Visualization of Trustworthiness Graphs

Stephen Mayhew, Dan Roth
University of Illinois, Urbana-Champaign
201 N. Goodwin
Urbana, Illinois, 61801
{mayhew2,danr}@illinois.edu

ABSTRACT

Trustworthiness is a field of research that seeks to estimate the credibility of information by using knowledge of the source of the information. The most interesting form of this problem is when different pieces of information share sources, and when there is conflicting information from different sources. This model can be naturally represented as a bipartite graph. In order to understand this data well, it is important to have several methods of exploring it. A good visualization can help to understand the problem in a way that no simple statistics can.

This paper defines several desiderata for a "good" visualization and presents three different visualization methods for trustworthiness graphs.

The first visualization method is simply a naïve bipartite layout, which is infeasible in nearly all cases. The second method is a physics-based graph layout that reveals some interesting and important structure of the graph. The third method is an orthogonal approach based on the adjacency matrix representation of a graph, but with many improvements that give valuable insights into the structure of the trustworthiness graph.

We present interactive web-based software for the third form of visualization.

Categories and Subject Descriptors

H.5.0 [**Information Interfaces and Presentation**]: General

Keywords

trustworthiness; visualization; graph; adjacency matrix

1. INTRODUCTION

Problems that involve inference over graph structures are ripe for visualization. This is due to the fact that the structure of the graph nearly always matters to the inference algorithm. The world of graph theory gives us many ways to characterize the structure of a graph (degree distribution, connectedness, number of cliques, diameter, PageRank [1]), but if a picture is worth a thousand words, a graph visualization is worth a thousand statistics.

Copyright is held by the International World Wide Web Conference Committee (IW3C2). IW3C2 reserves the right to provide a hyperlink to the author's site if the Material is used in electronic media.
WWW 2015 Companion, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3473-0/15/05.
http://dx.doi.org/10.1145/2740908.2742574.

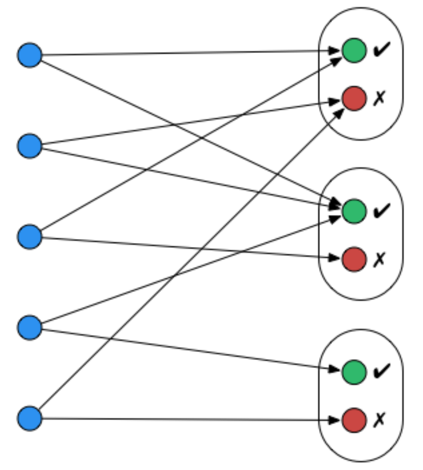

Figure 1: A standard bipartite layout of a trustworthiness graph. Sources are blue, claims are green if true, red if false. Edges connect claims to their sources. This is toy data.

Even disregarding the possibility of an incomplete set of characteristics, it is entirely possible that when analyzing the graph, one may not think to check even obvious characteristics (e.g. graph connectedness). Although data exploration is a crucial first step to a good understanding of a problem, it is often skipped, either because it is not immediately clear how to visualize, or because algorithm development may seem more important.

This paper argues that a good visualization of a trustworthiness graph can bring understanding to the problem that no other metric can, and illustrates this with real-world data. We believe that our visualization recommendations can help trustworthiness researchers to develop better trustworthiness algorithms.

We begin by defining the problem of trustworthiness, give a brief literature review, then describe the data that is being analyzed in these settings, and then move to our proposed visualization approaches.

2. TRUSTWORTHINESS

Trustworthiness is a field in which algorithms are used to infer such qualities as believability, credibility, or trustworthiness of propositional statements, or claims, by using the source of the claims. Intuitively, if we don't know the veracity of a given claim, but we do know that the source of the claim has been reliable in the past, then we can make a reasonable judgement on the unknown claim. This may be called decision by context, where the context refers to neighbors in an underlying graph. (In an orthogonal approach, it certainly makes sense to discern trustworthiness from inherent characteristics of data, as in textual entailment. This might be called decision by content. We focus only on the contextual side of trustworthiness here.)

We define a trustworthiness graph $G(V, E)$ as having vertices V, partitioned into sources $S \subset V$ and claims $C \subset V$. That is, $S \cup C = V$, and $S \cap C = \emptyset$. Edges are directed and point from S to C. Further, C is partitioned into M mutual exclusion sets. The mutual exclusion set is important semantically for this to be a trustworthiness graph: there is typically one correct answer in each mutual exclusion set. That is, a source may not claim that X is true and also claim that X is false. There are versions of the problem that relax these restrictions, which might allow multiple correct answers.

The simplest trustworthiness algorithm is majority voting, in which the "true" answer in each mutual exclusion set is taken to be the one with the greatest degree (the most sources asserting it). The next simplest algorithm is Hubs and Authorities, or HITS [2], also known as Sums [3]. For each node, we initialize the hub score, y_p, and the authority score, x_p, to 1. Then, we iterate until convergence (normalizing at each iteration).

$$x_p \leftarrow \sum_{q:(q,p)\in E} y_q, \quad y_p \leftarrow \sum_{q:(p,q)\in E} x_q$$

Finally, having obtained a hub and authority score for each node, we order the claim nodes in each mutual exclusion set by the authority score. The claim with the largest authority is chosen to be "true". (Note that in the specific case of using a bipartite graph, directed from left to right, the sources will have authority score 0 and the claims will have hub score 0).

One open question in the field is regarding how the structure of the trustworthiness graph affects the algorithm. For example, HITS can be shown to perform well on certain hand-made graphs, but for most graphs it produces results nearly identical to the majority voting algorithm. An important step in understanding how the structure of trustworthiness graphs affects the result is to visualize each graph.

3. RELATED WORK

Although there is no work directly related to visualization of trustworthiness, there is work on trustworthiness, and work on visualization of graphs.

Trustworthiness algorithms have been studied in a variety of applications: information retrieval [1, 2], fact finding [3], knowledge base population [4, 5], crowdsourcing [6], reputation estimation [7], classification [8], and information extraction validation [9]. One common thread between all of these is the trustworthiness graph.

Traditionally, research on visualization of graphs has focused on node-link representations [10]; that is, a visualization consisting of nodes and links or edges (as seen in Figures 1 and 2). The problems to be solved involve minimizing edge crossings, accurately depicting communities and structure, and various other aesthetic and structural criteria.

However, work in [11] suggests that optimizing for these graph criteria may not result in aesthetically pleasing and easily understandable graphs. Motivated by this result, [12] proposes that adjacency matrix-based representations can be more readable, especially for graphs that are large or dense. An adjacency matrix obviates the need for complicated and slow layout algorithms, and has the ability to present data in a compact format. Since then, there have been several projects taking this approach [13, 14].

In this work, we propose a new matrix-based representation for visualizing trustworthiness graphs.

Sources	
Total	52
Claims	
Total	49496
Unique	15828
Claims per Source	
Mean	952
Max	2731
Min	103
Std dev	601
Mutual Exclusion Sets	
Total	1897
Claims per ME	
Mean	26
Max	598
Min	1
Std dev	47

Table 1: Statistics over the KBP2013 data. Notice how this table alone gives little insight into the data.

4. DATA

For concrete illustration purposes, we use data from the Knowledge Base Population (KBP) Slot Filling Validation (SFV) task, from the year 2013 [15]. This task is a follow-on task from the Slot Filling (SF) task. In SF, a participating system is given a set of queries, and a set of documents (mostly newswire documents). Each query consists of an entity, which is either a person or an organization, and several "slots" to be filled. Slots are dependent on the type of the entity. For example, a person entity has such slots as "date_of_birth", or "cause_of_death", while an organization entity has slots like "date_founded" and "website".

For each query, each participating system fills in as many slots as it can. Then, the outputs from all systems are aggregated to form the input dataset for SFV, where the goal is to decide if a "filler" for a given slot is true or false.

Each slot may be filled by several different sources separately. Since not all sources have the same quality or accuracy, this means that there may be conflicting information, which makes this an interesting trustworthiness problem.

To make the connection explicit, each SF system is considered a "source" and each filler is considered a "claim". A given slot for a given query (e.g. "Johanna Smith : date of birth") forms a mutual exclusion set, and all fillers for that query+slot are the members in the mutual exclusion set.

This data is labeled, which means that every filler has a binary true or false label associated with it. The visualization methods described below are targeted at labeled data, but are valuable also for unlabeled data.

5. VISUALIZATION

As motivation, let us consider some simple statistics over the KBP 2013 data (Table 1). It is helpful to remember that this is a baseline representation. It is useful to ask, what is missing? What is hidden?

As a way to formalize such questions, we define a set of desiderata for a thorough understanding of trustworthiness data. Items with stars (*) are only applicable if a labeling of claims is available.

1. Know if the graph is connected. (If not, know the distribution of component sizes).
2. Know the number of sources and the number of claims.
3. Know the distribution of claims over sources and sources over claims.
4. For each claim:
   (a) Know how many sources made this claim.
   (b) Know if responding sources responded to similar claims.
   (c) * Know the accuracy of responses for this claim.
   (d) * Know the accuracy of responses to similar claims.
   (e) * Know the accuracy of each responding source.
5. For each source:
   (a) Know how many claims this source made.
   (b) Know which types of claims this source tended to respond to.
   (c) * Know the accuracy of this source.
   (d) * Know how this source compares to other sources in terms of claims answered, and in terms of accuracy.
6. For each mutual exclusion set:
   (a) Know how many claims are in this set.
   (b) * Know the breakdown of true and false claims.
   (c) * Compare claim response patterns with similar mutual exclusion sets.

While the first three of these can be represented with simple statistics, some important information may remain hidden. For example, while point 1 is certainly useful, it is not always informative. Imagine a subgraph that is isolated from the main graph except for a small number of edges. A connectedness algorithm would correctly call this graph connected, but would miss the important subtlety of being *almost not* connected. (The graph is Figure 2 is one example of this situation.)

It may be said that knowing the minimum-cut of the graph could be useful. This is not quite true – the min-cut on most trustworthiness graphs is 1. A more useful form could be the balanced min-cut algorithm, which sets constraints on the size of each partition. Nonetheless, the right visualization can give this information and more.

We now present three different visualization options.

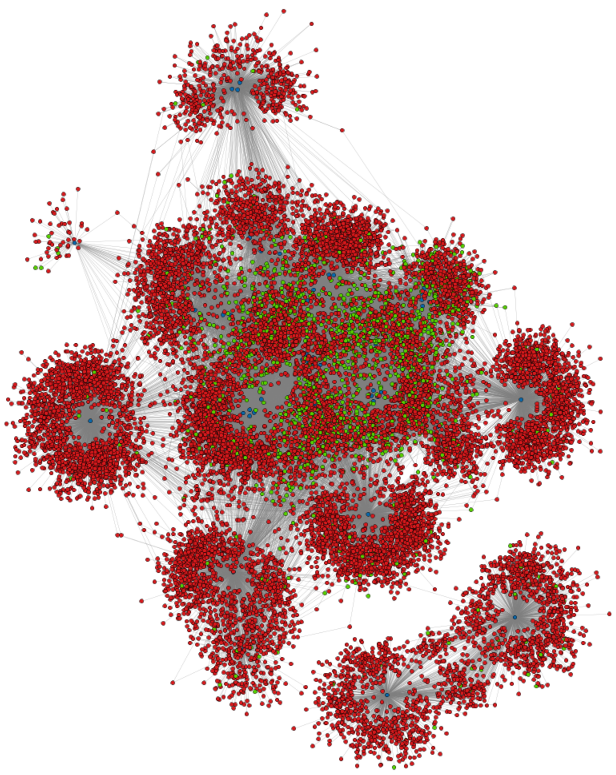

Figure 2: Physics-based layout. Red and green nodes represent false and true claims respectively, blue nodes represent sources, edges are grey. This visualization highlights how communities naturally form in the data. This is KBP2013 SFV data.

5.1 Standard Bipartite

Because the trustworthiness graph is bipartite, there is an obvious representation choice – the node-link format, with sources stacked vertically on the left, claims, grouped by mutual exclusion set, stacked on the right, and edges between them. Figure 1 shows an example of this on a toy dataset.

While this satisfies several of the simple desiderata, it breaks down at scale. When there are hundreds or thousands of claims, this becomes unwieldy. If we put the sources on the left and the claims on the right, each arranged vertically as in Figure 1, then there could well be more nodes in the claims column then there are vertical pixels in most screens.

But even if that weren't a fatal flaw, there is the problem of minimizing edge crossings, which is known to be NP-complete [16]. There are heuristics to aid in solving this problem, but we would like to suggest that the best solution is to not have the problem in the first place.

This visualization is presented mostly as a warning: we have tried it, and it is not helpful.

5.2 Physics-based Layout

One way to obviate the problem of edge-crossings is to allow entirely new node placements (i.e. not in two rigid lines). One nice way to do this is to use a physics-based layout, as seen in Figure 2. This is using Gephi [17] as the

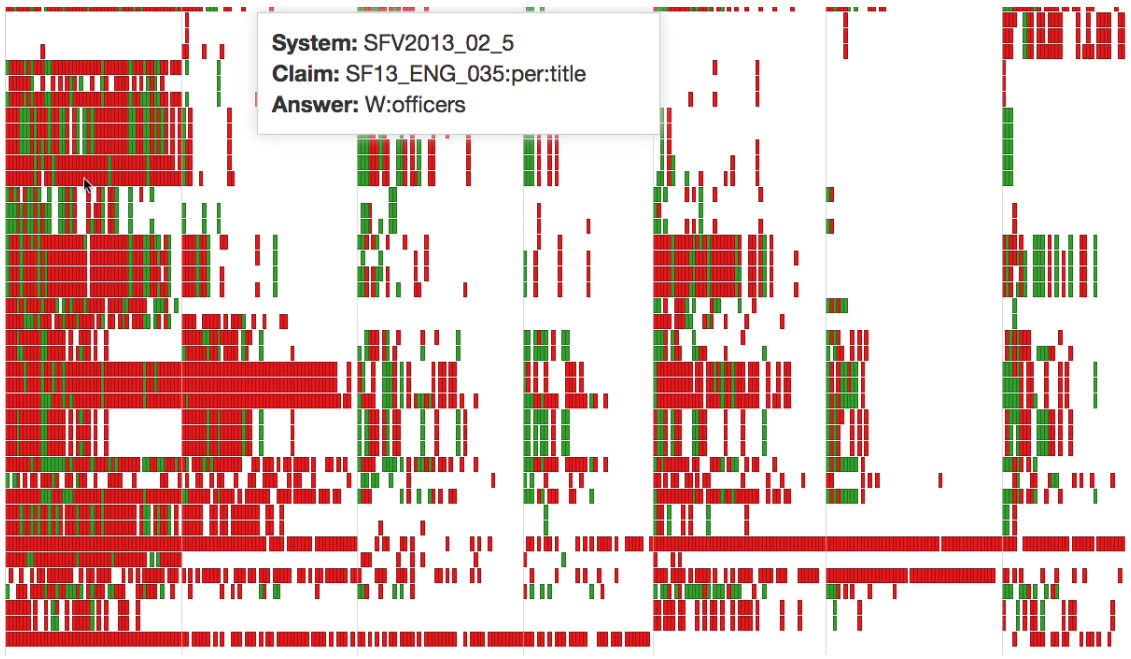
System: SFV2013_02_5
Claim: SF13_ENG_035:per:title
Answer: W:officers

Figure 3: Adjacency matrix representation. Each column represents a mutual exclusion set, and each rectangle in the column represents an individual claim. Each row represents a source, and all rectangles in that row are claims made by that source. In this image, mutual exclusion sets (columns) are grouped by their slots (i.e. title, or spouse). The faint gray lines delineate different groups. This is KBP2013 SFV data.

visualization engine, and the OpenOrd layout[1] [18], using the KBP2013 data introduced in Section 4. In a physics-based layout, each node is treated as a mass (with some small repellent forces for all other masses), and each edge is treated as a spring. The layout runs a physical simulation in which forces operate on the 2-dimensional graph until there is an equilibrium. Intuitively, this tends to make edges as short as possible, and to isolate strongly-connected communities of nodes. There are a variety of layout algorithms, physics-based and otherwise [10]. We chose OpenOrd because it produced the most informative layout.

The nodes are color-coded according to correctness: the green nodes are true, and the red nodes are false. Blue nodes represent sources, and edges are grey. In this dataset, clearly, most claims are incorrect.

This layout can provide some interesting insights. First, and most importantly, it gives a clear idea of how connected is the graph. This is very important among algorithms like HITS [2], where a disconnected graph can produce meaningless results (in a nutshell: influence cannot flow between disconnected graphs).

In a similar vein, this gives a good idea of the communities in the graph. Claims that are asserted by a single source are placed close to the source. Such claims are intuitively unlikely to be correct – in general, we would expect at least two witnesses.

Another value of this layout is a quick idea of where the correct nodes tend to lie. For example, it's possible that a trustworthiness graph may have one source providing nearly all of the correct answers, and the rest making false claims. This layout will show this clearly. For the dataset presented, the visualization confirms the intuition that "loner" sources are less likely to produce correct answers, and that correct answers tend to correlate with high degree.

5.3 Adjacency Matrix Representation

Although a physics-based layout is useful for getting a bird's eye view of the data, for quickly discovering communities, and for seeing how sources interact with each other, it still leaves something to be desired. In particular, information about individual mutual exclusion sets is nonexistent (see point 6 in the desiderata). It is possible and even likely that the nodes of a mutual exclusion set are spread out.

Our final proposal is built on the observation that we want to see structure on both sides of the bipartite graph, not just the source side. Further, on the claim side, we would like to have visual information about mutual exclusion sets.

This visualization has two modes. The first mode (seen in Figure 3) is based on a modified adjacency matrix representation of a graph. It is modified in the sense that each mutual exclusion set becomes a supernode, replacing all claim nodes which it contains. A source is connected to the supernode if it is connected to a claim in the set. Since the graph is bipartite and directed, only one quadrant of the adjacency matrix is non-zero, and only this quadrant is shown.

In other words, the rows are sources, and the columns are mutual exclusion sets. A cell is filled if the source (row) made a claim in the mutual exclusion set (column). In the visualization, a filled cell is represented with a colored rectangle. Again, red and green represent false and true claims respectively. When the mouse is hovered over a rectangle, an information box displays the values for that claim.

In the second mode, each column is collapsed to look like

[1]https://marketplace.gephi.org/plugin/openord-layout/

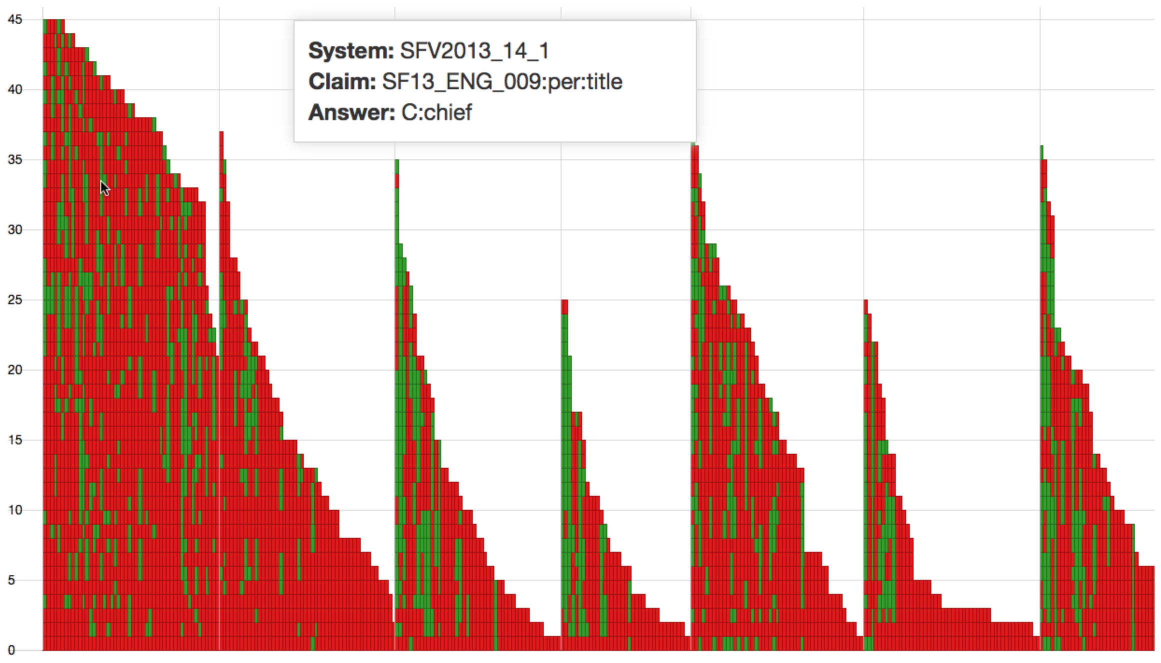

Figure 4: Collapsed adjacency matrix representation. Each column represents a mutual exclusion set, and each square in the column represents an individual claim. The vertical axis merely serves to measure the number of claims in each column. This is the collapsed view of Figure 3.

a histogram of claims per mutual exclusion set, as shown in Figure 4.

Note that each rectangle represents an individual claim, not a unique answer. These are differentiated from each other not by the content, but by the source. This is different from the graph-based representations where a claim node represents a unique answer, and the degree of the node represents how many systems have made that claim. There are many more rectangles in this representation than there are nodes in the graph-representations. The rectangles loosely correspond to edges in the graph.

One stroke against the bipartite layout was that few screens are large enough to display all the nodes. Although this approach may seem similar, the matrix form affords a substantial benefit. First, the claims can be drawn much more compactly because the x and y dimensions have meaning. Second, in the bipartite layout, it is difficult (or at least tedious) to see the source of a claim because one needs to follow the edge from source to claim. There is no guarantee that source and claim will be close to each other.

For both modes, there is the option to sort the columns by a number of different metrics, for example, by number of claims in the column, or by number of correct claims in the column. In the KBP2013 data, each mutual exclusion set is composed of a query and a slot, so these are sorting options also. Naturally, it is possible to write custom sorting metrics for any new data set.

The power of this visualization comes from the ability to sort columns, and from the ability to expand the columns and see how sources respond over different sortings. With these features, we are now equipped to answer many questions that the two prior visualizations could not.

We can now see the distribution of claims inside different groupings. For example, in the KBP2013 data, it is natural to group claims by slot. Each group then has a secondary sort on the number of claims per column. From this view, we are able to quickly and easily compare the number of claims per slot. Since the group is sorted by the number of claims, we can also see the how claims are distributed. For example, the leftmost group in Figure 4 has a large number of claims and what looks like a linear distribution, while the group second from the right has far fewer claims, and follows something similar to an exponential distribution.

An analogous analysis can be made when the claims are grouped by query.

We can also see how systems respond by different groupings. For example, when grouping by slot, we can see how certain systems respond only to certain slots, or avoid certain slots. Perhaps more valuable is the view that certain systems tend to answer almost indiscriminately, even when other systems are cautious in their answers. In Figure 3, the bottom system is an example of both these points, evidenced by a nearly solid line of red punctuated by a complete silence in two slot groups. To a designer of a trustworthiness algorithm, this gives the valuable intuition that high output systems may be untrustworthy.

All of this information is useful because it gives interesting insights into the structure of the graph. There are theoretical results on the performance of trustworthiness algorithms on specific types of graphs, namely regular graphs [6]. However, it is not clear how to create an algorithm to perform optimally on graphs that are not regular (such as the example graph in this paper). In such a case, it is not even clear how to quantify “irregularity”, and the need for other ways to explore the data becomes apparent.

Finally, this visualization is not only useful in the initial data exploration phase. After a trustworthiness algorithm has been run on the data, it is possible to filter the dataset by the labeling of each claim (i.e. keeping only the claims labeled as true). It is interesting to see which systems are adjudged to have provided the correct answers.

This visualization is available as an interactive demo.[2]

6. DISCUSSION

We have presented a list of desiderata for a good trustworthiness visualization, and argued that these cannot be satisfied using only simple statistics on the graph. To meet these desiderata, we have shown three different visualization techniques, and commented on their effectiveness.

The first method is using a simple standard bipartite layout of a graph. We argued that this method is infeasible for the purpose because it does not scale elegantly. The second method is using a physics-based layout of the trustworthiness graph. We found that this visualization is useful in discovering large-scale communities of nodes, and for getting a global picture of the data. Unlike the first method, this scales to a large number of nodes. But while it does well at big-picture understanding, it fails to reveal more fine-grained facets. Our final method, based on the adjacency matrix representation of the graph, gives much more control by allowing exploration of the data by different attributes of claims and sources.

7. ACKNOWLEDGEMENTS

We would like to thank the reviewers for their thoughtful and helpful comments and the creators of the d3 library[3] for making such an effective and beautiful tool.

This research was supported by Army Research Laboratory (ARL) under agreement W911NF-09-2-0053, by DARPA under agreement number FA8750-13-2-0008, and by the Multimodal Information Access & Synthesis Center at UIUC, part of CCICADA, a DHS Science and Technology Center of Excellence. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. Any opinions, findings, conclusions or recommendations are those of the authors and do not necessarily reflect the view of ARL, DARPA or the U.S. Government.

8. REFERENCES

[1] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1):107–117, 1998.

[2] Jon M Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

[3] Jeff Pasternack and Dan Roth. Knowing what to believe (when you already know something). In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 877–885. Association for Computational Linguistics, 2010.

[4] Xin Luna Dong, Evgeniy Gabrilovich, Kevin Murphy, Van Dang, Wilko Horn, Camillo Lugaresi, Shaohua Sun, and Wei Zhang. Knowledge-Based Trust: Estimating the Trustworthiness of Web Sources. *arXiv.org*, February 2015.

[2]`http://cogcomp.cs.illinois.edu/~mayhew2/kbpvis/`
[3]`http://d3js.org/`

[5] Xin Luna Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Kevin Murphy, Shaohua Sun, and Wei Zhang. From data fusion to knowledge fusion. *Proceedings of the VLDB Endowment*, 7(10), 2014.

[6] David R Karger, Sewoong Oh, and Devavrat Shah. Iterative learning for reliable crowdsourcing systems. In *Advances in neural information processing systems*, pages 1953–1961, 2011.

[7] Sepandar D Kamvar, Mario T Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the 12th international conference on World Wide Web*, pages 640–651. ACM, 2003.

[8] Jennifer Neville and David Jensen. Iterative classification in relational data. In *Proc. AAAI-2000 Workshop on Learning Statistical Models from Relational Data*, pages 13–20, 2000.

[9] Dian Yu, Hongzhao Huang, Taylor Cassidy, Heng Ji, Chi Wang, Shi Zhi, Jiawei Han, Clare Voss, and Malik Magdon-Ismail. The wisdom of minority: Unsupervised slot filling validation based on multi-dimensional truth-finding. In *Proc. The 25th International Conference on Computational Linguistics (COLING2014)*, 2014.

[10] Yifan Hu and Lei Shi. Visualizing large graphs. *Wiley Interdisciplinary Reviews: Computational Statistics*, 7(2):115–136, 2015.

[11] Helen C. Purchase, Robert F. Cohen, and Murray I James. An experimental study of the basis for graph drawing algorithms. *Journal of Experimental Algorithmics (JEA)*, 2:4, 1997.

[12] Mohammad Ghoniem, J Fekete, and Philippe Castagliola. A comparison of the readability of graphs using node-link and matrix-based representations. In *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*, pages 17–24. IEEE, 2004.

[13] Niklas Elmqvist, Thanh-Nghi Do, Howard Goodell, Nathalie Henry, and J Fekete. Zame: Interactive large-scale graph visualization. In *Visualization Symposium, 2008. PacificVIS'08. IEEE Pacific*, pages 215–222. IEEE, 2008.

[14] J Fekete. Visualizing networks using adjacency matrices: Progresses and challenges. In *Computer-Aided Design and Computer Graphics, 2009. CAD/Graphics' 09. 11th IEEE International Conference on*, pages 636–638. IEEE, 2009.

[15] Mihai Surdeanu. Overview of the tac2013 knowledge base population evaluation: English slot filling and temporal slot filling. In *Proceedings of the Sixth Text Analysis Conference (TAC 2013)*, 2013.

[16] Peter Eades and NicholasC. Wormald. Edge crossings in drawings of bipartite graphs. *Algorithmica*, 11(4):379–403, 1994.

[17] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: An open source software for exploring and manipulating networks. In *International AAAI Conference on Weblogs and Social Media*, 2009.

[18] Shawn Martin, W Michael Brown, Richard Klavans, and Kevin W Boyack. Openord: An open-source toolbox for large graph layout. In *IS&T/SPIE Electronic Imaging*, pages 786806–786806. International Society for Optics and Photonics, 2011.