INFOBOX INTERLINKER

A Wikipedia tool to visualize the Semantic Web

By Matias I. B. Oddo
moddo@eoas.ubc.ca

The University of British Columbia
Department of Computer Science

November 15th, 2022

INTRODUCTION

This project started out of my interest in unsupervised, emergent structures in data. The Semantic Web, sometimes known as the 3.0 Web, has the goal of making the Internet data machine-readable. Tim Berners-Lee, creator of the Internet, originally expressed his vision of the Semantic Web in 1999 as follows:

"I have a dream for the Web [in which computers] become capable of analyzing all the data on the Web – the content, links, and transactions between people and computers. A "Semantic Web", which makes this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy and our daily lives will be handled by machines talking to machines. The "intelligent agents" people have touted for ages will finally materialize."

Xia et al (2021) note that information corpora such as Wikipedia and Baidu Bake have grown to be de-facto repositories of modern encyclopedic knowledge. Efforts to extract knowledge networks, which are data structures intuitively imagined as vast node-link graphs, have been underway over the past decades. Popular knowledge network databases that have computerized corpora are Freebase, YAGO, Wikidata, and BDpedia. At the core of these knowledge networks are RDF triples, defined and standardized in 2004 by W3.org (Figure 1).

Figure 1. An illustrative example of an Resource Description Framework (RDF) triplet.

Xia et al (2021) highlight that DBpedia is now the de-facto knowledge network of Wikipedia. Since its inception in 2007, the DBpedia project has been continuously releasing large, open datasets, extracted from Wikimedia projects such as Wikipedia and Wikidata (Hofer et al, 2020). Xia et al (2021) identify two types of RDF triplet. On one hand there are "assertion triplets", which state facts about the subject, and on the other "ontological triples", which are more complex and form the schema of the network structure. I propose a third type of RDF triplet, one that can be recursively scraped. This is only possible when the predicate is redundant and the object is the same type as the subject. For example, if the predicate is "Influenced", which is a semantic connection, then an emergent deductive network can be extracted purely from the interlinking of these "Influenced" RDFs. This kind of triplet is present in Wikipedia's front-end facing infobox data template.

According to Wikipedia's infobox template page, "the use of infoboxes is neither required nor prohibited for any article. Whether to include an infobox, which infobox to include, and which parts of the infobox to use, are determined through discussion and consensus among the editors at each individual article." [1] The only consistency across infoboxes is that when they are present they have the same HTML tag. It is up to community editors to ensure the content matches across pages of the same theme or genre. That said, there are efforts to establish templates for infoboxes, and these templates sometimes include semantically interlinked parameters. Figure 2 illustrates how infoboxes exist at the overlap of human-friendly format and machine-readable data from Wikipedia's knowledge corpora.

[1] https://en.wikipedia.org/wiki/Category:Infobox_templates

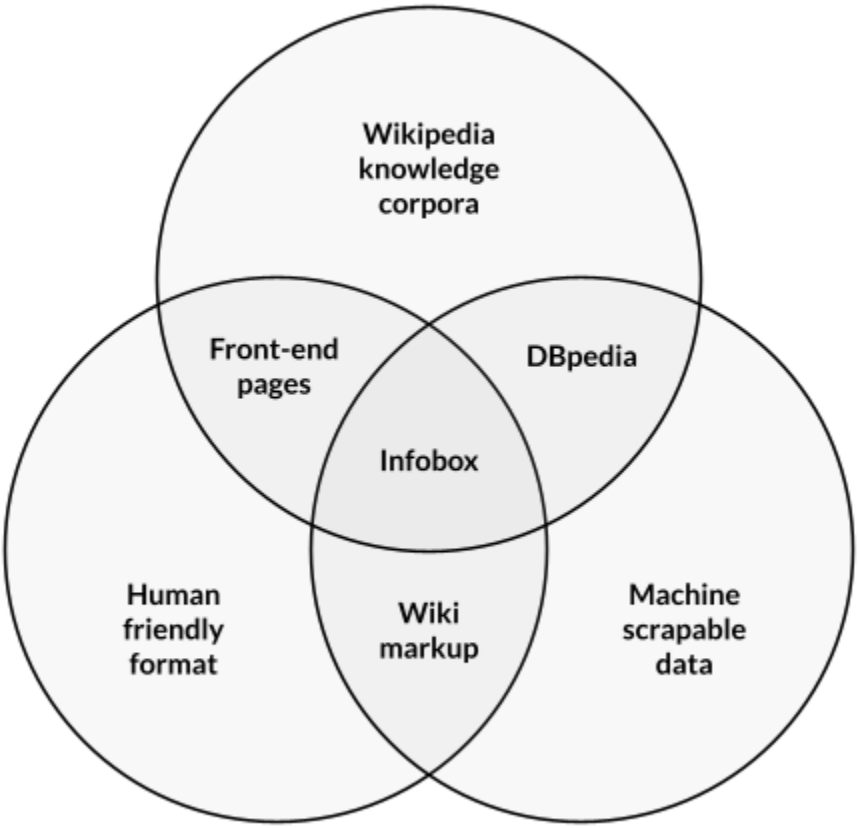

Figure 2. Wikipedia's infobox is at the intersection of machine-scrapable data and Wikipedia's knowledge corpora, and yet it is also present in Wikipedia's front-end with a human-friendly format.

Many noteworthy programming languages have an infobox that follows the template in Figure 3, with the parameters "Influenced by" and "Influenced" at the bottom of the box. For the Plankalkül page the listed hyperlinks link to the Begriffsschrift, Superplan, and ALGOL 58 pages. Recursively scraping all of these links crosscuts through thousands of Wikipedia pages, resulting in a collection of pages that effectively forms a network with a shared theme. Wu and Weld (2008) coined that the genre or theme that underpins this network as "class", and thus infoboxes have "classes" that they ontologically belong to.

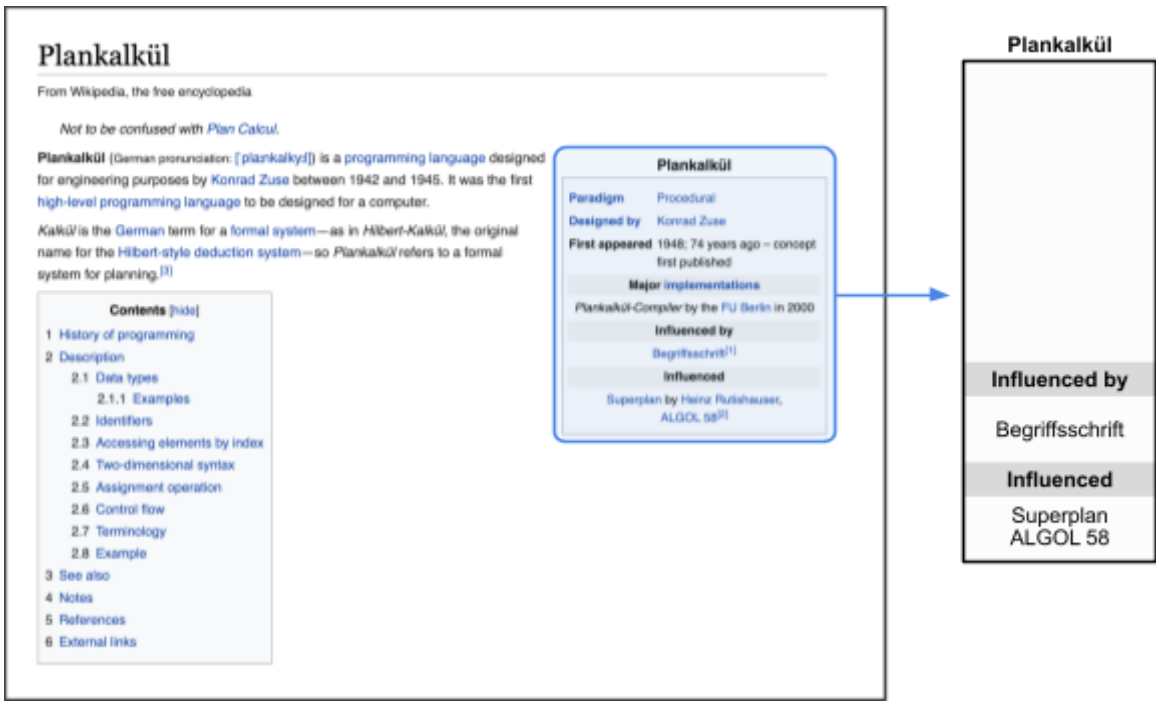

Figure 3. Wikipedia page for Plankalkül, the first high-level programming language. Highlighted in blue is the page's infobox, and extracted to the right a simplified version with key parameters and hyperlinks.

This infobox "class" network is the core dataset of this project, but it must be created first. I wrote a Python scraper [2] for front-end Wikipedia infoboxes, extracting first the subject (main page name), then two predicates ("Influenced" and "Influenced by"), and all the objects listed under each predicate (the listed page names). The scraper is recursive, so in the next step an object (one of the listed pages) becomes the subject (now a main page, with its own infobox different from the already scraped one), and the process continues until the entire "class" network is extracted.

Figure 4 shows a simplified illustrative example of how five infoboxes are interlinked by this special kind of RDF triplet. It is important to note that the recursive scraper will extract the entire "class" network no matter which page it starts from (Figure 5). In the Data section is a summary of two networks extracted with this method, first for programming languages with the starting page "Fortran", and then a much larger (16x) philosophers network with the starting page "Carl_Jung".

I propose a visualization tool that helps Wikipedia editors tap into class-level knowledge networks to understand 1. Understand the structure of the class-level knowledge network 2. Deductive interlinking to fill gaps in infoboxes 3. Ensure template cohesion across a thematic class.

[2] https://github.com/diredіredock/infobox_interlinker

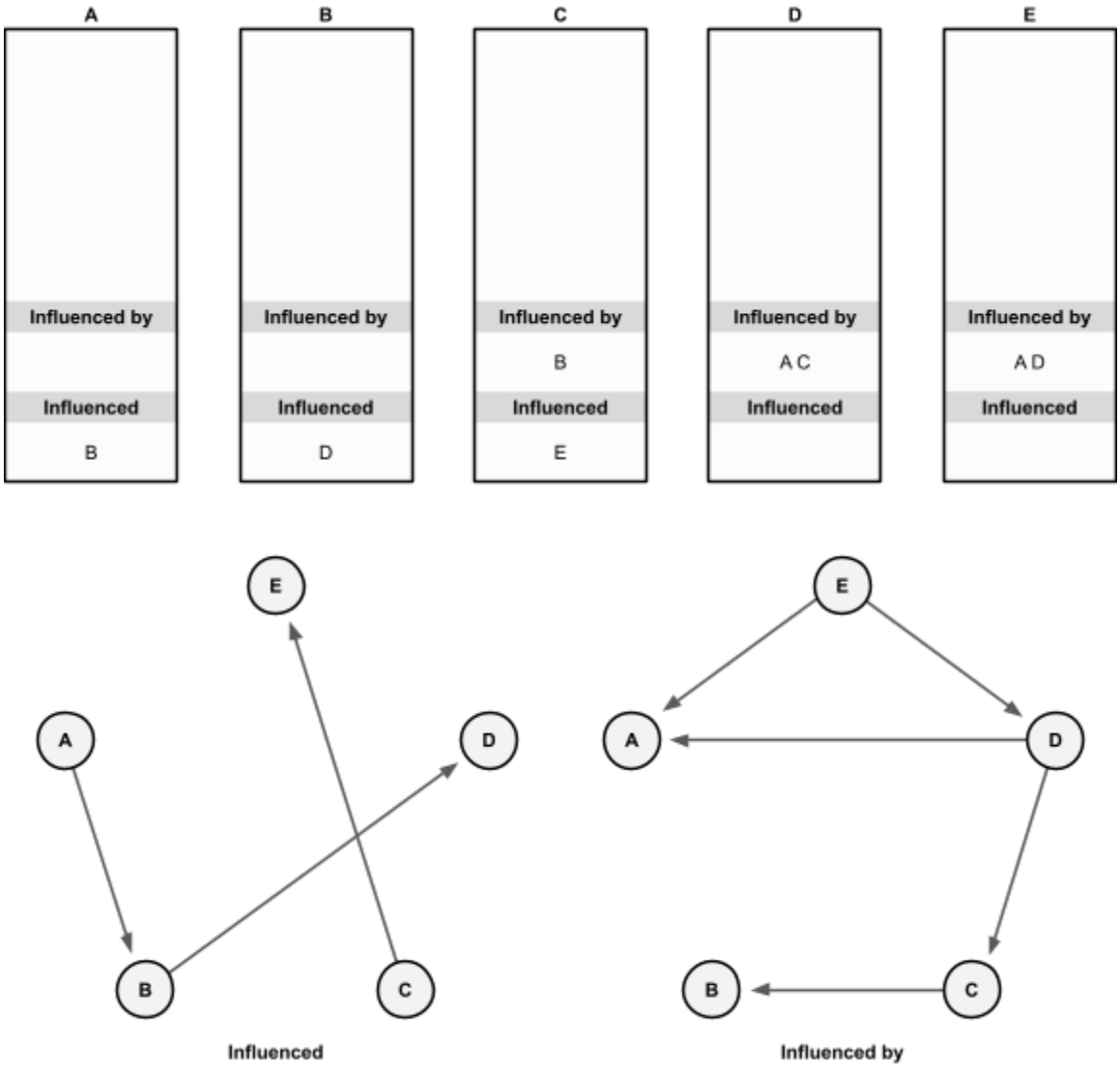

Figure 4. In this example only one hyperlink connection exists between any two node pairs. Given the globally decentralized and community-driven authorship of Wikipedia, this example is a representation of the current state of infobox interlinking.

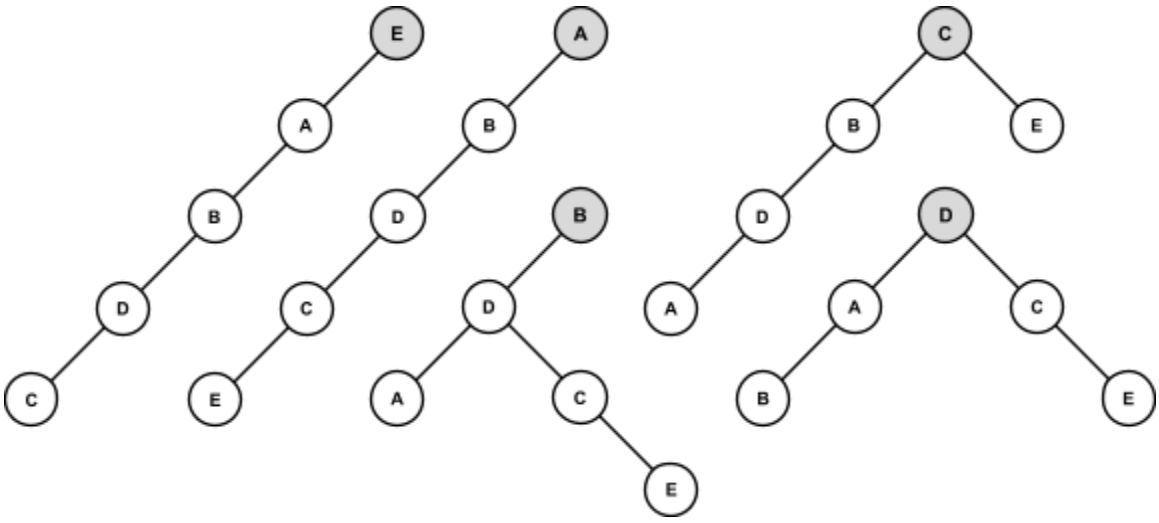

Figure 5. Scraping with a unique-key dictionary storage. All the recursive call tree examples for the example infoboxes.

RELATED WORK

There are extensive examples of network visualization attempts to visualize the web, almost all of them of the node-link type. In effect, Abdelaal et al (2022) recognize that 80% of studies are conducted in networks with less than 100 nodes, and 68% on undirected networks. This project has large networks (500+ nodes) and directed graphs, so this section has to be revised.

Table 1. Some network visualization projects based on Wikipedia data.

Wikimaps	http://www.ickn.org/wikimaps/
Tony Hirst	SPARQL querying DBpedia through Gephi to extract a network (2012) https://blog.ouseful.info/2012/07/03/visualising-related-entries-in-wikipedia-using-gephi/ Musical genres (2012) https://blog.ouseful.info/2012/07/04/mapping-related-musical-genres-on-wikipediadbpedia-with-gephi/ Programming languages (2012) https://blog.ouseful.info/2012/07/03/mapping-how-programming-languages-influenced-each-other-according-to-wikipedia/ Goth bands (2016) https://blog.ouseful.info/2016/03/31/semantic-cartography-mapping-bands-by-common-members-using-wikipedia-data/
Grant Louis Oliveira	https://kumu.io/GOliveira/philosophers-web#map-b9Ts7W5r https://grantoliveira.wordpress.com/2017/01/09/social-network-visualization-a-history-of-philosophy/ https://dailynous.com/2017/01/11/visualization-influence-history-philosophy/

```
"Fortran"

      Nodes
               total                   577
                  with infobox         471
                  without infobox      106
      Edges
               incoming graph         1106
               outgoing graph         1007

      Density
               incoming/nodes         1.92
               outgoing/nodes         1.75

"Carl_Jung"

      Nodes
               total                  8765
                  with infobox        6908
                  without infobox     1857
      Edges
               incoming graph        16275
               outgoing graph        13633

      Density
               incoming/nodes         1.86
               outgoing/nodes         1.56
```

Figure 6. Summary of the two extracted datasets, the “class” network for Carl_Jung is 16 times bigger than Fortran.

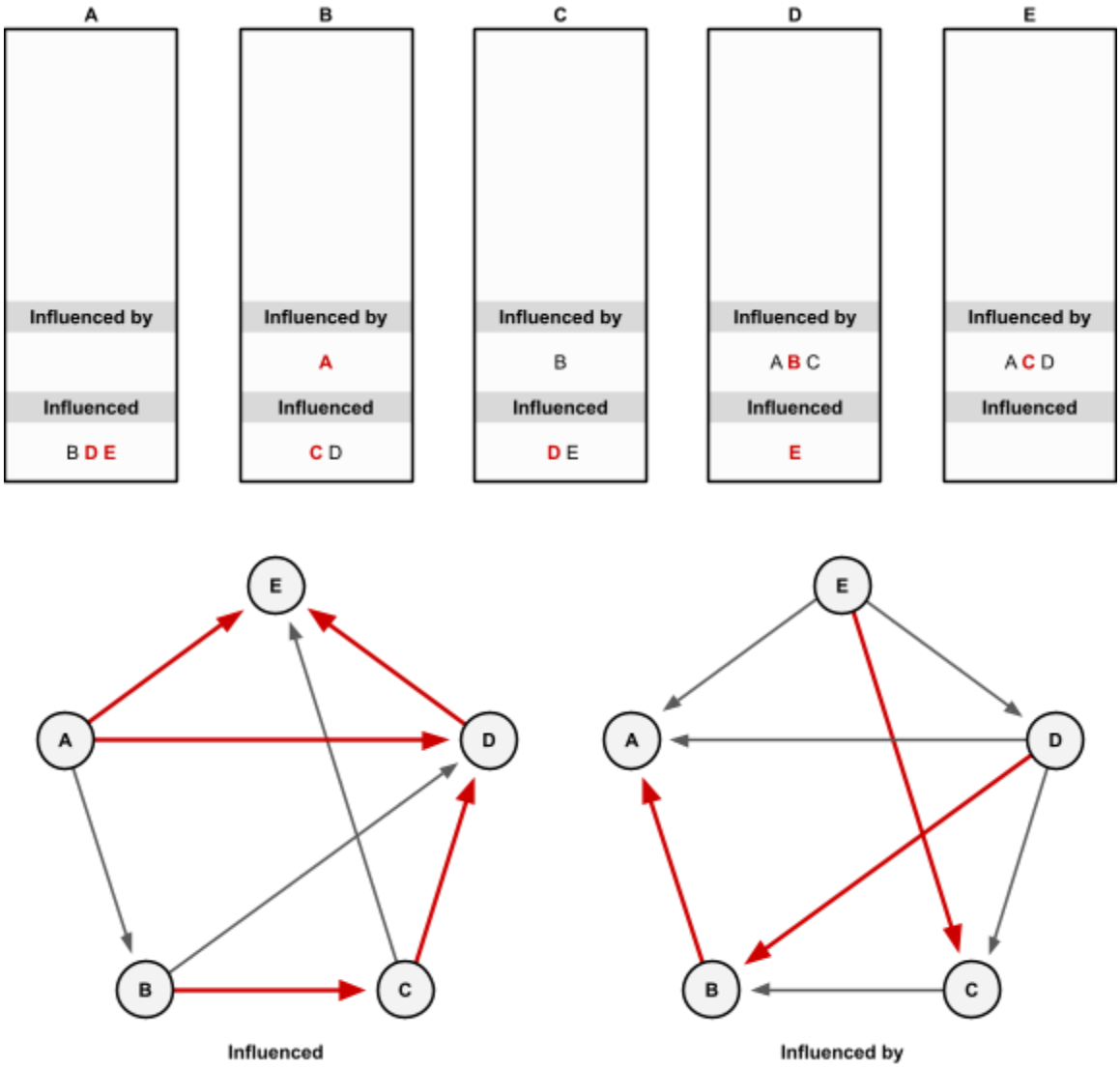

Figure 7. Illustrative node-link view of deductive interlinking, where the class network can aid a Wikipedia editor complete missing information in infoboxes.

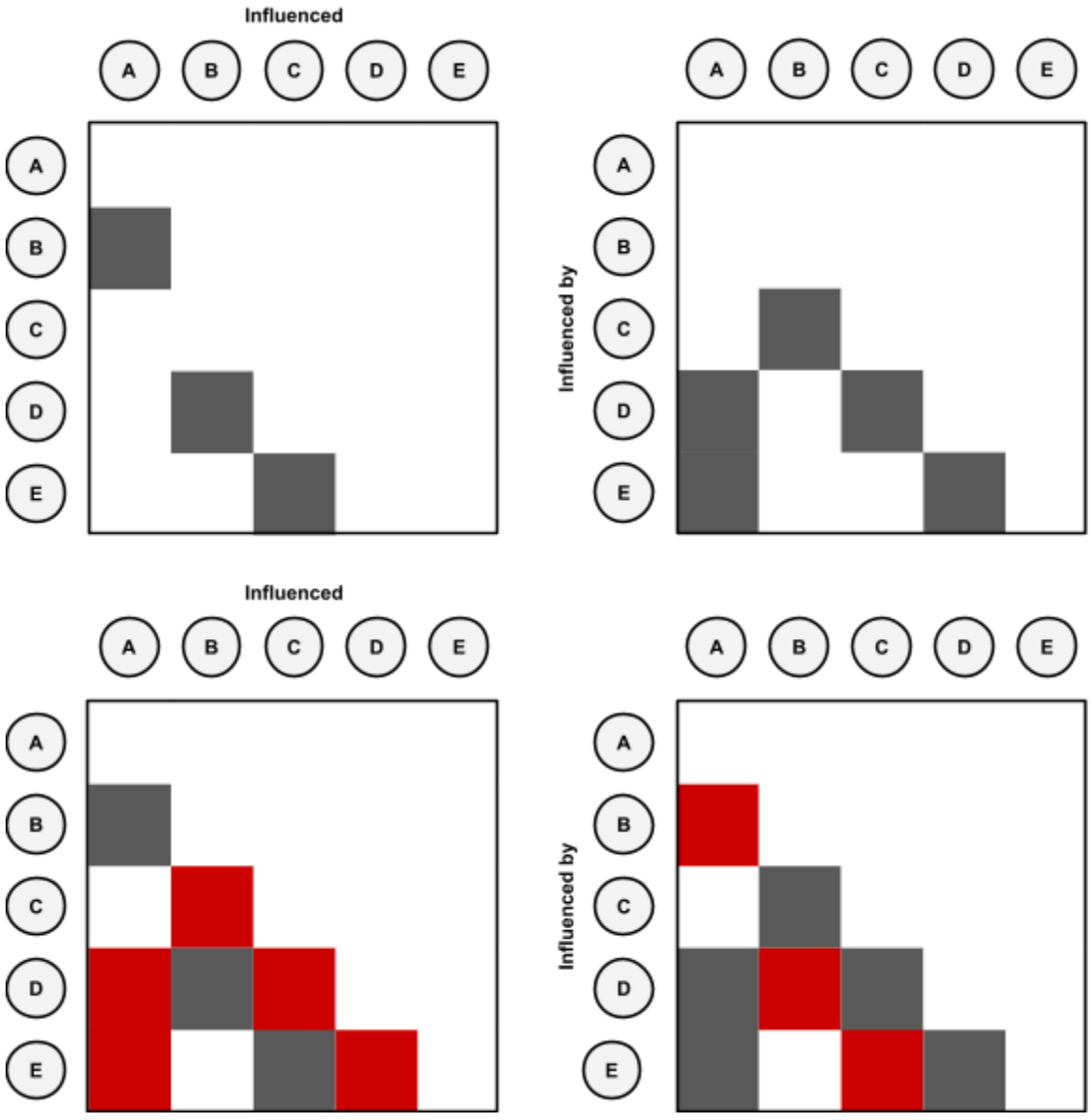

Figure 7. Illustrative matrix view of deductive interlinking. Note that the “Influenced by” matrices have to transposed, because it is the inverse of “Influenced by” that directly overlaps the “Influenced” matrix and allows deductive interlinking.

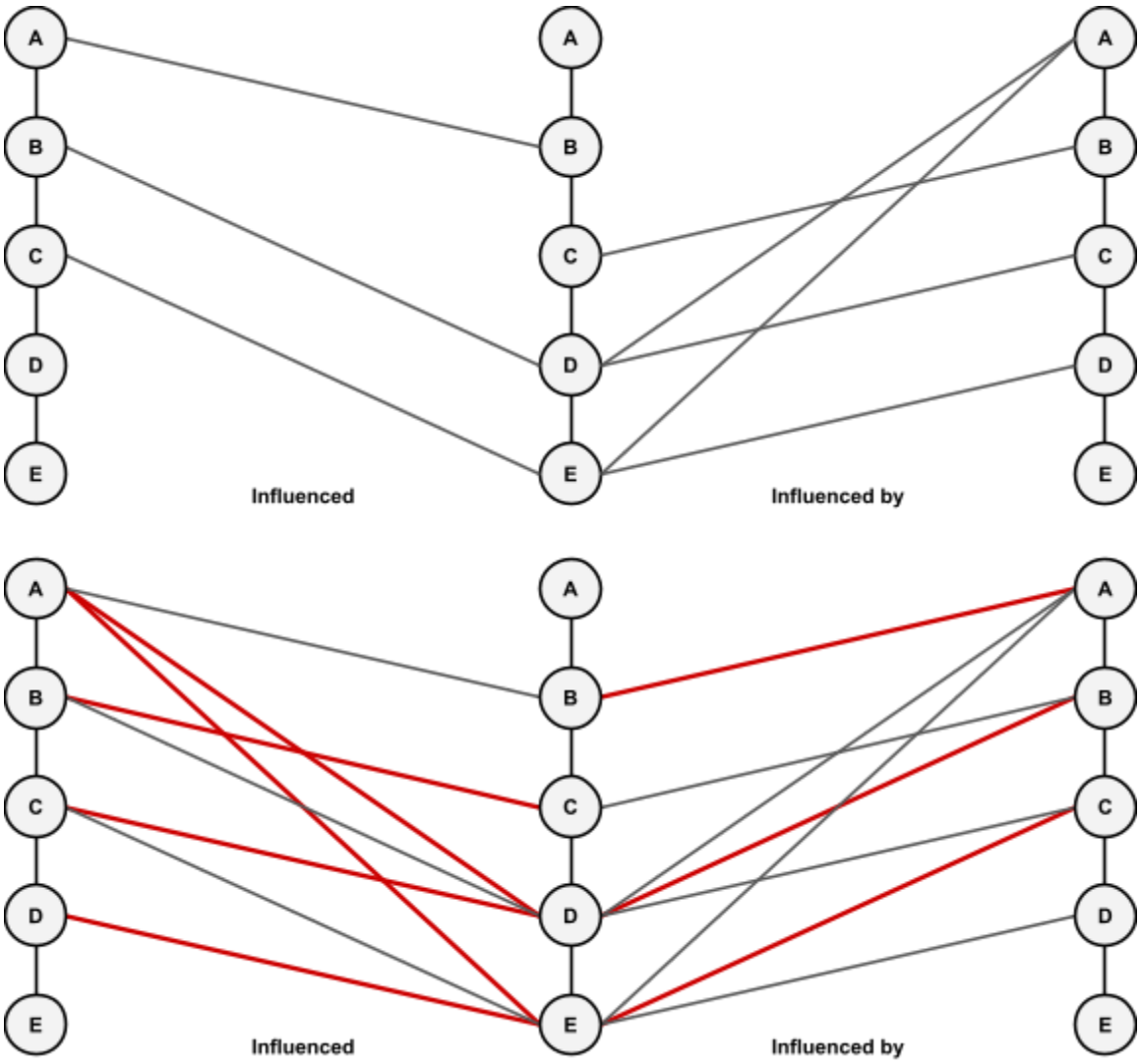

Figure 8. Illustrative bipartite graph view of deductive interlinking.

SOLUTION

Following the findings of Abdelal et al (2022) that large networks (500+ nodes) require more overview tasks, I propose a symmetric bipartite graph (Figure 8) following Burch et al (2011) pixel rendering. By performing deductive interlinking edge cluttering, which is one of the main hurdles of bipartite graphs, can be alleviated so that the Wikipedia editor has a better pulse of the semantic health of the class network at hand.

RESULTS

None yet.

REFERENCES

Hofer, Hellmann, Dojchinovski and Johannes, 2020. The new DBpedia release cycle: Increasing agility and efficiency in knowledge extraction workflows. International Conference on Semantic Systems.

Wu and Weld, 2008. Automatically Refining the Wikipedia Infobox Ontology. WWW '08: Proceedings of the 17th international conference on the World Wide Web.

Xia et al, 2021. KTabulator: Interactive Ad hoc Table Creation using Knowledge Graphs.

W3.org, 2004. Resource Description Framework (RDF): Concepts and Abstract Syntax. https://www.w3.org/TR/rdf-concepts/

Abdelaal et al, 2022. Comparative evaluation of bipartite, node-link, and matrix-based network representations.

MILESTONES

A total of 80 hours allocated to 14 weeks is 5.7 hours a week (Thursday mornings).

- Week 01 - Sep 12 to Sep 16
  - Started work on recursive scraper and JSON format for data export.
- Week 02 - Sep 19 to Sep 23
  - Scraper and JSON data export, complete scraping of programming language infoboxes.
- Week 03 - Sep 26 to Sep 30
  - Tentative plan is to study the network structure of programming languages according to Wikipedia.
  - [DELIVERABLE] Project Pitch due September 28 at noon.
- Week 04 - Oct 03 to Oct 07
  - Data analysis of cardinality and properties of programming language directed graphs.
  - [FEEDBACK] Important to focus on tasks, warning about investing in a dead end project path.
- Week 05 - Oct 10 to Oct 14
  - No coding or data analysis, only focus on how to pivot towards a design study.
- Week 06 - Oct 17 to Oct 21
  - Design study based on a user (Wikipedia editor) and a specific task (improving infoboxes).
  - [DELIVERABLE] Project Proposal (this document) due October 21 at noon.
- Week 07 - Oct 24 to Oct 28
  - Genericizing the scraper so it works with more topics, not just programming languages.
  - Scrape JSON datasets from other infobox classes (e.g. biologists, artists, physicists, philosophers).
- Week 08 - Oct 31 to Nov 04
  - Knowing of interesting cases such as pages with infobox missing or hyperlink to page subsection, look for more situations like these that an editor would be interested in knowing about.
- Week 09 - Nov 07 to Nov 11 [READING WEEK]
  - Finalize datasets for “Fortran” and “Carl_Jung”
- Week 10 - Nov 14 to Nov 18
  - Update written report.
  - [DELIVERABLE] Project Update due November 15 at 3pm.
- Week 11 - Nov 21 to Nov 25
  - Work on medium fidelity prototype of system.
- Week 12 - Nov 28 to Dec 02
  - Work on Final Report.
- Week 13 - Dec 05 to Dec 09
  - Complete Final Report.
- Week 14 - Dec 12 to Dec 16
  - [DELIVERABLE] Final Report due December 16 at 8pm.