Computational Fact-Checking

Joshua Chen

Seminar Natural Language Processing, Universität Bonn, Summer 2017

Contents

1 Introduction

Fact-checking is the task of distinguishing and verifying the truth of factual claims, in particular those that are propagated to a large public audience. Examples of such claims include reports found in news media, statements made in political or other public debates, or the content of social media posts. Also of particular concern in recent times is the rise of “fake news”—journalistic content designed to mislead or otherwise dishonestly influence the general public.

Traditionally, fact-checking has been the job of journalists, whose responsibility it is to report on the truthfulness of public figures’ claims, and to ensure that their sources are reliable and accurate. Today this task has become harder due to two challenges: first of which is the sheer amount, diversity and rate of information produced in our “big data” age. Unbacked claims are very easy to make and typically hard to check, since one would need to go through various source materials to find evidence that either confirms or refutes the claim. This only becomes harder with more information to check and more of it being produced every hour.

Second, social media has now made it possible for ordinary people to have the same kind of reach that only broadcast media and journalists had before—now anyone can create content that is consumed by potentially hundreds of thousands of people. Without methods to cope with this new channel of information, misinformation can spread much wider and negatively affect much larger groups of people.

These issues motivate the field of *computational fact-checking*, which we define as the application of machine learning, natural language processing, and statistical techniques to automate the fact-checking process. Full automation of this process is a complex undertaking with many different aspects and subtasks, each of which provides ample research opportunities and challenges in its own right. In this report however we focus on the task of actually fact-checking claims, and consider two approaches:

1. Automated checking of relational claims using knowledge graphs.
2. A more general claim-checking framework, including methods of finding counterarguments and reverse-engineering vague claims.

2 Knowledge graph based fact-checking

In this section we consider automatic fact-checking of **relational claims**, which are statements that relate two *entities* via some *predicate*. More concretely, a relational claim is one that can be represented as a subject-predicate-object **relational triple** (s, p, o), e.g. (*"Barack Obama"*, *"is a"*, *"Muslim"*), or (*"the Danube"*, *"flows through"*, *"Serbia"*).

2.1 Knowledge graphs

We assume a knowledge base in the form of a collection $\mathcal{C}$ of relational triples. (Examples of such collections are given by the DBpedia RDF datasets [3], or the Semantic MEDLINE Database (SemMedDB) [6].)

Such a collection gives rise to a **knowledge graph** $\mathcal{G}$, in which every subject or object entity in $\mathcal{C}$ corresponds to a vertex of $\mathcal{G}$, and every predicate of a triple in $\mathcal{C}$ to an edge between the corresponding subject and object vertices. Depending on the specific application $\mathcal{G}$ may be directed or undirected, and may also have extra information attached to its vertices or edges.

A key idea underlying knowledge graph fact-checking algorithms is that fact-checking can be viewed as a **link prediction problem** over $\mathcal{G}$. This means the following: assume that the graph $\mathcal{G} = (V, E)$ is correct but incomplete, i.e. there is a "true" edge set $\widetilde{E}$ of $\mathcal{G}$, of which $E \subseteq \widetilde{E}$ is a subset. This models a situation of incomplete knowledge, where edges $e \in \widetilde{E} \setminus E$ correspond to claim triples (s, p, o) that are true but missing from our knowledge base $\mathcal{C}$. Link prediction is the problem of using structural properties of $\mathcal{G}$ to approximate $\widetilde{E}$. If, given some claim $c = (s, p, o)$, we predict its corresponding edge e to be in $\widetilde{E}$, then we predict c to be true. Of course, if e is in E then c is already in $\mathcal{C}$ and is known to be true. So the interesting case is to predict when a candidate edge $e \notin E$ is in $\widetilde{E}$.

2.2 Truth as proximity in knowledge graphs

Here we present and critique the method of Ciampaglia et al. [2].

Given $\mathcal{C}$ as before, we define the knowledge graph $\mathcal{G} = (V, E)$ to be the undirected graph with vertex set

$$V = \{v \mid v \text{ is a subject or object entity}\}$$

and edge set

$$E = \{\{s, o\} \mid (s, p, o) \in \mathcal{C} \text{ for some predicate } p\}.$$

We label vertices with their corresponding entity names, but forget the predicate information used to form the edges, so that $\mathcal{G}$ is a vertex- but not edge-labelled graph. Note that this "throwing away" of the valuable predicate information results in a much less precise representation of our knowledge base.

We model the truth of a claim (s, p, o) by a *proximity measure* on the entities s and o in $\mathcal{G}$—the intuition is that vertices that are "closer" together under such a measure are more strongly related, and hence it is more likely that the claimed relation between them is true. In the setting of link prediction, the closer the vertices, the stronger our prediction that there is an edge between them in $\widetilde{E}$.

2.2.1 Semantic proximity and truth assessment

Let $P = v_1 v_2 \cdots v_n$ be a path between endpoints v_1 and v_n in $\mathcal{G}$. We define the **semantic proximity of** v_1 **and** v_n **along** P by

$$W(P) = \left(1 + \sum_{i=2}^{n-1} \log \delta(v_i)\right)^{-1}$$

where $\delta(v)$ is the degree of v in $\mathcal{G}$.

Paths P between v_1 and v_n that pass through vertices of lower degree yield higher semantic proximity scores $W(P)$. This captures the heuristic of *specificity*: vertices with larger degree, being related to more entities, are more general and thus provide less support for the particular given claim.

By considering the maximum semantic proximity along all paths between s and o we define the **truth score** $\tau(c)$ of a claim $c = (s, p, o)$:

$$\tau(c) = \max\{W(P) \mid P \text{ is a path between } s \text{ and } o \text{ in } \mathcal{G}\}.$$

Note that by definition $\tau(c) \in [0, 1]$, and if $\{s, o\}$ is an edge in $\mathcal{G}$ then $\tau(c) = 1$.

Then we fact-check a given claim c by computing its truth score $\tau(c)$. The higher the truth score, the more confident we are that c is true. Actually computing $\tau(c)$ involves finding an optimal path between the subject and object vertices. The authors claim this can be transformed into a metric closure problem on $\mathcal{G}$, but we do not discuss this here.

2.2.2 Experimental results

We present the results obtained by the authors by calculating the truth scores of relational claims from four datasets, of the form (`Director`, *"directed"*, `Oscar-winning film`), (`U.S. President`, *"was married to"*, `Spouse`), (`U.S. state capital`, *"is the capital of"*, `U.S. state`), and (`City`, *"is the capital of"*, `Country`). Relational triples were obtained from the Wikipedia infobox datasets extracted by DBpedia, and edges between related subject-object pairs were removed from $\mathcal{G}$ to prevent the fact-checking problem from becoming trivial.

Figure 1 shows the confusion matrices consisting of the truth scores for each possible subject-object pair of a particular dataset. Higher-intensity colors indicate higher truth scores, and the diagonals correspond to correct subject-object pairs.

Besides these results, the authors also consider a few other datasets, and compute ROC curves and other statistical scores showing that their method consistently scores correct claims higher than incorrect ones.

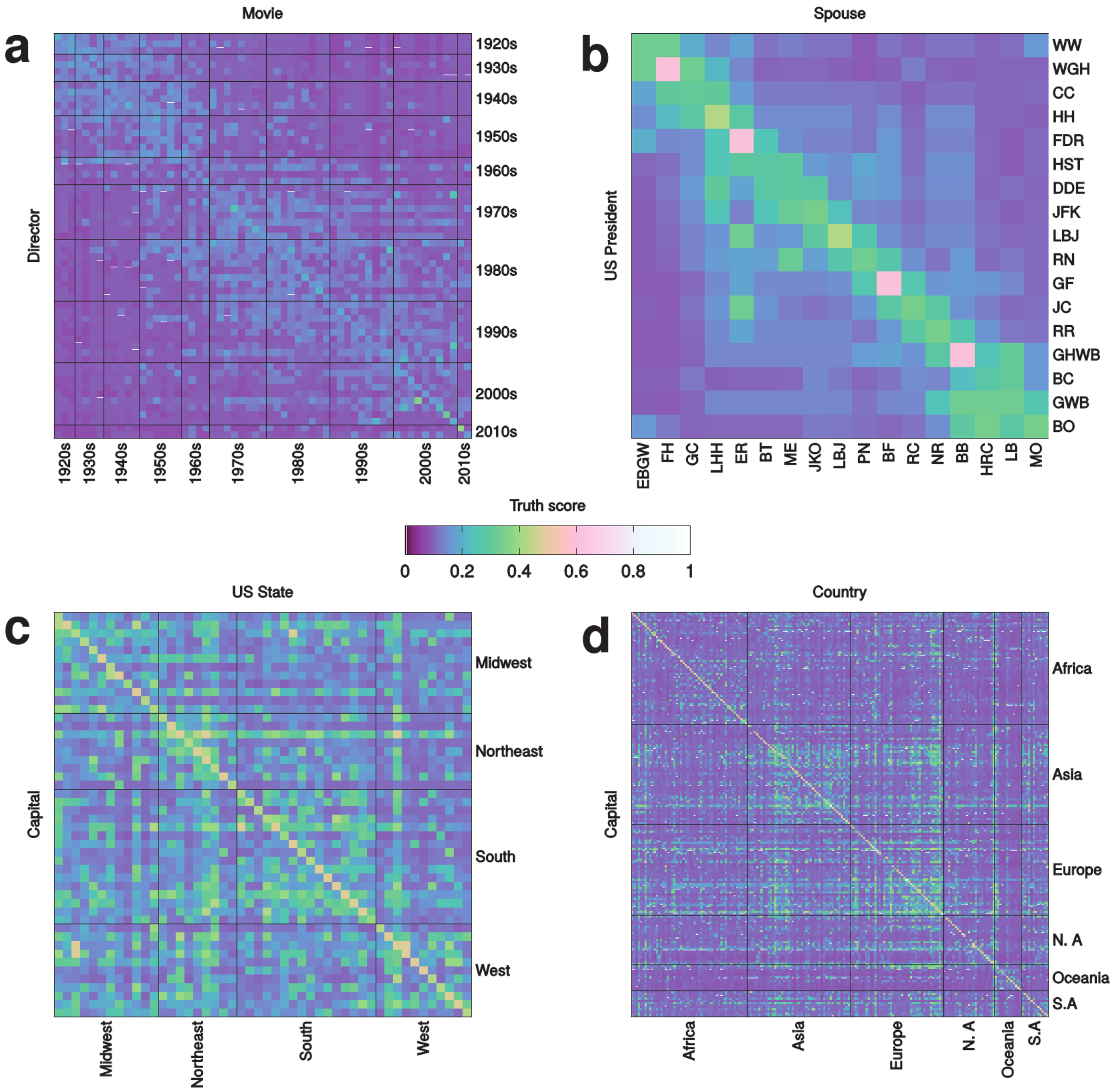

Figure 1: Plots taken from [2]. Rows denote subjects and columns denote objects. Diagonals correspond to correct subject-object pairs. (a) Oscar-winning films and their directors, grouped by decade. (b) U.S. presidents and spouses. (c) U.S. states and their capital cities. (d) Countries and their capital cities.

2.2.3 Discussion

While the results above appear promising, the model we have presented is severely limited by its failure to use the predicate information present in $\mathcal{C}$. This can cause problems when fact-checking distinct relational claims between the same subject and object, since the presence of just one triple $(s, p, o) \in \mathcal{C}$ causes *any* claim triple (s, p', o) to be assigned a truth score of 1, so that for example (*"Pierre Curie"*, *"husband of"*, *"Marie Curie"*) being in $\mathcal{C}$ causes *"Pierre Curie is the brother of Marie Curie"* and *"Pierre Curie is the son of Marie Curie"* to both be evaluated as true claims. Thus in its current form the model is only suitable for very simple claims, or for situations where the predicate is fixed and the domain is constrained, as in the experimental datasets in Section 2.2.2.

2.3 Discriminative path mining

Shi and Weninger [7] introduce a more sophisticated knowledge graph fact-checking scheme, which in addition to the relational triples also uses an ontology defined on the entities in $\mathcal{C}$. Due to a lack of space we will briefly sketch the approach here without going into too much technical detail.

2.3.1 Ontology-enriched knowledge graphs

Let $\mathcal{G}$ be a directed knowledge graph arising from $\mathcal{C}$ in the usual way, where we label vertices and edges with their respective entity names and predicates, and orient edges from subject to object. We furthermore assume an ontology $\mathcal{O}$ on the set of all entities in $\mathcal{C}$, and additionally associate to each vertex v its label set in $\mathcal{O}$.

2.3.2 Discriminative paths

Given a claim (s, p, o) between entities s and o, our aim is to understand the predicate p by mining *discriminative paths*, which are paths between vertices having the same ontology as s and o that alternatively describe the relation p. The idea is that by passing to the ontology of the endpoints we achieve *generality*, e.g. we can better fact-check a claim (*"New York city"*, *"capital of"*, *"New York"*) if we first understand what the triple (`U.S. city`, *"capital of"*, `U.S. state`) entails.

More precisely, let x and y be labels from the ontology $\mathcal{O}$. A **discriminative path** of length k is a sequence of k edges $e_1, \ldots, e_k$ that form an undirected path in $\mathcal{G}$ between a vertex with label x and a vertex with label y (that is, we allow ourselves to "go backwards" along a directed edge).

Now given a claim (s, p, o), consider the set

$$T := \{(s', o') \mid s' \text{ and } o' \text{ have the same ontology labels as } s \text{ and } o \text{ respectively}\}.$$

This set can be partitioned into the set T^+ consisting of those pairs for which there is an edge (s', o') labeled p, and the set T^- consisting of those pairs for which there is no such edge. We then wish to find "good" discriminative paths, i.e. those that occur often among pairs in T^+ but rarely among pairs in T^-.

2.3.3 Further work

The authors discuss in much more detail an algorithm that mines good discriminative paths, and thereafter uses such paths to predict if a given claim (s, p, o) should be present as an edge in $\mathcal{G}$.

They also run experiments to fact-check relational claims over various DBpedia knowledge bases, and find that their method outperforms existing algorithms.

3 More general fact-checking

In Wu et al. [8] a more general framework is presented, which allows us to not only fact-check statements of various forms, but also to find counterarguments to dubious claims, and to reverse-engineer vague claims in order to make them more precise.

The following claim is taken as a running example throughout this section.

Example (Giuliani's adoption increase claim [4, 8])**.** *In a 2007 Republican presidential candidate debate, former New York mayor Rudy Giuliani claimed that "adoptions went up 65 to 70 percent" during his tenure as mayor. The precise comparison (clarified later by the campaign) is of the total adoptions between the two fiscal year periods beginning 1990–1995 and 1996–2001. Giuliani was mayor from 1994–2001.*

We will use this example later to demonstrate reverse-engineering (the claim did not initially state the precise comparison window) as well as counterargument finding (there was already an existing trend of increase in adoptions before 1996, and in fact after 1998 they decreased until total adoptions at the end of Giuliani's term were only 17 percent higher than at the beginning [4]).

3.1 Basic framework

Given a factual claim we identify *parameters* in it that we can vary in order to change the *result* of the stated claim. For example in the claim "unemployment decreased by 20 percent between 2012 and 2016", the parameter is the time period under consideration, and the result is the percentage decrease in unemployment. Parameters are often quantitative or date-based, though one can also consider relational claims, e.g. "Obama is a Muslim" can be considered to have a single parameter varying over all world religions.

Given a claim, we consider what happens to its result when we perturb its parameters to yield different claims of the same form.

More precisely, assume we have access to a function $q\colon \mathcal{P} \to \mathcal{R}$, known as a **parametrized query template**, mapping the **parameter space** $\mathcal{P}$ of a claim to the **result space** $\mathcal{R}$ consisting of all possible query results over a given knowledge database. A **claim** of type q is given by a triple (q, p, r) where $p \in \mathcal{P}$ are the parameter values of that particular claim and $r \in \mathcal{R}$ is the claimed result.

Example. *Assume we have access to a database of yearly adoptions numbers via a function* adopt*, where* $\operatorname{adopt}(y)$ *gives the total number of adoptions in year* y*.*

The parameter space of the claim $\mathcal{P}$ *consists of triples* $p = (w, t, d)$ *where* w *is the length of the period being compared,* t *the final year of the second comparison period, and* d *the distance between the start of each period. Define the parametrized query template as*

$$q(w, t, d) = \frac{\sum_{y=t-w+1}^{t} \operatorname{adopt}(y)}{\sum_{y=t-d-w+1}^{t-d} \operatorname{adopt}(y)}.$$

Results $q(p) \in \mathcal{R} = \mathbb{R}$ *are ratios of the total number of adoptions in the second period compared to the first period.*

Giuliani's precise claim is given by $(q, p = (w = 6, t = 2001, d = 6), r = 1.665)$*.*

The claimed result r may differ from the "true" result $q(p)$, in which case a significant difference indicates that the claim (q, p, r) is incorrect. For the example it is true that $r \approx q(p)$ [4], and so the claim appears to be supported by the data.

3.2 Relative result strength and parameter sensibility

We are however not simply interested in obviously false claims, but also in cases where a claim may be technically true, yet misleading. This is the case in the example, where the date periods were chosen in order to inflate the apparent increase in adoptions. If one instead compares

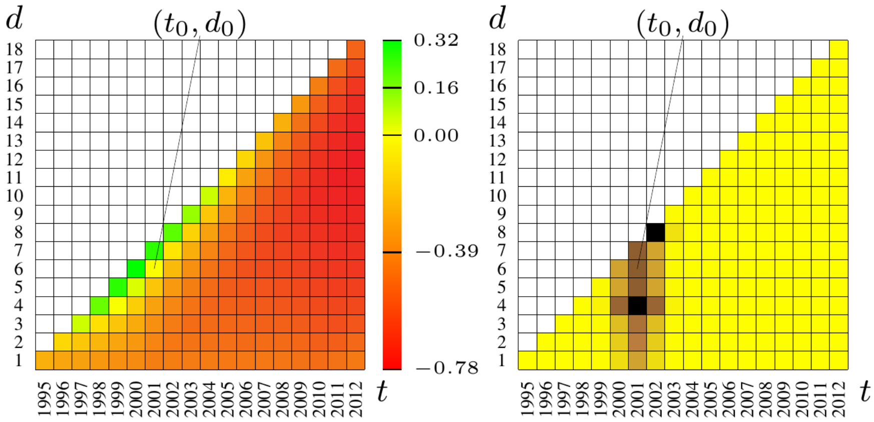

Figure 2: Plots taken from [8], and show only the level $w = 6$. (Left) Relative result strength. Values indicated by color as shown. (Right) Relative parameter sensibility. Darker colors indicate higher sensibility scores.

Giuliani's first and second mayoral terms (1994–1997 vs. 1998–2001), one finds that the total adoptions actually *decreased*—that is, the result for those parameter settings is smaller than the result stated by the claim. This suggests the following general strategy to detect cherry-picking and find counterarguments: look for more "sensible" parameter settings that "weaken" the result of a stated claim. This motivates the following definitions.

Relative result strength is a function

$$S_R \colon \mathcal{R} \times \mathcal{R} \to \mathbb{R}$$

satisfying $S_R(r, r) = 0$ for all $r \in \mathcal{R}$. Given a claim $c = (q, p, r)$ and a *reference claim* $c_0 = (q, p_0, r_0)$, we say that c is *weaker* (respectively *stronger*) than c_0 if $S_R(r, r_0) < 0$ (respectively $S_R(r, r_0) > 0$).

We can define parameter sensibility in two different ways. A **relative parameter sensibility function** is a function

$$S_P \colon \mathcal{P} \times \mathcal{P} \to \mathbb{R}.$$

The *relative sensibility score* of a parameter p with respect to a reference parameter p_0 is given by $S_P(p, p_0)$. As before, higher scores indicate more sensible parameter settings.

A relative parameter sensibility function induces a weak order on $\mathcal{P}$. However sometimes there is no sensible way for such an order to exist, in which case it suffices to have a partial order $\preceq^{p_0}$ on $\mathcal{P}$, known as a **relative parameter sensibility relation** with respect to p_0. Then $p_1 \preceq^{p_0} p_2$ means that p_2 is at least as sensible as p_1, relative to p_0.

Given a reference claim (q, p_0, r_0) we define the **relative result strength surface**

$$\{(p, SR_{(}q(p), r_0)) \mid p \in \mathcal{P}\}$$

and the **relative parameter sensibility surface**

$$\{(p, S_P(p, p_0) \mid p \in \mathcal{P}\}.$$

Example. *Figure 2 shows the result strength and parameter sensibility surfaces relative to Giuliani's original claim* $w_0 = 6, t_0 = 2001, d_0 = 6, r = 1.665$. *For simplicity only the portion of each surface with* $w = 6$ *is shown.*

Relative result strength is defined as

$$S_R(r, r_0) \coloneqq \frac{r}{r_0} - 1,$$

and parameter sensibility is defined as product of "naturalness" and "relevance" functions

$$S_P(p, p_0) \coloneqq S_P^{nat}(p) \cdot S_P^{rel}(p, p_0)$$

(for details we refer to Section 4.1 of [8]).

From the plots we see that most alternative parameter settings weaken the original claim. We also see that there are at least two alternative parameter settings with $w = 6$ that are more sensible than the original claim, namely $(t = 2001, d = 4)$ and $(t = 2002, d = 8)$.

In the next sections we will see how various fact-checking tasks can be formulated as optimization problems over these surfaces.

3.3 Task formulations

3.3.1 Counterargument finding

A **counterargument** to (q, p_0, r_0) is a weaker claim (q, p, r), i.e. a claim with $S_R(r, r_0) < 0$. As seen in Figure 2 there may be multiple counterarguments to a claim, but we are interested only in *good* ones, namely those that both weaken the claim significantly and are highly sensible.

Thus the task of finding counterarguments is a two-objective optimization problem over relative result strength and relative parameter sensibility. Depending on our goals we can formulate this problem in different ways. Given a claim (q, p_0, r_0) we define:

> **(CA-τ_R)** Given $\tau_R \leq 0$ and a relative parameter sensibility relation $\preceq^{p_0}$ find all $\preceq^{p_0}$-maximal $p \in \mathcal{P}$ with $S_R(q(p), r_0) < \tau_R$. (Maximize parameter sensibility subject to a minimum constraint on the amount of weakening.)
>
> **(CA-τ_P)** Given a relative parameter sensibility function S_P and $\tau_P \geq 0$ find all $p \in \mathcal{P}$ minimizing $S_R(q(p), r_0)$ such that $S_P(p, p_0) > \tau_P$. (Maximize the amount of weakening subject to a minimum sensibility constraint.)
>
> **(CA-po)** Given S_P and $k \in \mathbb{N}$ enumerate the k most-sensible Pareto-optimal $p \in \mathcal{P}$.

In CA-po, Pareto-optimality of a claim is with respect to increasing relative sensibility and decreasing relative strength.

3.3.2 Reverse-engineering vague claims

Often a claim will be vague—for example it may state its result r_0, but not explicitly provide parameter values p_0 that we can directly substitute into a parametrized query template. In other cases the result itself may be rounded or imprecisely stated. In such cases we would like to *reverse-engineer* the claim to make it more precise, while staying as close as possible to its most probable original intention.

To do this we use the context of the claim to approximate "reasonable" initial values for p_0 or r_0, and then look for sensible parameter settings p relative to p_0 that yield results $q(p) \approx r_0$. The precise formulations of the reverse-engineering tasks are analogous to counterargument finding. For the sake of brevity here we only define:

> **(RE-τ_R)** Given a vague claim (q, p_0, r_0), relative sensibility relation $\preceq^{p_0}$, and $\tau_R > 0$, find all $\preceq^{p_0}$-maximal parameter settings $p \in \mathcal{P}$ such that $|S_R(q(p), r_0)| < \tau_R$.

RE-τ_P and RE-po are defined similarly to CA-τ_P and CA-po, where we minimize $|S_R(q(p), r_0)|$ instead of $S_R(q(p), r_0)$.

3.4 Algorithms

Solving both counterargument finding (CA) and reverse-engineering (RE) tasks essentially involves enumerating parameter values $p \in \mathcal{P}$ to find solutions to the particular variant of the optimization problem being considered.

Focusing on CA-po and the case where $\mathcal{P}$ is discrete and finite, the authors present three general algorithms of increasing complexity and efficiency. The most basic algorithm simply generates all possible parameter values and looks for Pareto optima among them. This latter problem is efficiently solvable [1, 5], so the runtime is dominated by the cost of enumerating all $p \in \mathcal{P}$. More efficient algorithms are given by enumerating parameters in order of decreasing relative sensibility, as well as divide-and-conquer over $\mathcal{P}$, but for brevity we do not report on these approaches here.

3.5 Experimental results

We return to our example claim "adoptions increased by 65 to 70 percent over Giuliani's tenure as mayor". This is a vague claim since the result is rounded to "65 to 70 percent", and the parameter value is imprecisely given as "when [Giuliani] was mayor".

So we would like to reverse-engineer the claim. Take $r_0 = 1.67$ to be the geometric mean of 1.65 and 1.70. Knowing that Giuliani was mayor from 1994 to 2001 (the *claim context*), we set p_0 to be given by the approximation $(w = 1, t = 2001, d = 8)$ comparing the one-year windows beginning 1993 and 2001, spaced eight years apart. Solving RE-po for the vauge claim (q, p_0, r_0), the authors obtain $(w = 6, t = 2001, d = 6)$ as one of the top two most sensible answers. This corresponds to the claim comparing the periods 1990–1995 and 1996–2001, exactly the claim as later clarified by the campaign.

Given this reverse-engineered claim, the authors now solve CA-po to find counterarguments. The most sensible parameter setting was $p = (w = 4, t = 2001, d = 4)$, comparing total adoptions over 1994–1997 and 1998–2001, i.e. Giuliani's first and second terms. The result of this comparison is a 1 percent decrease over the entirety of his tenure—the actual trend after he took office.

3.6 Further work

Here we have focused on the problems of reverse-engineering and counterargument finding for a specific kind of claim which the authors call *window aggregate comparison* claims, and have not gone into detail about the algorithms to solve the formulated tasks. In [8] and the accompanying technical report [9] the authors consider other kinds of claims, give experimental results for more datasets, and provide more detail and analysis of the algorithms, including comparisons of their runtimes and scalability.

References

[1] S. Börzsönyi, D. Kossmann, and K. Stocker. 2001. The Skyline Operator. In *Proceedings of the 17th International Conference on Data Engineering.* IEEE Computer Society, Washington, DC, USA, 421-430.

[2] G. L. Ciampaglia, P. Shiralkar, L. M. Rocha, J. Bollen, F. Menczer, and A. Flammini. 2015. Computational Fact Checking from Knowledge Networks. *PLoS ONE* 10, 6 (June 2015), e0128193. DOI: `http://doi.org/10.1371/journal.pone.0128193`

[3] DBpedia: Towards a Public Data Infrastructure for a Large, Multilingual, Semantic Knowledge Graph. 2017. Retrieved July 23, 2017 from `http://wiki.dbpedia.org/`

[4] B. Jackson. 2007. Levitating Numbers. (May 2007). Retrieved July 17, 2017 from `http://www.factcheck.org/2007/05/levitating-numbers/`

[5] H. T. Kung, F. Luccio, and F. P. Preparata. 1975. On Finding the Maxima of a Set of Vectors. *J. ACM* 22, 4 (October 1975), 469-476. DOI: `http://dx.doi.org/10.1145/321906.321910`

[6] NIH Semantic Knowledge Representation. 2017. Semantic MEDLINE Database. Retrieved July 23, 2017 from `https://skr3.nlm.nih.gov/SemMedDB/`

[7] B. Shi and T. Weninger. 2016. Discriminative predicate path mining for fact checking in knowledge graphs. *Know.-Based Syst.* 104, C (July 2016), 123-133. DOI: `http://dx.doi.org/10.1016/j.knosys.2016.04.015`

[8] Y. Wu, P. K. Agarwal, C. Li, J. Yang, and C. Yu. 2014. Toward computational fact-checking. *Proc. VLDB Endow.* 7, 7 (March 2014), 589-600. DOI: `http://dx.doi.org/10.14778/2732286.2732295`

[9] Y. Wu, P. K. Agarwal, C. Li, J. Yang, and C. Yu. 2015. Toward computational fact-checking. Technical report, Duke University. Retrieved July 22, 2017 from `http://db.cs.duke.edu/papers/WuAgarwalEtAl-13-fact_check.pdf`