Visualizing Inference

Authors Anonymized
Address

Abstract

This work aims to bring the power of visualization to understanding knowledge bases and the operation of inference engines in AI. We present *Alar*, a visualization system for the knowledge base ConceptNet and inference engine AnalogySpace. Alar presents dynamically controllable node-and-arc graphs of concepts, and of assertions both supplied to the system and inferred. The links represent semantic similarity between entities, allowing a user to get an overview of a relevant neighborhood of a semantic space, rather than query it one assertion at a time.

Alar is designed to help the user answer the questions:

• How "liberal vs. conservative" should the inference be? Essentially, how easy is it to jump to conclusions vs. the risk of making false conclusions?
• How good is the knowledge in the KB? Are there incorrect assertions? Are there missing concepts or assertions that would enable desirable inferences?

We present a user study testing the efficacy of debugging a KB with Alar vs. a textual representation; and an empirical evaluation of our similarity measure for assertions, a key contribution.

The problem of understanding inference in AI

AI programs do inference. Therein lies their power. But also therein, lies a challenge. Because inference is often a black box to end-users, it is sometimes hard for people to develop confidence in AI programs. We believe that one route to increasing confidence in AI inference and AI applications is to open up the black box, and to provide a route for understanding in-depth the operation of AI algorithms. We also believe visualization will be valuable for AI application developers to help debug their programs and assure that they meet users' needs.

As in other scientific fields, graphical visualization is a very powerful tool for achieving understanding. It uses the enormous parallel computational power of the visual system to enable users to grasp the effect of a program on many examples at once. Interactive controls of graphical visualizations permit performing a multitude of experiments in the flash of an eye. However, very little prior work in AI is concerned with visualizing inference processes (we will survey related work later in the paper).

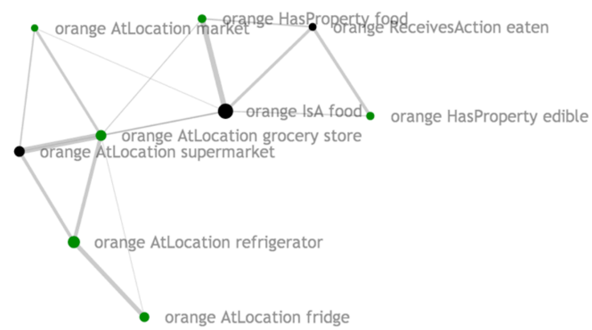

Figure 1. An Alar visualization, centered on the assertion "Orange is a food". Inferred assertions use related knowledge about food to infer new assertions, e.g. "Orange AtLocation grocery store".

We present Alar, a visualization system for a large commonsense knowledge base, ConceptNet, and its associated heuristic inference technique, AnalogySpace [Speer et al 08]. Alar can visualize both graphs of concepts, and also graphs of assertions, as in Figure 1. We believe the visualization of assertion graphs to be novel.

There are many kinds of inference in AI: logical inference, cognitive modeling of human inference, and many kinds of statistical inference in modern machine learning. Some kinds of inference will merit their own specific visualizations. Other kinds of visualizations may be valid across many different inference techniques. Inference techniques typically come with a large number of control parameters, which regulate things like weighting of particular kinds of knowledge sources, breadth-first vs. depth-first, and other tradeoffs. Visualization can be a powerful tool to tune these kinds of parameters.

Another role for visualization is to do quality assurance on the knowledge sources themselves. Incorrect inferences are sometimes due to incorrect knowledge. When evaluating inference, the gold standard is human inference, and it is sometimes difficult for a person to jump out of their own skin enough to see why a program is or isn't making a particular inference. Visualization externalizes the inference process so that people can see what is taking place.

ConceptNet and AnalogySpace

We are most interested in the problem of representing commonsense knowledge and commonsense inference. We use the knowledge base ConceptNet [Havasi et al 09], roughly similar in goals (though not in details) to Cyc [Lenat 95]. Assertions are derived from natural language sentences. Knowledge is represented via *concepts* (named by a word or noun phrase in natural language), and *assertions*, here a triple of two concepts and a relation: "Fork UsedFor Eating".

The basic inference is not strictly logical, nor probabilistic inference, but a kind of analogical inference called AnalogySpace. AnalogySpace uses similarity of concepts to infer new assertions, and similarity of assertions to infer similarity of concepts. Mathematically, it makes a matrix whose cells are the truth values of assertions, and computes the principal components via Singular Value Decomposition (SVD). Since the subject here is visualization and not the inference per se, it is not necessary to understand the algorithm. Knowledgeable readers can find a full explanation in [Speer et al 08], and the central equation below.

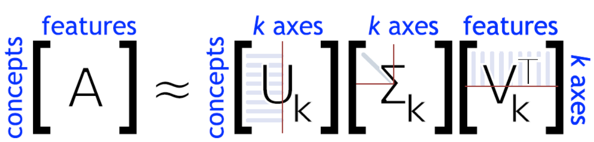

Figure 2. The central equation of AnalogySpace inference. It computes *k* "axes" which represent important semantic distinctions, like "good vs. bad". *K* controls how "liberal vs conservative" the inference is.

The thing to note is that there is an approximation parameter *k*, controlling *dimensionality* – roughly, how "liberal" (easy to jump to conclusions) vs. "conservative" (requiring a lot of evidence before concluding) the inference is. We would like the visualization to control this parameter in real time. Many statistical or inexact inference algorithms also include parameters that control this aspect, and so some of our visualization and computational techniques will also apply.

Here's a simplified example. If we know "A fork is used for eating", we might ask "Is a spoon used for eating?" If one of the axes happens to represent things that are found in a kitchen, the approximation may cause differences between a spoon and a fork to disappear and conclude that indeed, a spoon is used for eating, too.

Alar's interface

Alar's interface revolves around a graph representation of a portion of the semantic space. It can either show graphs of concepts, or graphs of assertions. The links represents the similarity of the entities. The thicker and shorter the link, the more similar the two entities it connects.

Alar treats nodes as charged particles which repel each

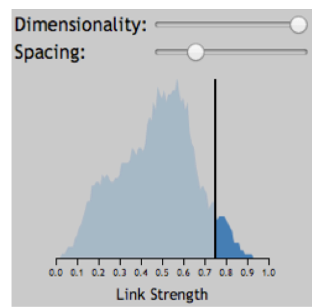

Figure 3. Interactive control over permissiveness of the inference, layout, and level of detail

other and links as springs which draw its respective nodes together. Links with higher semantic relatedness pull their nodes closer together. Spatial relationships *within* connected components express similarities, but absolute X-Y position of each node is not significant.

The visualization dynamically adjusts using the *force-directed layout* of the visualization toolkit D3JS [Bostock 14]. There are three interactive controls over the visualization, shown in Figure 3.

First, dimensionality, which controls how liberal or conservative the inference is. For concepts, liberal inference results in more links; for assertions, more inferences. Spacing supplies "negative gravity" that counteracts the pull of the inferential associations, making semantic clusters more readable. The link strength is a movable slider on a histogram of number of links vs. strength. Only links to the right of the slider are displayed; to the left is the "below water", thus invisible, part of the iceberg. This gives control over the level of detail displayed, and previews the effect of moving the slider.

The interface is seeded with one or more initial concepts (e.g. "Orange") or assertions ("Orange is a food"). The operation, "Add Related Nodes" finds the most similar concepts (or assertions) to the seeds and expands the graph. Figure 4 shows a concept graph centered around "Orange" that clearly delineates semantic clusters for the word's meaning as a color, and its alternative meaning as a fruit. [Havasi et al. 10] details the potential of commonsense knowledge for word-sense disambiguation.

Exploring assertion spaces in Alar

When Alar displays graphs of assertions, the size of the node's dot indicates the degree of truth ascribed to that assertion. Assertions that appear in the original sparse knowledge base (the axioms, in terms of traditional logic) are represented as black dots, the inferred assertions as green dots. Links between assertions represent similarity of assertions, as they do for similarity between concepts. They don't indicate directly that one assertion is derived from another, as they do in proof-tree visualizations like the Transparent Prolog Machine.

One of the uses of the visualization is to get a feeling for how the dimensionality interacts with the truth of a set of related assertions. In general, lower dimensionality will result in stronger connections between assertions, but less confidence in the truth of similar assertions.

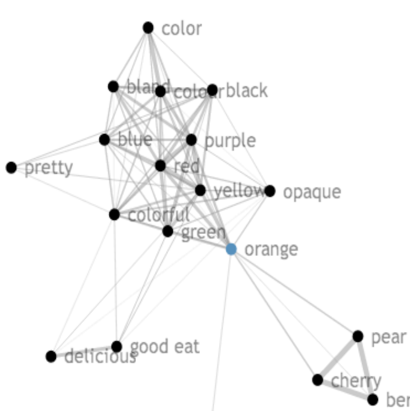

Figure 4. A concept graph. The two senses of "orange" {color, fruit} are clearly distinguished in semantic clusters.

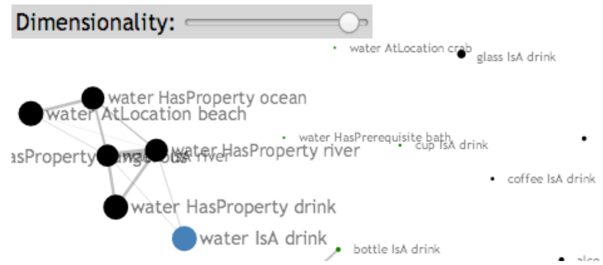

Figure 5. An assertion graph at high dimensionality. Near the seed "Water IsA Drink" are some reasonable assertions, but the nonsense "Cup IsA Drink" at middle right and "Glass IsA Drink" at upper right, are false (small dots).

Figure 6. An assertion graph at lower dimensionality. Assertions about water are still pretty reasonable, but the false "Glass IsA Drink" is now given more credulity (larger dot).

Figure 5 shows an assertion graph seeded with the assertion, "Water IsA Drink". To the upper left is a set of connected assertions that seem reasonable, such as "Water AtLocation Beach", and "Water HasProperty Drink". At the upper right is a nonsense assertion "Cup IsA Drink" (it was probably supposed to be "Cup UsedFor Drink"). Nonsense assertions can appear in the knowledge base for a number of reasons. They can be entered by malicious or careless users. They can result from a failure of the natural language parser. They can result from a failure of inference caused by missing but essential information, or other reasons. However, the assertion "Cup IsA Drink" appears as a relatively small dot, indicating that the system does not consider it very likely that it is true.

As we decrease the dimensionality, as in Figure 6, we can see in this case that there is little effect on the cluster of assertions above and to the left of the original "Water IsA Drink". Since we're performing more inference, we do get additional plausible inferences such as "Water HasProperty

Good", above and to the right of the original assertion. However, the dot representing the nonsense assertion "Glass IsA Drink" is also larger, which is undesirable. If we wanted to investigate further, we could move the link strength slider to the left, which would reveal a more detailed network containing more related assertions.

Figure 7. An expanded network showing that "Cup IsA Drink" was inferred from another bogus assertion, "Glass IsA Drink".

In this case, the "Cup IsA Drink" was inferred from another bogus assertion, "Glass IsA Drink" that appeared in the knowledge base. Removing the cup assertion would also eliminate the glass assertion.

A static picture doesn't fully convey the feeling of interacting with this visualization. In knowledge bases that are reasonably well behaved, it is often possible to visually arrive at good intermediate values by interactively playing with the slider.

Implementation

Alar is implemented as a single Web page in HTML and Javascript, with Ajax requests to the server. D3JS's [Bostock 14] physics simulation with "spring-loaded" links provide smooth transitions as the user drags nodes. Links higher above the threshold have more gravity than others, contributing to smoothness as links enter and leave the display.

The key implementation challenge is how to avoid unnecessary recomputation in order to keep the interface responsive. When dimensionality changes, we must update relatedness of concepts or assertions, and the truth of an assertion. So, when a node or link is to be added to the graph, the server also supplies the coefficients of the polynomial at the current dimensionality, then returns the current value to be displayed. In this way, the frontend merely executes a function call per data point, without needing to communicate with the server.

First, for a total of k dimensions, the value of each data point in question is found exactly for each of the k dimensions. Then, a closest fit polynomial of order $k-1$ is used. So the value of the polynomial evaluated at any integer dimension is the answer. Furthermore, there isn't a well-defined answer for any inferred value at anything other than an integer number of singular values. However, the slider itself is real valued in the interface and makes a much more fluid transition between dimensions. So, the polynomials smoothly change the values at play as the slider changes.

Alar represents both concepts and assertions as vectors in k-dimensional space, where k is the rank of approximated matrix. Per AnalogySpace, the concept vectors are formed from each row of the U matrix. Alar presents a novel formulation of the assertion vectors. Specifically, it is the change in its inferred truth at each of the k dimensions. This is a row of Us scaled by a row in V, for the concept and feature which form the assertion. This is motivated by the semantics, because assertions with similar vectors indicate they were inferred to be true or false at similar times in the reconstruction process. Normalized similarities of both concepts and assertions are found by finding the inner product of their vectors normalized by their norm, nicely mapping them to [-1,1].

We are able to render results in the interface immediately, at any dimensionality, because the results of the SVD for k dimensions can be used to find the results of the SVD at all dimensions $k' < k$. We truncate the concept or assertion vector in question to the first k' entries, as an SVD of k' dimensions is equivalent to zeroing out all but the top k' values in the top k' values in the Σ found with k. This means only the left k' columns are ever considered in U, Σ and V, which would result in truncating the vectors.

Now that these values are efficiently found, they are communicated to the interface as the coefficients of the order $k-1$ polynomial of the value of the metric in question over all k dimensions. This is the only possible polynomial of its order to exactly go through each value at each dimension, so answers are kept very close to the truth without overfitting. The coefficients are sent to the client to be reconstructed as a function in Javascript and re-evaluated each time the dimensionality slider changes, allowing the user to smoothly vary the results of inference with dimensionality.

Evaluation

We present two different evaluations. The first is a usability evaluation of Alar's interface. It supports our claim that Alar is useful as a debugging tool. We ask subjects to perform a task of finding a bug in a particular knowledge base (a false assertion that causes several incorrect inferences to be made), and we compare Alar's visual representation versus a textual representation, the conventional alternative.

The second is an empirical evaluation of the metric for semantic relatedness of assertions. Recall that Alar provides a dynamic display of a subset of assertions most related to one or more seed assertions chosen by the user. What makes Alar possible is that this can be computed and displayed in real time for realistically-sized knowledge bases. We cannot invoke heavyweight inference mechanisms for large sets of assertions on every mouse movement or redisplay. This evaluation checks that the metric is a good proxy for determining which assertions caused other assertions to be inferred.

Usability Test

Five users who were experienced computer users (but not AI experts), were given a knowledge base where we pointed out an obviously incorrect assertion. The task was to determine, within 10 minutes, what other assertions in the KB might have contributed to the incorrect inference (we are testing people's ability to *generate hypotheses* about what may have gone wrong, not to make definitive judgments about what *did* go wrong). There were asked to do this twice, once using a Microsoft Excel spreadsheet of all the original assertions (they could use any spreadsheet tools like sorting or searching for navigating the assertion set), once using Alar (randomizing the order). Again, since this was not intended to be a test of the inference algorithm itself, we relied on the users' own judgment as to whether the hypotheses they found were useful in the debugging process (but see below for our independent test of assertion relatedness). Replies to the following questions are on a Likert-5 scale. $C^2 = (1,N = 5) >=6.75$; $p <=0.01$ for all tests. All users said they preferred Alar to the spreadsheet when asked to choose between the two for daily use.

- This interface was easy to use.
- I was able to find the facts I needed to know.
- I was confident the facts I suggested led to the given incorrect fact being generated.
- I enjoyed using the interface.

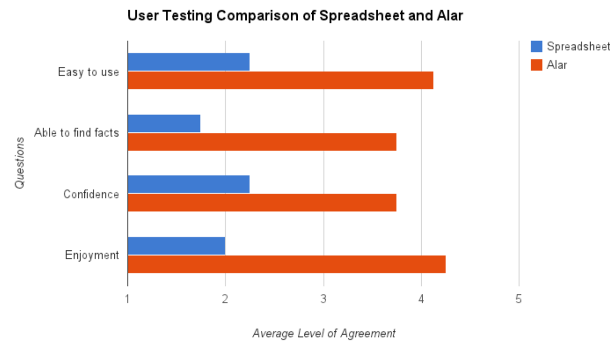

Figure 8. Usability evaluation of Alar vs. spreadsheet for finding potentially incorrect assertions.

Assertion Relatedness

Users would often search for the given assertion, and would subsequently examine assertions which made sense to them as being related. In this setting, assertion relatedness serves a good proxy for identifying assertions that led to the central assertion being inferred.

For the assertion used in user testing, "cup IsA drink," the 40 most related assertions were calculated. Then, starting with the most related, it would remove assertions from the KB and recalculate the inferred truth of "cup IsA drink." It then calculated how much the inferred truth of "cup IsA drink" changed, as a fraction of its original truth. For example, "something AtLocation cup" was found to be the most related assertion. So its entries in the original assertion matrix were set to 0, the inference process redone, the truth of "cup IsA drink" recomputed, and compared to the value originally found.

This was repeated for the forty most related assertions. At each step, the previously zeroed entries in the assertion matrix were left to be zero, so the results indicate the impact of removing all the first *n* related assertions, not just that particular one. This isn't a very viable option for a user interface because an SVD of a matrix of the scale used takes at least several seconds, and an SVD must be computed for each of the top assertions we are considering.

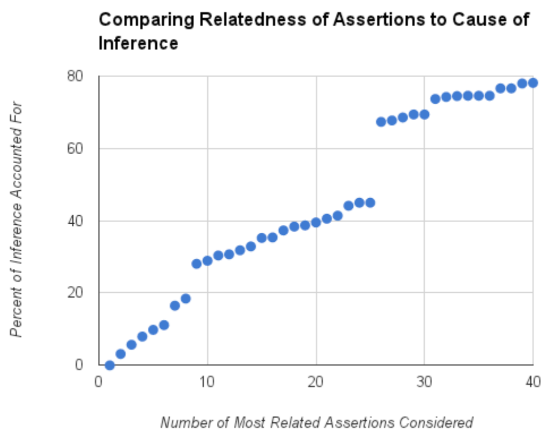

Figure 9. The relationship between relatedness and cause of inference between assertions for the assertion "cup IsA drink", which was used in user testing. It reveals, for the x most related assertions, what fraction of "cup IsA drink's" inferred truth value came from them.

In summary, the 40 most related assertions to "cup IsA drink" accounted for 78% of the inference. That is, it identified the 0.00000156% of the 2.5 billion assertions that could have potentially contributed to this inference. We feel this is grounds to claim that the relatedness metric in a good proxy by which users can identify assertions which may have caused other assertions to be inferred.

Related Work

There has been surprisingly little work in visualizing inference in AI.

In AI approaches that use logical inference, there has been work on visualizing the proof process itself. Most visualizations take the form of proof trees of the derivation of an assertion. Perhaps the best known of these is the Transparent Prolog Machine [Eisenstadt et al 91], a debugging tool for Prolog programs that visualizes the resolution proof procedure. It is especially useful for understanding backtracking. But few prior works treat the inference of sets of assertions not part of a inference chain.

Of course, there have been many visualizations of static structures, such as visualizations of ontologies. [Katifori et al 07] surveys such systems. A good example is Protegé [Protegé 14], which provides a gallery of visualizations, including node-and-arc diagrams, tag clouds, treemaps, and other views of hierarchical structures.

In statistical approaches, work has concentrated on visualizing the result of an inference process on a large data set, usually specific to the kind of data involved. But visualizing the output of an inference process sometimes doesn't deliver much insight into "what the machine is thinking". [Amershi et al 11] does organize sets of examples into visualization spaces, providing some insight into internal abstractions of learning algorithms. Amershi's work is one of the few to stress the importance of visual feedback in AI inference algorithms.

Visualizing control parameters of machine learning algorithms also helps, but requires nontrivial understanding of the algorithms. [Olden 02] visualizes neural networks; [Cossalter et al 11] and [Becker et al 01] visualize Bayesian networks; [Talbot et al 09] show a confusion matrix for ensemble machine learning methods.

Previous work in visualization specifically for ConceptNet and AnalogySpace appears in [Speer et al 10]. This computes "semantic dimensions" that represent a spectrum of distinctions between concepts, such as "good vs. bad". Each concept is a point in this multi-dimensional space, and it can be explored through a standard 3D fly-through visualization. It doesn't directly visualize assertions.

Conclusion

We have presented Alar, a visualization system for the ConceptNet knowledge base and AnalogySpace inference technique. It dynamically visualizes networks of concepts or assertions related by semantic similarity. It provides interactive control over the level of detail displayed. It uses a physics simulation visualization package to provide a smooth (and entertaining!) graphical presentation.

Alar is especially suited for graphical exploration of how "liberal vs. conservative" the inference is allowed to be, and of finding incorrect or missing assertions that might be causing expected inference to fail. Alar also makes a contribution in working out how to incrementally update a large-scale visualization of inference to maintain a lively interactive response.

We believe the present paper is one of the first works to address the problem of large-scale visualization of symbolic inference of multiple assertions. We hope that this paper will spark others to explore the very rich space of graphical and semantic possibilities for visualization of inference. We're not sure what the best way to do inference in AI really is, but maybe we'll know it when we see it.

References

Amershi, S., Fogarty, J., Kapoor, A., and Tan, D.(2011) Effective End-User Interaction with Machine Learning. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2011), Nectar Track, pp. 1529-1532.

Becker, B., Kohavi, R., and Sommerfield, D. Visualizing the simple Bayesian classifier. In Fayyad, U, Grinstein, G. and Wierse A. (Eds.) Information Visualization in Data Mining and Knowledge Discovery, (2001), 237-249

Bostock, M. Data Driven Documents, D3JS. http://www.d3js.org, February 2014.

Cossalter, Michelle, Ole J. Mengshoel, Ted Selker, Visualizing and Understanding Large-Scale Bayesian Networks, AAAI Workshop on Scalable Integration of Analytics and Visualization, San Francisco, California, USA, 2011.

Havasi, Catherine, James Pustejovsky, Robert Speer, and Henry Lieberman, Digital Intuition: Applying Common Sense Using Dimensionality Reduction, IEEE Intelligent Systems 24(4), Special Issue on Human-Level Intelligence, pp. 24-35, July 2009.

Havasi, Catherine, Robert Speer and James Pustejovsky, Coarse Word-Sense Disambiguation Using Common Sense. AAAI 2010 Symposium on Common Sense Knowledge.

Eisenstadt, Marc, Mike Brayshaw, Jocelyn Paine, The Transparent Prolog Machine: Visualizing Logic Problems, Springer, 1991

Katifori, A., Halatsis, C., Lepouras, G., Vassilakis, C., and Giannopoulou, E. 2007. Ontology visualization methods—A survey. ACM Comput. Surv. 39 , 4, Article 10 (October 2007).

Lenat, Douglas, CYC: a large-scale investment in knowledge infrastructure, Communications of the ACM, Volume 38 Issue 11, Nov. 1995 Pages 33-38

Lin, Xia, Yen Bui, Dongming Zhang, Visualization of Knowledge Structures, International Conference on Information Visualization, 2007.

Olden, J.D., Jackson, D.A. 2002. Illuminating the 'black-box': a randomization approach for understanding variable contributions in artificial neural networks. Ecological Modelling. 154:135-150.

Protege Project, http://www.protege.stanford.edu. Protege's visualization facilites are described in http://protegewiki.stanford.edu/wiki/Visualization, accessed February 2014.

Speer, Robert, Catherine Havasi and Henry Lieberman, AnalogySpace: Reducing the Dimensionality of Commonsense Knowledge, Conference of the Association for the Advancement of Artificial Intelligence (AAAI-08), Chicago, July 2008.

Speer, Robert, Catherine Havasi, Nichole Treadway, and Henry Lieberman, Finding Your Way in a Multi-dimensional Semantic Space with Luminoso, ACM International Conference on Intelligent User Interfaces (IUI), Hong Kong, China, February 2010, pp. 385-388.

Talbot, J., Lee, B., Kapoor, A., & Tan, D. S. EnsembleMatrix: interactive visualization to support machine learning with multiple classifiers. In Proceedings of the 27th international conference on Human factors in computing systems (pp. 1283-1292), 2009.