Proxy voting; Publicly funded health care; Puerto Rican independence movement; Quebec sovereignty movement; Qutbism; Racialism; Radical; Radicalism (historical); Rakhine people; Rankovićism; Redistribution of wealth; Reform; Reformism; Reform movement; Regional development; Regionalism (politics); Regions of Ethiopia; Reintegrationism; Religion; Religious denomination; Religious nationalism; Religious Zionism; Republicanism; Republicanism in Australia; Republicanism in New Zealand; Republicanism in the United Kingdom; Republicanism in the United States; Revisionism (Marxism); Revisionist Zionism; Revolutionary socialism; Rhodesia; Right-libertarianism; Rights; Right-wing politics; Right-wing populism; Royalist; Ruhollah Khomeini; Rule of law; Russia; Russian immigration to Israel in the 1990s; Russian nationalism; Russians in Estonia; Russians in Latvia; Russians in Lithuania; Russo-centrism; Russophilia; Sahrawi people; Salafi; Sami people; Sandinismo; Satire; Savoy; Scientific development concept; Scientific socialism; Scottish independence; Scottish Labour Party; Scottish national identity; Scottish nationalism; Secluarism; Secular humanism; Secularism; Secularism in Pakistan; Secularity; Self-determination;

Senior citizen; Separatism; Sephardic Haredim; Serbia; Serbian–Montenegrin unionism; Serbian nationalism; Serbian progressivism; Serbs of Montenegro; Sex-positive movement; Shia Islam; Sindhi nationalism; Single-issue politics; Sinhalese Buddhist nationalism; Slavic nationalism; Slavonia; Slovaks; Slovenian nationalism; Small government; Social change; Social conservatism; Social Conservatism; Social conservatism in the United States; Social corporatism; Social Credit; Social democracy; Social ecology; Social humanism; Social Individualism; Socialism; Socialism and Islam; Socialism of the 21st century; Socialist economics; Socialist feminism; Social Justce; Social justice; Social liberalism; Social market economy; Social republicanism; Social sphere; Solidarism; Somalia; Songun; South Africa; South Sudan; Souverainism; Sovereignty; Spain; Spanish nationalism; Spiritualism; Spirituality; Sri Lankan Tamil nationalism; Stalinism; State Peace and Development Council; State Shinto; States' rights; Statism; Statism in Shōwa Japan; Strasserism; Structuralism; Sudan; Šumadija; Sunni Islam; Sunshine Policy; Sustainable development; Sweden; Swedish-speaking Finns; Syncretic politics; Syndicalism; Syrian nationalism; Syrmia; Szeged Idea; Taiwanese nationalism; Taiwan independence; Taiwanization; Tamil nationalism; Technocracy; Temperance movement; Tertium quids; Thatcherism; Third camp; Third Position; Third Way; Third Way (centrism); Third Way (United Kingdom); Three Principles of the People; Three Represents; Tigray-Tigrinya people; Titoism; Torah; Trade union; Traditionalism; Traditionalist conservatism; Transcendental Meditation; Transnistria; Transparency (behavior); Treaty of Lisbon; Tribalism; Triple Entente; Tripuri nationalism; Trotskyism; Truth and reconciliation commission; Turkic nationalism; Turkish nationalism; Two-Nation Theory; Two-state-solution; Ujamaa; Ukrainian nationalism; Ulster loyalism; Ulster nationalism; Ultramontanism; Unification Church; Unilateral Declaration of Independence; Unionism in Ireland; Unionism in Scotland; Unionism in the United Kingdom; Union of European Federalists; Union State; Unitarisation; United Ireland; United States Congress; Urban design; Uribism; Uruguay; Uzbekistan; Valencian nationalism; Venetian nationalism; Venizelism; Vietnam; Vojvodina; Völkisch movement; Volksgemeinschaft; Voluntaryism; Walloon Movement; Warsaw; Wars of national liberation; Weapons Rights; Welfare; Welfare State; Welsh independence; Welsh nationalism; Wessex; Western conservatism; Western world; Whiggism; White nationalism;

White separatism; White supremacy; Solidarity; Women's rights; Workerism; Xenophobia; Yalkut Yosef; Youth rights; Zionism; Zulu people.

Bibliography

L. A. Adamic and E. Adar. Friends and Neighbors on the Web. *Social Networks*, 25(3):211–230, 2003.

L. A. Adamic and N. Glance. The Political Blogosphere and the 2004 US Election: Divided they Blog. In *Proceedings of the 3rd International Workshop on Link Discovery*, pages 36–43. ACM, 2005.

R. Agrawal, T. Imieliński, and A. Swami. Mining Association Rules between Sets of Items in Large Databases. In *ACM SIGMOD Record*, volume 22, pages 207–216. ACM, 1993.

R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

H. Allcott and M. Gentzkow. Social Media and Fake News in the 2016 Election. Technical report, National Bureau of Economic Research, 2017.

D. Allemang and J. Hendler. *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*. Elsevier, 2011.

C. Barrière. *Natural Language Understanding in a Semantic Web Context*. Springer, 2016.

H. Bast, B. Buchhold, and E. Haussmann. Relevance Scores for Triples from Type-Like Relations. In *SIGIR*, pages 243–252, 2015.

H. Bast, B. Buchhold, and E. Haussmann. Overview of the Triple Scoring Task at WSDM Cup 2017. In *Proceedings of the 2nd WSDM Cup at the ACM WSDM Conference on Web Search and Data Mining (WSDM Cup 17)*, 2017.

T. Berners-Lee, J. Hendler, O. Lassila, et al. The Semantic Web. *Scientific American*, 284(5):28–37, 2001.

A. Bessi, F. Petroni, M. Del Vicario, F. Zollo, A. Anagnostopoulos, A. Scala, G. Caldarelli, and W. Quattrociocchi. Homophily and Polarization in the Age of Misinformation. *The European Physical Journal Special Topics*, 225(10):2047–2059, 2016.

C. Bizer. The Emerging Web of Linked Data. *IEEE intelligent systems*, 24(5), 2009.

C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. DBpedia-a Crystallization Point for the Web of Data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154–165, 2009.

K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a Collaboratively Created Graph Database for Structuring Human Knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of data*, pages 1247–1250. ACM, 2008.

A. Bordes, J. Weston, R. Collobert, and Y. Bengio. Learning Structured Embeddings of Knowledge Bases. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, AAAI'11, pages 301–306. AAAI Press, 2011. URL http://dl.acm.org/citation.cfm?id=2900423.2900470.

A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating Embeddings for Modeling Multi-relational Data. In *Advances in Neural Information Processing Systems*, pages 2787–2795, 2013.

A. Bordes, X. Glorot, J. Weston, and Y. Bengio. A Semantic Matching Energy Function for Learning with Multi-relational Data. *Machine Learning*, 94(2):233–259, 2014.

B. Borel. *The Chicago Guide to Fact-Checking*. University of Chicago Press, 2016. ISBN 9780226290935.

S. P. Borgatti. Centrality and Network Flow. *Social Networks*, 27(1):55–71, 2005.

L. Breiman. Stacked Regressions. *Machine Learning*, 24(1):49–64, 1996.

L. Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.

D. Brickley, R. V. Guha, and B. McBride. RDF Schema 1.1. *W3C Recommendation*, 25: 2004–2014, 2014.

D. Bánky, G. Iván, and V. Grolmusz. Equal Opportunity for Low-Degree Network Nodes: A PageRank-Based Method for Protein Target Identification in Metabolic Graphs. *PLOS ONE*, 8(1): 1–7, 01 2013. doi: 10.1371/journal.pone.0054204. URL https://doi.org/10.1371/journal.pone.0054204.

E. Cambria and B. White. Jumping NLP Curves: A Review of Natural Language Processing Research [Review Article]. *IEEE Computational Intelligence Magazine*, 9(2):48–57, 2014.

A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka Jr, and T. M. Mitchell. Toward an Architecture for Never-Ending Language Learning. In *AAAI*, volume 5, page 3, 2010.

C. Castillo, M. Mendoza, and B. Poblete. Information Credibility on Twitter. In *Proceedings of the 20th International Conference on World Wide Web*, pages 675–684. ACM, 2011.

S. Cohen, J. T. Hamilton, and F. Turner. Computational Journalism. *Communications of the ACM*, 54(10):66–71, 2011a.

S. Cohen, C. Li, J. Yang, and C. Yu. Computational Journalism: A Call to Arms to Database Researchers. In *CIDR*, volume 2011, pages 148–151, 2011b.

M. Conover, J. Ratkiewicz, M. R. Francisco, B. Gonçalves, F. Menczer, and A. Flammini. Political Polarization on Twitter. *ICWSM*, 133:89–96, 2011.

L. F. Cranor and B. A. LaMacchia. Spam! *Communications of the ACM*, 41(8):74–83, 1998.

C. d'Amato, N. Fanizzi, and F. Esposito. Inductive Learning for the Semantic Web: What does it Buy? *Semantic Web*, 1(1, 2):53–59, 2010.

M. d'Aquin and E. Motta. The Epistemology of Intelligent Semantic Web Systems. *Synthesis Lectures on the Semantic Web: Theory and Technology*, 6(1):1–88, 2016.

R. Davis, H. Shrobe, and P. Szolovits. What is a Knowledge Representation? *AI Magazine*, 14(1): 17, 1993.

M. Del Vicario, A. Bessi, F. Zollo, F. Petroni, A. Scala, G. Caldarelli, H. E. Stanley, and W. Quattrociocchi. The Spreading of Misinformation Online. *Proceedings of the National Academy of Sciences*, 113(3):554–559, 2016. doi: 10.1073/pnas.1517441113. URL http://www.pnas.org/content/113/3/554.abstract.

J. Demšar. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, 7(Jan):1–30, 2006.

E. W. Dijkstra. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, 1(1): 269–271, 1959.

X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. Knowledge Vault: A Web-scale Approach to Probabilistic Knowledge Fusion. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 601–610. ACM, 2014.

X. L. Dong, E. Gabrilovich, K. Murphy, V. Dang, W. Horn, C. Lugaresi, S. Sun, and W. Zhang. Knowledge-Based Trust: Estimating the Trustworthiness of Web Sources. *Proceedings of the VLDB Endowment*, 8(9):938–949, 2015.

L. Drumond, S. Rendle, and L. Schmidt-Thieme. Predicting RDF Triples in Incomplete Knowledge Bases with Tensor Factorization. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, pages 326–331. ACM, 2012.

M. C. du Plessis, G. Niu, and M. Sugiyama. Analysis of Learning from Positive and Unlabeled Data. In *Advances in Neural Information Processing Systems*, pages 703–711, 2014.

C. Elkan and K. Noto. Learning Classifiers from Only Positive and Unlabeled Data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 213–220. ACM, 2008.

D. W. Embley, M. Hurst, D. Lopresti, and G. Nagy. Table-processing Paradigms: a Research Survey. *International Journal on Document Analysis and Recognition*, 8(2):66–86, 2006.

R. Ennals, B. Trushkowsky, and J. M. Agosta. Highlighting Disputed Claims on the Web. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 341–350, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-799-8. doi: 10.1145/1772690.1772726. URL http://doi.acm.org/10.1145/1772690.1772726.

D. Erdos and P. Miettinen. Discovering Facts with Boolean Tensor Tucker Decomposition. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, pages 1569–1572. ACM, 2013.

F. Erxleben, M. Günther, M. Krötzsch, J. Mendez, and D. Vrandečić. Introducing Wikidata to the Linked Data Web. In *International Semantic Web Conference*, pages 50–65. Springer, 2014.

E. Estrada and N. Hatano. Communicability in Complex Networks. *Physical Review E*, 77(3): 036111, 2008.

M. Färber, B. Ell, C. Menne, and A. Rettinger. A Comparative Survey of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. *Semantic Web Journal, July*, 2015.

T. Fawcett. An Introduction to ROC Analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

E. Ferrara. Manipulation and Abuse on Social Media. *SIGWEB Newsletter*, pages 4:1–4:9, Apr. 2015. ISSN 1931-1745. doi: 10.1145/2749279.2749283. URL http://doi.acm.org/10.1145/2749279.2749283.

E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini. The Rise of Social Bots. *Commun.*

ACM, 59(7):96–104, June 2016. ISSN 0001-0782. doi: 10.1145/2818717. URL http://doi.acm.org/10.1145/2818717.

D. A. Ferrucci. Introduction to "This is Watson". *IBM Journal of Research and Development*, 56 (3.4):1–1, 2012.

T. Flew, C. Spurgeon, A. Daniel, and A. Swift. The Promise of Computational Journalism. *Journalism Practice*, 6(2):157–171, 2012. URL http://dx.doi.org/10.1080/17512786.2011.616655.

L. R. Ford and D. R. Fulkerson. Maximal Flow through a Network. *Canadian Journal of Mathematics*, 8(3):399–404, 1956.

W. E. Forum. Global Risks 2013. http://www3.weforum.org/docs/WEF_GlobalRisks_Report_2013.pdf, 2013.

T. Franz, A. Schultz, S. Sizov, and S. Staab. Triplerank: Ranking Semantic Web Data by Tensor Decomposition. In *International Semantic Web Conference*, pages 213–228. Springer, 2009.

L. C. Freeman. A Set of Measures of Centrality based on Betweenness. *Sociometry*, pages 35–41, 1977.

Y. Freund and R. E. Schapire. A Decision-theoretic Generalization of On-line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

E. Gabrilovich and S. Markovitch. Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis. In *IJCAI*, volume 7, pages 1606–1611, 2007.

L. A. Galárraga, C. Teflioudi, K. Hose, and F. Suchanek. AMIE: Association Rule Mining under Incomplete Evidence in Ontological Knowledge Bases. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 413–422. ACM, 2013.

M. Gardner and T. Mitchell. Efficient and Expressive Knowledge Base Completion Using Subgraph Feature Extraction. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1488–1498, 2015.

M. Gardner, P. P. Talukdar, B. Kisiel, and T. Mitchell. Improving Learning and Inference in a Large Knowledge-base Using Latent Syntactic Cues. In *EMNLP 2013*, 2013.

L. Getoor and C. P. Diehl. Link Mining: a Survey. *ACM SIGKDD Explorations Newsletter*, 7(2): 3–12, 2005.

L. Getoor and B. Taskar. *Introduction to Statistical Relational Learning*. MIT press, 2007.

F. Goasdoué, K. Karanasos, Y. Katsis, J. Leblay, I. Manolescu, and S. Zampetakis. Fact Checking and Analyzing the Web. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pages 997–1000. ACM, 2013.

A. V. Goldberg and R. E. Tarjan. A New Approach to the Maximum-flow Problem. *Journal of the ACM (JACM)*, 35(4):921–940, 1988.

L. Goode. Social News, Citizen Journalism and Democracy. *New Media & Society*, 11(8): 1287–1305, 2009.

A. Gunawardana and G. Shani. Evaluating Recommender Systems. In *Recommender Systems Handbook*, pages 265–308. Springer, 2015.

A. Gupta, P. Kumaraguru, C. Castillo, and P. Meier. TweetCred: Real-time Credibility Assessment of Content on Twitter. In *International Conference on Social Informatics*, pages 228–243. Springer, 2014.

K. Guu, J. Miller, and P. Liang. Traversing Knowledge Graphs in Vector Space. In *EMNLP*, 2015.

S. Harispe, S. Ranwez, S. Janaqi, and J. Montmain. Semantic Similarity from Natural Language and Ontology Analysis. *Synthesis Lectures on Human Language Technologies*, 8(1):1–254, 2015.

N. Hassan, A. Sultana, Y. Wu, G. Zhang, C. Li, J. Yang, and C. Yu. Data In, Fact Out: Automated Monitoring of Facts by FactWatcher. *Proceedings VLDB Endowment*, 7(13):1557–1560, Aug. 2014. ISSN 2150-8097. doi: 10.14778/2733004.2733029. URL http://dx.doi.org/10.14778/2733004.2733029.

N. Hassan, B. Adair, J. T. Hamilton, C. Li, M. Tremayne, J. Yang, and C. Yu. The Quest to Automate Fact-Checking, 2015.

N. Hassan, G. Zhang, F. Arslan, J. Caraballo, D. Jimenez, S. Gawsane, S. Hasan, M. Joseph, A. Kulkarni, A. K. Nayak, et al. ClaimBuster: The First-ever End-to-end Fact-checking System. *Proceedings of the VLDB Endowment*, 10(7), 2017.

T. H. Haveliwala. Topic-Sensitive Pagerank. In *Proceedings of the 11th International Conference on World Wide Web*, WWW '02, pages 517–526, New York, NY, USA, 2002. ACM. ISBN 1-58113-449-5. doi: 10.1145/511446.511513. URL http://doi.acm.org/10.1145/511446.511513.

S. Heindorf, M. Potthast, H. Bast, B. Buchhold, and E. Haussmann. WSDM Cup 2017: Vandalism Detection and Triple Scoring. In *WSDM*. ACM, 2017.

P. Hernon. Disinformation and Misinformation through the Internet: Findings of an Exploratory Study. *Government Information Quarterly*, 12(2):133–139, 1995.

P. Heymann, G. Koutrika, and H. Garcia-Molina. Fighting Spam on Social Web Sites: A Survey of Approaches and Future Challenges. *IEEE Internet Computing*, 11(6), 2007.

J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum. YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia. *Artificial Intelligence*, 194:28–61, 2013.

T. N. Jagatic, N. A. Johnson, M. Jakobsson, and F. Menczer. Social Phishing. *Communications of the ACM*, 50(10):94–100, 2007.

A. Jain and P. Pantel. Factrank: Random Walks on a Web of Facts. In *Proceedings of the 23rd*

International Conference on Computational Linguistics, pages 501–509. Association for Computational Linguistics, 2010.

S. Jain, M. White, and P. Radivojac. Recovering True Classifier Performance in Positive-Unlabeled Learning. In *AAAI*, pages 2066–2072, 2017.

G. Jeh and J. Widom. SimRank: a Measure of Structural-context Similarity. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Dining*, pages 538–543. ACM, 2002.

R. Jenatton, N. L. Roux, A. Bordes, and G. R. Obozinski. A Latent Factor Model for Highly Multi-relational Data. In *Advances in Neural Information Processing Systems*, pages 3167–3175, 2012.

J. J. Jiang and D. W. Conrath. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. *CoRR*, cmp-lg/9709008, 1997.

X. Jiang, V. Tresp, Y. Huang, and M. Nickel. Link Prediction in Multi-relational Graphs Using Additive Models. In *Proceedings of the 2012 International Conference on Semantic Technologies Meet Recommender Systems & Big Data-Volume 919*, pages 1–12. CEUR-WS. org, 2012.

T. Kamada and S. Kawai. An Algorithm for Drawing General Undirected Graphs. *Information Processing Letters*, 31(1):7–15, 1989.

L. Katz. A New Status Index Derived from Sociometric Analysis. *Psychometrika*, 18(1):39–43, 1953.

T. Khot, S. Natarajan, K. Kersting, and J. Shavlik. Gradient-based Boosting for Statistical Relational Learning: the Markov Logic Network and Missing Data Cases. *Machine Learning*, 100(1):75, 2015.

G. Klyne and J. J. Carroll. Resource Description Framework (RDF): Concepts and Abstract Syntax.

W3C Recommendation, 2004. *World Wide Web Consortium, http://w3c. org/TR/rdf-concepts*, 2004.

D. Kontokostas, P. Westphal, S. Auer, S. Hellmann, J. Lehmann, R. Cornelissen, and A. Zaveri. Test-Driven Evaluation of Linked Data Quality. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 747–758. ACM, 2014.

S. Kumar, R. West, and J. Leskovec. Disinformation on the Web: Impact, Characteristics, and Detection of Wikipedia Hoaxes. In *Proceedings of the 25th International Conference on World Wide Web*, pages 591–602. International World Wide Web Conferences Steering Committee, 2016.

M. Kusumoto, T. Maehara, and K.-i. Kawarabayashi. Scalable Similarity Search for SimRank. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, pages 325–336. ACM, 2014.

N. Landwehr, K. Kersting, and L. De Raedt. NFOIL: Integrating Naïve Bayes and FOIL. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI-05)*, pages 795–800, 2005.

N. Lao and W. W. Cohen. Fast Query Execution for Retrieval Models based on Path-Constrained Random Walks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 881–888. ACM, 2010a.

N. Lao and W. W. Cohen. Relational Retrieval Using a Combination of Path-constrained Random Walks. *Machine Learning*, 81(1):53–67, 2010b.

N. Lao, T. Mitchell, and W. W. Cohen. Random Walk Inference and Learning in a Large Scale Knowledge Base. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 529–539. Association for Computational Linguistics, 2011.

J. Lehmann. DL-learner: Learning Concepts in Description Logics. *Journal of Machine Learning Research*, 10(Nov):2639–2642, 2009.

E. A. Leicht, P. Holme, and M. E. Newman. Vertex Similarity in Networks. *Physical Review E*, 73 (2):026120, 2006.

D. B. Lenat. CYC: a Large-scale Investment in Knowledge Infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.

D. Liben-Nowell and J. Kleinberg. The Link-prediction Problem for Social Networks. *Journal of the American society for Information Science and Technology*, 58(7):1019–1031, 2007.

D. Lin. An Information-Theoretic Definition of Similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 296–304, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 1-55860-556-8. URL http://dl.acm.org/citation.cfm?id=645527.657297.

Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *AAAI*, pages 2181–2187, 2015.

W. Liu and L. Lü. Link Prediction Based on Local Random Walk. *EPL (Europhysics Letters)*, 89 (5):58007, 2010.

V. Lopez, M. Fernández, E. Motta, and N. Stieler. PowerAqua: Supporting Users in Querying and Exploring the Semantic Web. *Semantic Web*, 3(3):249–265, 2012.

L. Lü and T. Zhou. Link Prediction in Complex Networks: A Survey. *Physica A: Statistical Mechanics and its Applications*, 390(6):1150–1170, 2011.

L. Lü, C.-H. Jin, and T. Zhou. Similarity Index Based on Local Paths for Link Prediction of Complex Networks. *Physical Review E*, 80(4):046122, 2009.

W. Lu, J. Janssen, E. Milios, and N. Japkowicz. Node Similarity in Networked Information Spaces. In *Proceedings of the 2001 Conference of the Centre for Advanced Studies on Collaborative research*, page 11. IBM Press, 2001.

A. G. Maguitman, F. Menczer, F. Erdinc, H. Roinestad, and A. Vespignani. Algorithmic Computation and Approximation of Semantic Similarity. *World Wide Web*, 9(4):431–456, 2006.

C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA, 1999. ISBN 0-262-13360-1.

B. Markines and F. Menczer. A Scalable, Collaborative Similarity Measure for Social Annotation Systems. In *Proceedings of the 20th ACM Conference on Hypertext and Hypermedia*, pages 347–348. ACM, 2009.

D. L. McGuinness, F. Van Harmelen, et al. OWL Web Ontology Language Overview. *W3C Recommendation*, 10(10):2004, 2004.

G. A. Miller. WordNet: a Lexical Database for English. *Communications of the ACM*, 38(11): 39–41, 1995.

D. Milne and I. H. Witten. An Effective, Low-Cost Measure of Semantic Relatedness Obtained from Wikipedia Links. In *In Proceedings of AAAI 2008*, 2008.

T. M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997. ISBN 0070428077, 9780070428072.

T. Mitra, G. P. Wright, and E. Gilbert. A Parsimonious Language Model of Social Media Credibility Across Disparate Events. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, pages 126–145. ACM, 2017.

S. Muggleton, R. Otero, and A. Tamaddoni-Nezhad. *Inductive Logic Programming*, volume 38. Springer, 1992.

S. Natarajan, T. Khot, K. Kersting, B. Gutmann, and J. Shavlik. Gradient-based Boosting for Statistical Relational Learning: The Relational Dependency Network Case. *Machine Learning*, 86(1):25–56, 2012.

A. Neelakantan, B. Roth, and A. McCallum. Compositional Vector Space Models for Knowledge Base Inference. In *2015 AAAI Spring Symposium Series*, 2015.

P. Nemenyi. *Distribution-free Multiple Comparisons.* PhD thesis, Princeton University, Princeton, New Jersey, 1963.

J. Neville and D. Jensen. Relational Dependency Networks. *Journal of Machine Learning Research*, 8(Mar):653–692, 2007.

M. Newman. *Networks: An Introduction*. Oxford University Press, Inc., New York, NY, USA, 2010. ISBN 0199206651, 9780199206650.

M. E. Newman. The Structure and Function of Complex Networks. *SIAM Review*, 45(2):167–256, 2003.

M. Nickel and D. Kiela. Poincaré Embeddings for Learning Hierarchical Representations. *CoRR*, abs/1705.08039, 2017. URL http://arxiv.org/abs/1705.08039.

M. Nickel, V. Tresp, and H.-P. Kriegel. A Three-way Model for Collective Learning on Multi-relational Data. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 809–816, 2011.

M. Nickel, X. Jiang, and V. Tresp. Reducing the Rank in Relational Factorization Models by Including Observable Patterns. In *Advances in Neural Information Processing Systems*, pages 1179–1187, 2014.

M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich. A Review of Relational Machine Learning for Knowledge Graphs. *Proceedings of the IEEE*, 104(1):11–33, Jan 2016a. ISSN 0018-9219. doi: 10.1109/JPROC.2015.2483592.

M. Nickel, L. Rosasco, and T. Poggio. Holographic Embeddings of Knowledge Graphs. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 1955–1961. AAAI Press, 2016b. URL http://dl.acm.org/citation.cfm?id=3016100.3016172.

D. Nikolov, D. F. Oliveira, A. Flammini, and F. Menczer. Measuring Online Social Bubbles. *PeerJ Computer Science*, 1:e38, 2015.

F. Niu, C. Ré, A. Doan, and J. Shavlik. Tuffy: Scaling Up Statistical Inference in Markov Logic Networks using an RDBMS. *Proceedings of the VLDB Endowment*, 4(6):373–384, 2011.

A. Ohlheiser. This is how Facebook's Fake-news Writers Make Money. Washington Post, Nov 2016. http://wpo.st/tNyR2.

H. Paulheim. Knowledge Graph Refinement: A Survey of Approaches and Evaluation Methods. *Semantic Web*, 8(3):489–508, 2017.

H. Paulheim and C. Bizer. Improving the Quality of Linked Data using Statistical Distributions. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 10(2):63–86, 2014.

K. Pearson. LIII. on Lines and Planes of Closest Fit to Systems of Points in Space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

F. Pedregosa-Izquierdo. *Feature extraction and supervised learning on fMRI : from practice to theory*. Theses, Université Pierre et Marie Curie - Paris VI, Feb. 2015. URL https://tel.archives-ouvertes.fr/tel-01100921.

K. T. Poole and H. Rosenthal. Ideology and Congress: A Political Economic History of Roll Call Voting, 2007.

J. R. Quinlan. Learning Logical Definitions from Relations. *Machine Learning*, 5(3):239–266, 1990.

R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and Application of a Metric on Semantic Nets. *IEEE Transactions on systems, man, and cybernetics*, 19(1):17–30, 1989.

J. Ratkiewicz, M. Conover, M. Meiss, B. Gonçalves, S. Patil, A. Flammini, and F. Menczer. Truthy: Mapping the Spread of Astroturf in Microblog Streams. In *Proceedings of the 20th International Conference Companion on World Wide Web*, pages 249–252. ACM, 2011a.

J. Ratkiewicz, M. Conover, M. Meiss, B. Gonçalves, A. Flammini, and F. Menczer. Detecting and tracking political abuse in social media. In *Proceedings 5th International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2011b. URL http://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/view/2850.

J. D. Rennie and N. Srebro. Loss Functions for Preference Levels: Regression with Discrete Ordered Labels. In *Proceedings of the IJCAI Multidisciplinary Workshop on Advances in Preference Handling*, pages 180–186. Kluwer Norwell, MA, 2005.

P. Resnick, S. Carton, S. Park, Y. Shen, and N. Zeffer. RumorLens: A System for Analyzing the Impact of Rumors and Corrections in Social Media. In *Proceedings Computational Journalism Conference*, 2014.

P. Resnik. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, pages 448–453, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1-55860-363-8, 978-1-558-60363-9. URL http://dl.acm.org/citation.cfm?id=1625855.1625914.

M. Richardson and P. Domingos. Markov Logic Networks. *Machine Learning*, 62(1-2):107–136, 2006.

S. Riedel, L. Yao, B. M. Marlin, and A. McCallum. Relation Extraction with Matrix Factorization and Universal Schemas. In *Joint Human Language Technology Conference/Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL '13)*, June 2013.

S. Y. Rieh and D. R. Danielson. Credibility: A Multidisciplinary Framework. *Annual Review of Information Science and Technology*, 41(1):307–364, 2007.

N. Ruchansky, S. Seo, and Y. Liu. CSI: A Hybrid Deep Model for Fake News. *CoRR*, abs/1703.06959, 2017. URL http://arxiv.org/abs/1703.06959.

S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2 edition, 2003. ISBN 0137903952.

S. Sarawagi et al. Information Extraction. *Foundations and Trends® in Databases*, 1(3):261–377, 2008.

B. Shi and T. Weninger. Discriminative Predicate Path Mining for Fact Checking in Knowledge Graphs. *Knowledge-Based Systems*, 104:123–133, 2016.

B. Shi and T. Weninger. ProjE: Embedding Projection for Knowledge Graph Completion. *CoRR*, abs/1611.05425, 2017.

C. Shi, X. Kong, Y. Huang, S. Y. Philip, and B. Wu. HeteSim: A General Framework for Relevance Measure in Heterogeneous Networks. *IEEE Transactions on Knowledge and Data Engineering*, 26(10):2479–2492, 2014.

S. Shin, S. E. Ahnert, and J. Park. Ranking Competitors using Degree-neutralized Random Walks. *PlOS ONE*, 9(12):e113685, 2014.

T. Simas and L. M. Rocha. Distance Closures on Complex Networks. *Network Science*, 3(02): 227–268, 2015.

A. P. Singh and G. J. Gordon. Relational Learning Via Collective Matrix Factorization. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 650–658. ACM, 2008.

K. Stephenson and M. Zelen. Rethinking Centrality: Methods and Examples. *Social Networks*, 11 (1):1–37, 1989.

V. Subrahmanian, A. Azaria, S. Durst, V. Kagan, A. Galstyan, K. Lerman, L. Zhu, E. Ferrara, A. Flammini, and F. Menczer. The DARPA Twitter Bot Challenge. *Computer*, 49(6):38–46, 2016.

A. Sultana, N. Hassan, C. Li, J. Yang, and C. Yu. Incremental Discovery of Prominent Situational Facts. In *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*, pages 112–123. IEEE, 2014.

Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu. Pathsim: Meta Path-based Top-k Similarity Search in Heterogeneous Information Networks. *Proceedings of the VLDB Endowment*, 4(11):992–1003, 2011.

I. Sutskever, J. B. Tenenbaum, and R. R. Salakhutdinov. Modelling Relational Data Using Bayesian Clustered Tensor Factorization. In *Advances in Neural Information Processing Systems*, pages 1821–1828, 2009.

E. Tacchini, G. Ballarin, M. L. D. Vedova, S. Moret, and L. de Alfaro. Some Like it Hoax: Automated Fake News Detection in Social Networks. *CoRR*, abs/1704.07506, 2017. URL http://arxiv.org/abs/1704.07506.

A. Vlachos and S. Riedel. Fact Checking: Task Definition and Dataset Construction. *ACL 2014*, page 18, 2014.

B. Walenz, Y. W. Wu, S. A. Song, E. Sonmez, E. Wu, K. Wu, P. K. Agarwal, J. Yang, N. Hassan, A. Sultana, et al. Finding, Monitoring, and Checking Claims Computationally Based on Structured Data. In *Computation + Journalism Symposium*, 2014.

Z. Wang, J. Zhang, J. Feng, and Z. Chen. Knowledge Graph Embedding by Translating on Hyperplanes. In *AAAI*, pages 1112–1119. Citeseer, 2014.

Y. Wu, P. K. Agarwal, C. Li, J. Yang, and C. Yu. On One of the few Objects. In *Proceedings of the*

18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1487–1495. ACM, 2012.

Y. Wu, J. Gao, P. K. Agarwal, and J. Yang. Finding Diverse, High-value Representatives on a Surface of Answers. *Proceedings VLDB Endowment*, 10(7):793–804, Mar. 2017a. ISSN 2150-8097. doi: 10.14778/3067421.3067428. URL https://doi.org/10.14778/3067421.3067428.

Y. Wu, T. Mu, and J. Y. Goulermas. Translating on Pairwise Entity Space for Knowledge Graph Embedding. *Neurocomputing*, 2017b.

W. Xiong, T. Hoang, and W. Y. Wang. DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning. *arXiv preprint arXiv:1707.06690*, 2017.

Z. Xu, C. Pu, and J. Yang. Link Prediction based on Path Entropy. *Physica A: Statistical Mechanics and its Applications*, 456:294 – 301, 2016a. ISSN 0378-4371. doi: http://dx.doi.org/10.1016/j.physa.2016.03.091. URL http://www.sciencedirect.com/science/article/pii/S0378437116300899.

Z. Xu, C. Pu, and J. Yang. Link Prediction based on Path Entropy. *Physica A: Statistical Mechanics and its Applications*, 456:294–301, 2016b.

E. Yeh, D. Ramage, C. D. Manning, E. Agirre, and A. Soroa. WikiWalk: Random Walks on Wikipedia for Semantic Relatedness. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing*, pages 41–49. Association for Computational Linguistics, 2009.

A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann, and S. Auer. Quality Assessment for Linked Data: A Survey. *Semantic Web*, 7(1):63–93, 2016.

C. Zhang, M. Zhou, X. Han, Z. Hu, and Y. Ji. Knowledge Graph Embedding for Hyper-relational Data. *Tsinghua Science and Technology*, 22(02):185–197, 2017.

G. Zhang, X. Jiang, P. Luo, M. Wang, and C. Li. Discovering General Prominent Streaks in Sequence Data. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 8(2):9, 2014.

A. Zubiaga, A. Aker, K. Bontcheva, M. Liakata, and R. Procter. Detection and Resolution of Rumours in Social Media: A Survey. *CoRR*, abs/1704.00656, 2017. URL http://arxiv.org/abs/1704.00656.

Prashant Shiralkar

CONTACT

919 E. 10th Street, Bloomington, IN 47408
Email: pshiralk@indiana.edu
Web: https://sites.google.com/site/shiralkarprashant/

EDUCATION

Indiana University, Bloomington, IN — Sep 2017
Ph.D. in Computer Science
Minor: Statistics, GPA: 3.85/4.0
Advisor: Filippo Menczer

University of Iowa, Iowa City, IA — May 2010
Master of Computer Science
Focus: Software Engineering, GPA: 3.73/4.0

Nirma University, Gujarat, India — May 2008
Bachelor of Technology in Computer Engineering
Major: Computer Science, GPA: 7.78/10

RESEARCH INTERESTS

Network Science, Machine Learning, Statistical Relational Learning, Semantic Web, Computational Social Science, Natural Language Processing, Information Extraction, Information Retrieval, Web Science

PUBLICATIONS

Finding Streams in Knowledge Graphs to Support Fact Checking. P. Shiralkar, G. Ciampaglia, A. Flammini, F. Menczer. Accepted for *Proc. of the Intl. Conf. on Data Mining*, 2017.

RelSifter: Scoring Triples from Type-Like Relations. P. Shiralkar, M. Avram, G. Ciampaglia, F. Menczer, A. Flammini. Accepted for *Proc. of the WSDM Cup 2017*, 2017.

Computational Fact Checking from Knowledge Networks. G. Ciampaglia, P. Shiralkar, L. Rocha, J. Bollen, F. Menczer, A. Flammini. *PLOS ONE* 10(6):e0128193, 2015.

OSoMe: the IUNI observatory on social media. C. Davis, G. Ciampaglia, L. Aiello, K. Chung, M. Conover, E. Ferrara, A. Flammini, X. Gao, B. Gonçalves, P. Shiralkar, F. Menczer and others. *PeerJ Computer Science*, 2:e87, 2016.

The DARPA Twitter Bot Challenge. VS Subrahmanian, A. Azaria, S. Durst, V. Kagan, A. Galstyan, K. Lerman, L. Zhu, E. Ferrara, A. Flammini, F. Menczer, P. Shiralkar and others. *IEEE Computer Society*, 49(6):38–46, 2016.

RESEARCH EXPERIENCE

Computational fact checking from knowledge graphs Sep 2013 - May 2017
Designed network science and machine learning methods to assess veracity of claims, and surface useful patterns and contextual facts to support fact checking.

Clustering users on Twitter Jan 2014 - Dec 2014
Performed an exploratory study using unsupervised learning approaches with the goal to identify natural groupings of users on Twitter.

Social bot detection and estimation Jun 2014 - Aug 2014
Applied supervised learning approaches to classify a user on Twitter as bot or not. Investigated statistical techniques to estimate the population of social bot accounts on the platform.

Data Sciences Summer Institute (DSSI) May 2012 - Jun 2012
Built an "Event Search Engine" to identify and search real-world events for a news related query. This project was funded by the U.S. Department of Homeland Security and Yahoo!.

INDUSTRY EXPERIENCE

Amazon Inc., Seattle, WA Jun 2016 - Aug 2016
Machine Learning Intern

- Analyzed the role of taxonomy structure on Amazon.com in enabling discovery of products for customers.
- Built a machine learning model to predict the impact of changes to taxonomy structure on user experience.

Philips Research, New York, NY May 2015 - Aug 2015
Research Intern

- Contributed to the design and development of ranking component of an internal product search engine.
- Integrated product information from disparate sources to enable search and discovery.

Transamerica Capital Management, Cedar Rapids, IA Apr 2011 - May 2012
Programmer Analyst

- Built an analytic tool to hedge annuity rider funds worth $22M. Presented a proof of concept to evaluate Spring Batch framework for this project.
- Performed ETL operations on 5M annuity policies using Spring Batch. Wrote back-end code in SQL Server to report statistics of batch loads to actuaries.

IT-Enterprise Services, U. of Iowa, Iowa City, IA May 2010 - Mar 2011
Application Developer

- Developed a web interface for International Studies program coordinators to perform an automated comparison of current and future courses to aid their course planning activities.
- Enhanced and supported freshmen activity-tracking web applications at the University of Iowa.

Login Softech Pvt. Ltd, Ahmedabad, India Jan 2008 - May 2008
Software Engineering Intern

- Designed and developed a bus e-ticketing software for travel agents and customers of the state of Gujarat, India.

TEACHING EXPERIENCE

Associate Instructor, Indiana University Fall 2016
Tutored a Machine Learning class of 100 students. Graded homeworks and exams.

Associate Instructor, Indiana University Fall 2012, Spring 2013
Tutored 35 students on programming constructs in a Python programming lab.

TECHNICAL SKILLS

Languages: Python (Numpy, Scipy, Pandas), Cython, R, MATLAB, Java, SQL, Javascript
Frameworks: Hadoop, HBase, Django, J2EE (web)
Miscellaneous: RStudio, Eclipse, LaTeX, MS PowerPoint, Git
Previous or lightly used: C, C++, JSP, Oracle 11g, MS SQL Server, Stripes

CERTIFICATION & AWARDS

Outstanding Project Award by Yahoo! at DSSI, Univ. of Illinois	Jun 2012
Oracle Database SQL Certified Expert (1Z0-047)	Oct 2010
Sun Certified Web Component Developer (SCWCD), Java EE 5	Jun 2010
Sun Certified Programmer (SCJP), Java SE 6	Apr 2010

ACADEMIA SERVICE

Reviewer of WWW 2017, ACM BuildSys 2015, DYAD 2014.