INFOBOX INTERLINKER

A Wikipedia editing tool to improve the Semantic Web

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

The University of British Columbia
Department of Computer Science

October 21st, 2022

INTRODUCTION

This project started out of my interest in unsupervised, emergent structures in data. Wikipedia is a goldmine of information and a killer app of the 2.0 Web. The Semantic Web, sometimes known as the 3.0 Web, has the goal of making the Internet data machine-readable. [1] 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.*

Some features of Wikipedia already exhibit this underlying semantic structure. For example, the Wikipedia infobox is a structured format that lists key information about a page. Sometimes this information links to other pages, which also have infoboxes, that in turn link to other pages, also with infoboxes, and so on. This simple scheme creates an emergent network of both technically linked (via hyperlinks) and semantically linked (via meaning) digital content. I think this is what makes Wikipedia a killer app, the fact that technical links directly map to meaningful knowledge links.

I started this project with the interest of exploring the emergent structure behind the interlinked Wikipeda pages of programming languages. I built a Python recursive scraper that extracted infobox data, about 500 pages and 1600 links. [2] In doing so I discovered that a lot of infoboxes have missing data, but also that this data can be deduced from the entire network, and then added where it should be.

Making infoboxes completely semantically interlinked at the front-end Wikipedia level impacts the quality of information that modern Semantic Web projects such as DBpedia use to form their database. Therefore, a tool that improves semantic interlinking also improves the Semantic Web.

[1] https://en.wikipedia.org/wiki/Semantic_Web
[2] https://github.com/direditedock/wikigraph_infovis

RELATED WORK

Wikipedia has a standard called the infobox (Figure 1), which structures key information from a page into a recurring, easy-to-access format. 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." [3] 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. To illustrate, the infobox template for "scientist" [4] can have up to six interlinking parameters, of which three are incoming and three are outgoing:

1. "Doctoral advisor", an incoming semantic parameter.
2. "Other academic advisors", an incoming semantic parameter.
3. "Doctoral students", an outgoing semantic parameter.
4. "Other notable students", an outgoing semantic parameter.
5. "Influences", an incoming semantic parameter.
6. "Influenced", an outgoing semantic parameter.

A more illustrative example is the template of the "programming languages" infobox, which only has two interlinking parameters (Table 1). The example of Table 1 also serves as the general semantic case, one that maps to the directed graph network idiom, where a mid-network node (a Wikipedia page with infobox) has one or more incoming edges, and one or more outgoing edges.

Table 1. Parameter descriptions from /wiki/Template:Infobox_programming_language.

Influenced by	Name of notable concepts, methodologies, approaches, or practices that influenced the creation of the subject of the infobox.
Influenced	Name of notable concepts, methodologies, approaches, or practices that were created under the influence of the subject of the infobox.

[3] https://en.wikipedia.org/wiki/Category:Infobox_templates
[4] https://en.wikipedia.org/wiki/Template:Infobox_scientist

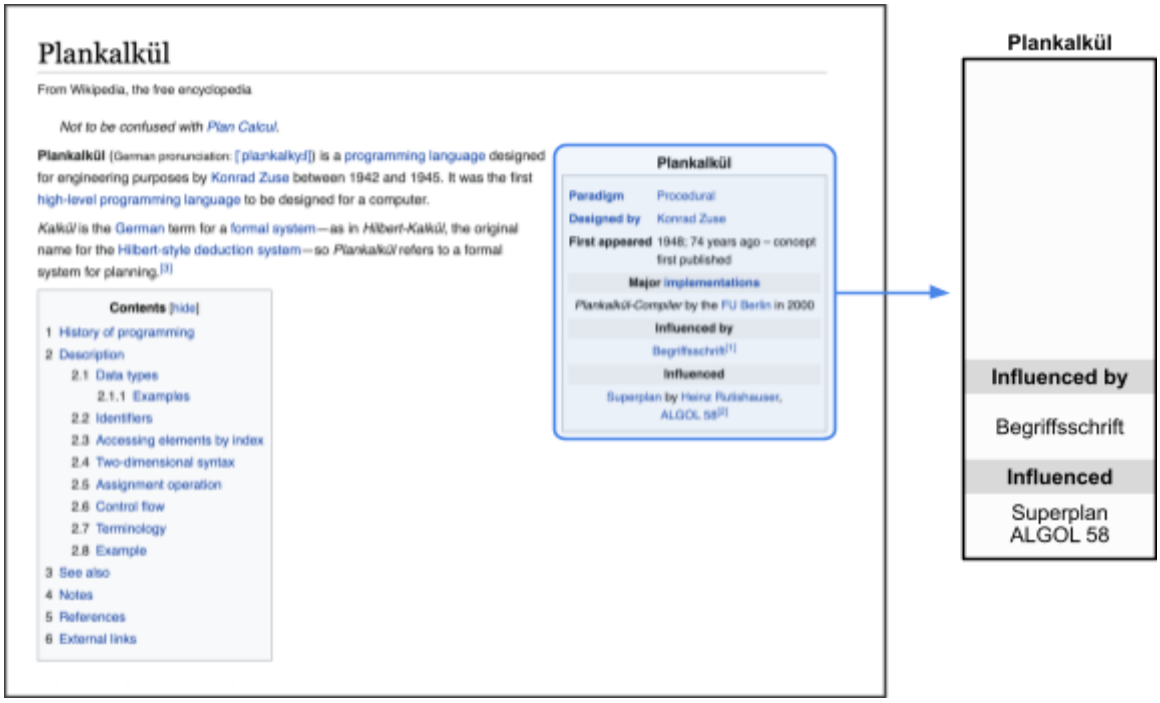

Figure 1. Wikipedia page for Plankalkül, the first high-level programming language. Highlighted in blue is the page's infobox, and extracted to the right a simplified version with key parameters and hyperlinks.

Many noteworthy programming languages have an infobox that follows the template in Table 1 and Figure 1, 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. Belonging to a class has an impact on the infobox's format and content. In practice infoboxes are highly similar within classes, and highly dissimilar between classes.

Recursive scraping does have limitations. For a page to be discovered by the scraper, at least one hyperlink for itself must be present in an already scraped infobox. This means that some important information can be left out because it escapes the infobox class format. This is the case of the Wikipedia page for Ada Lovelace, the first computer programmer, which does have information but not an infobox dedicated to the manuscript of the first computer program. In the context of the history of programming languages, this first program should have a standalone page and a dedicated "programming language" style infobox, so that it is semantically interlinked. This is not the case, currently this foundational piece of knowledge cannot be reached by the scraper, but it can be in the future if semantically interlinked.

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). DBpedia uses structured content extracted from infoboxes by machine learning algorithms to create a resource of linked data in the Semantic Web. It has been described by Tim Berners-Lee as "one of the more famous" components of the linked data project. [5] Regarding node-link network visualizations built on scraped Wikipedia semantic links and DBpedia, there are many examples (Table 2).

Table 2. Some network visualization projects based on Wikipedia data.

Wikimaps	http://www.ickn.org/wikimaps/
Tim 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/

A weakness of DBpedia is that it is not a real-time snapshot of the current state of Wikipedia. The database used to be extracted from full Wikipedia downloads every 12 to 17 month cycles, but a recent re-working of the pipeline focusing on agility and efficiency has the database updated and validated every month (Hofer et al, 2020). However, from a Wikipedia editor's perspective, it is more important to capture the current front-end state of Wikipedia infobox semantic health to triage pages that need editing.

[5] https://en.wikipedia.org/wiki/Infobox

DATA

Figure 3 and Figure 4 illustrate the best and worst case scenarios involving infobox semantic interlinking. Real-world and real-time recursively scraped data from Wikipedia exists within these two extremes.

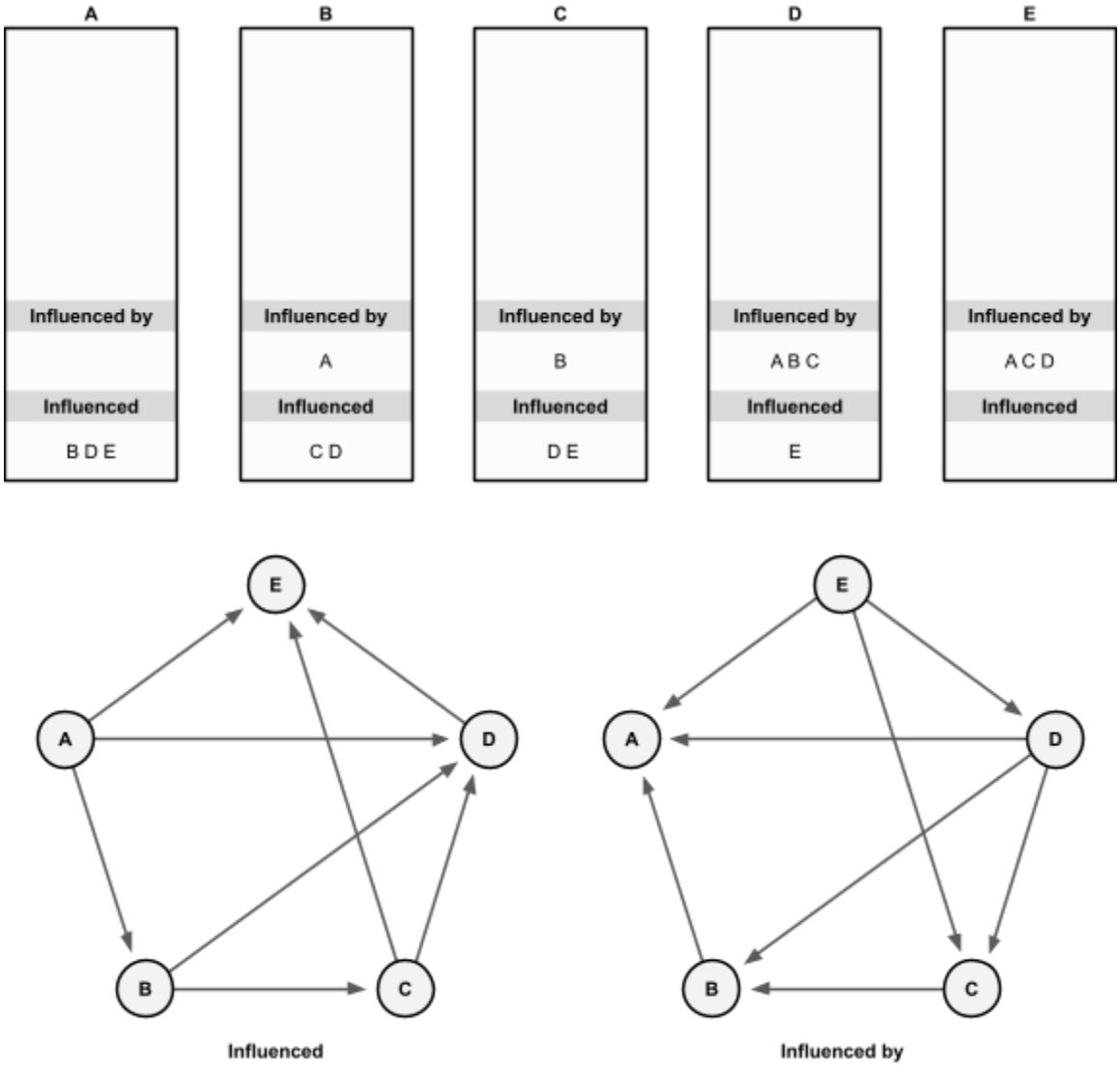

Figure 2. An illustrative example of complete semantic interlinking, the best case scenario. In this example, two node-link graphs can be extracted from five infoboxes (titled A, B, C, D, and E). One graph for all the "Influenced" links (left) and another graph for all the "Influenced by" links (right). The direction of the arrow means page redirection, marking a directed hyperlink connection. In this best case scenario every hyperlink connection in the left graph maps directly to the inverse of the right graph, and vice versa. This complementary mapping of hyperlinks is the technical underpinning of semantic interlinking.

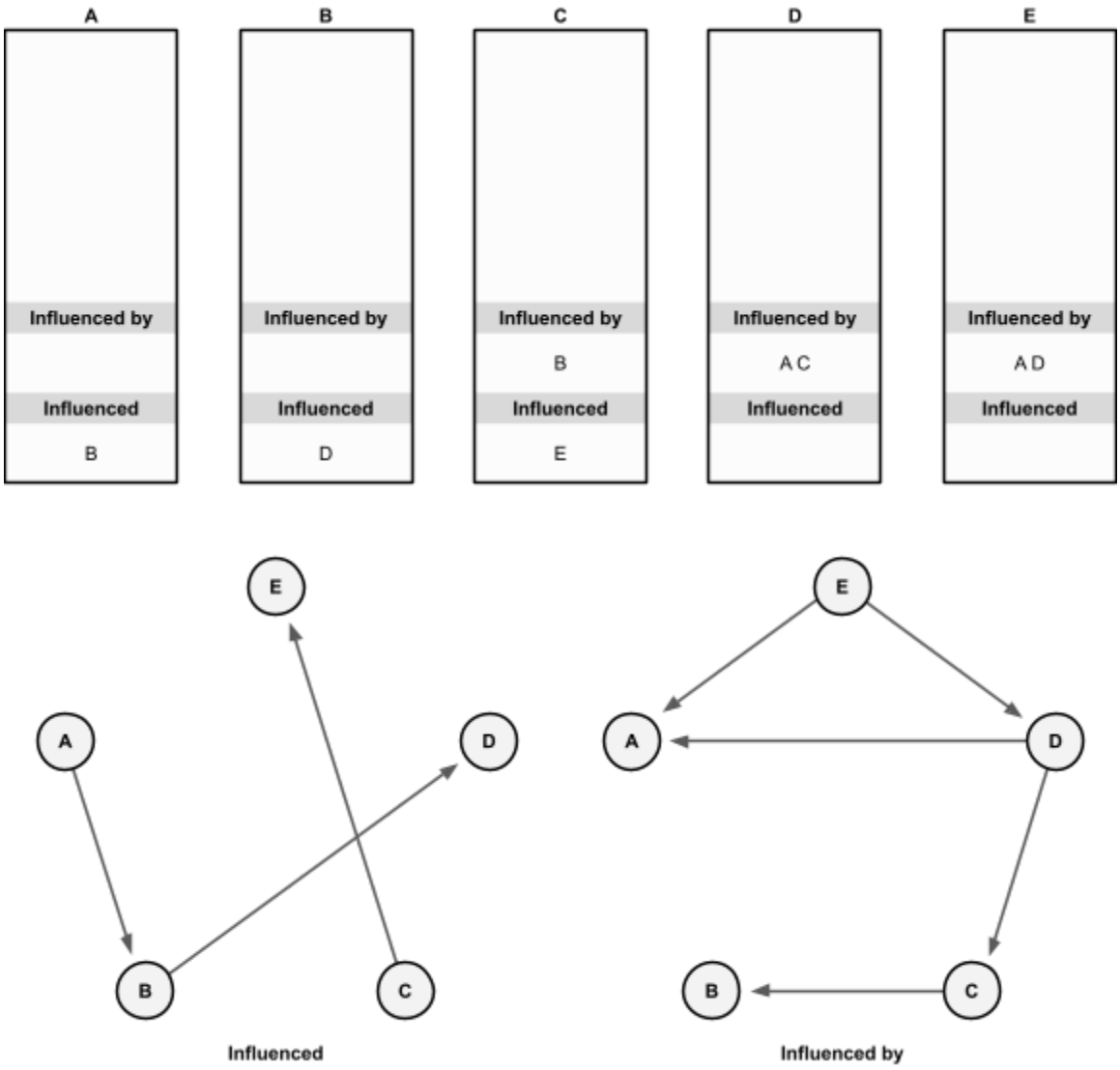

Figure 3. An illustrative example of zero semantic interlinking, the worst case scenario. In this example only one hyperlink connection exists between any two node pairs. Given the globally decentralized and community-driven authorship of Wikipedia, this example is a reasonable representation of the current state of infobox semantic interlinking.

TASK ABSTRACTION

The visualization solution “Infobox Interlinker” uses the deduced structure of the completely semantically interlinked network to suggest missing infobox hyperlinks (Figure 4). This is not automated because there is nuance that the Wikipedia editor has to consider, the solution helps the editor effectively navigate missing content through visual highlights, ranked sorting, and markings for special cases (Figure 5).

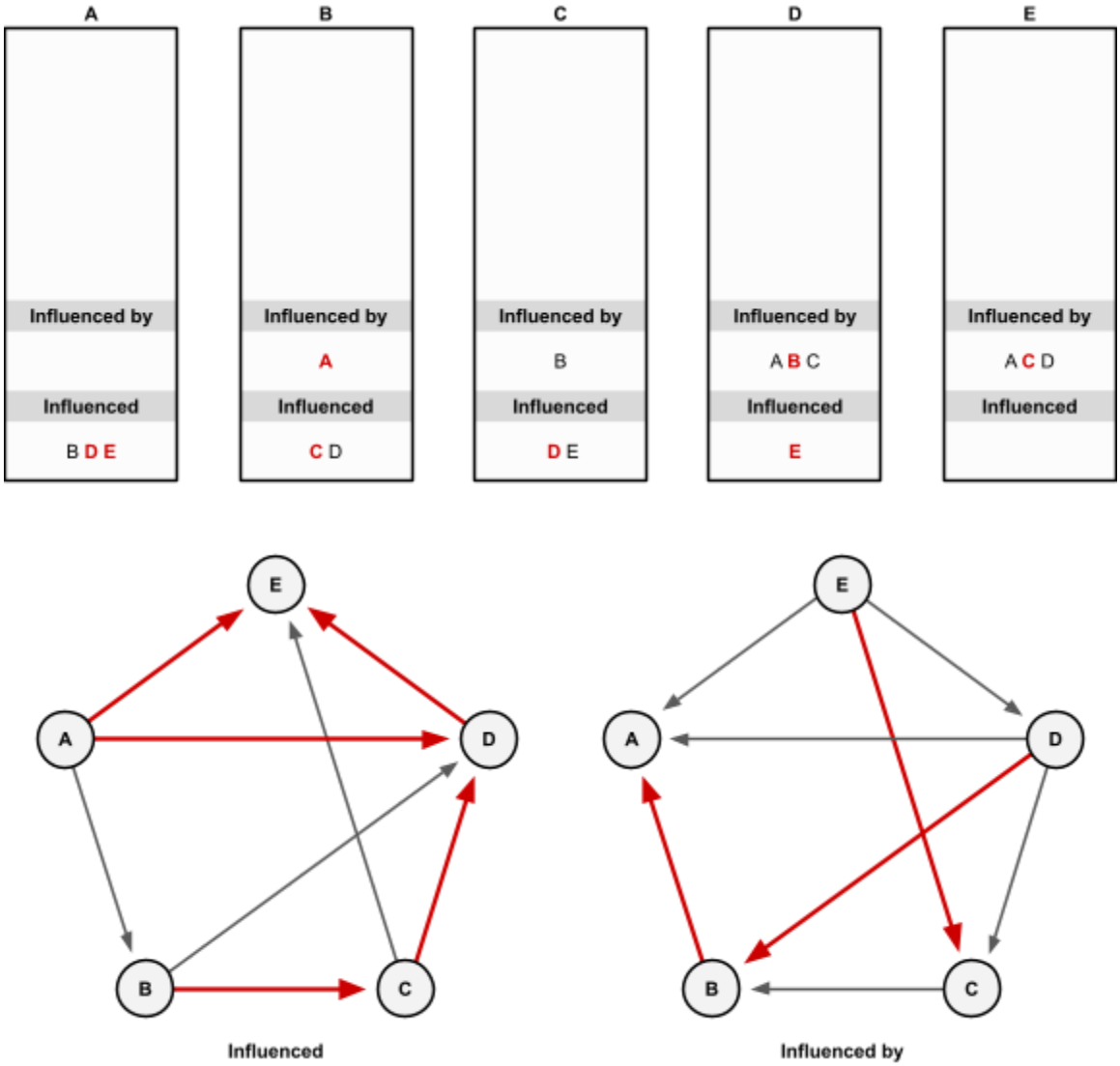

Figure 4. An illustrative example of the steps involved in solving for complete semantic interlinking. Here the bold red arrows and hyperlink text letters highlight all the deduced additions necessary to achieve semantic interlinking between infoboxes A, B, C, D, and E.

SOLUTION

Following the WHAT-WHY-HOW model for information visualization systems, we can ask:

WHAT is the solution? A node-link diagram extracted from a recursive scraper that traverses hyperlinks to or from at least one infobox in a Wikipedia page. Each node is a unique hyperlink (href) for a Wikipedia page. In addition to the node-link diagram, a per-node pop-up has specific highlights, chiefly the missing hyperlinks and other markings that aid in editing. Figure 5 illustrates a draft prototype of the solution.

WHY is this a solution? The task is to find pages with infoboxes that need semantic interlinking. A node-link graph shows all hyperlinked-connected pages once. A node-link graph with visual minimizing of nodes that are already interlinked and simultaneous highlighting of nodes that need fixing helps the editor triage where to focus editing resources to achieve format consistency, data standardization, and semantic interlinking.

HOW is the solution deployed? The node-link graph is manipulated by panning and selecting. A supplementary ranked list of nodes can be filtered and sorted by the editor to identify and triage which pages need fixing, the node-link graph showing these pages in tandem. Pages (nodes) that need semantic interlinking are highlighted by red (no highlight means no fix is needed) and size encoding, the bigger the node the more missing hyperlinks. Dataset cardinality is variable, it depends on the original seed page (where recursion starts), infobox class, and theme/genre within Wikipedia.

The goal is to add hyperlinks to infoboxes so that these markings vanish. When all highlight encondings vanish, success! The entire network is completely semantically interlinked.

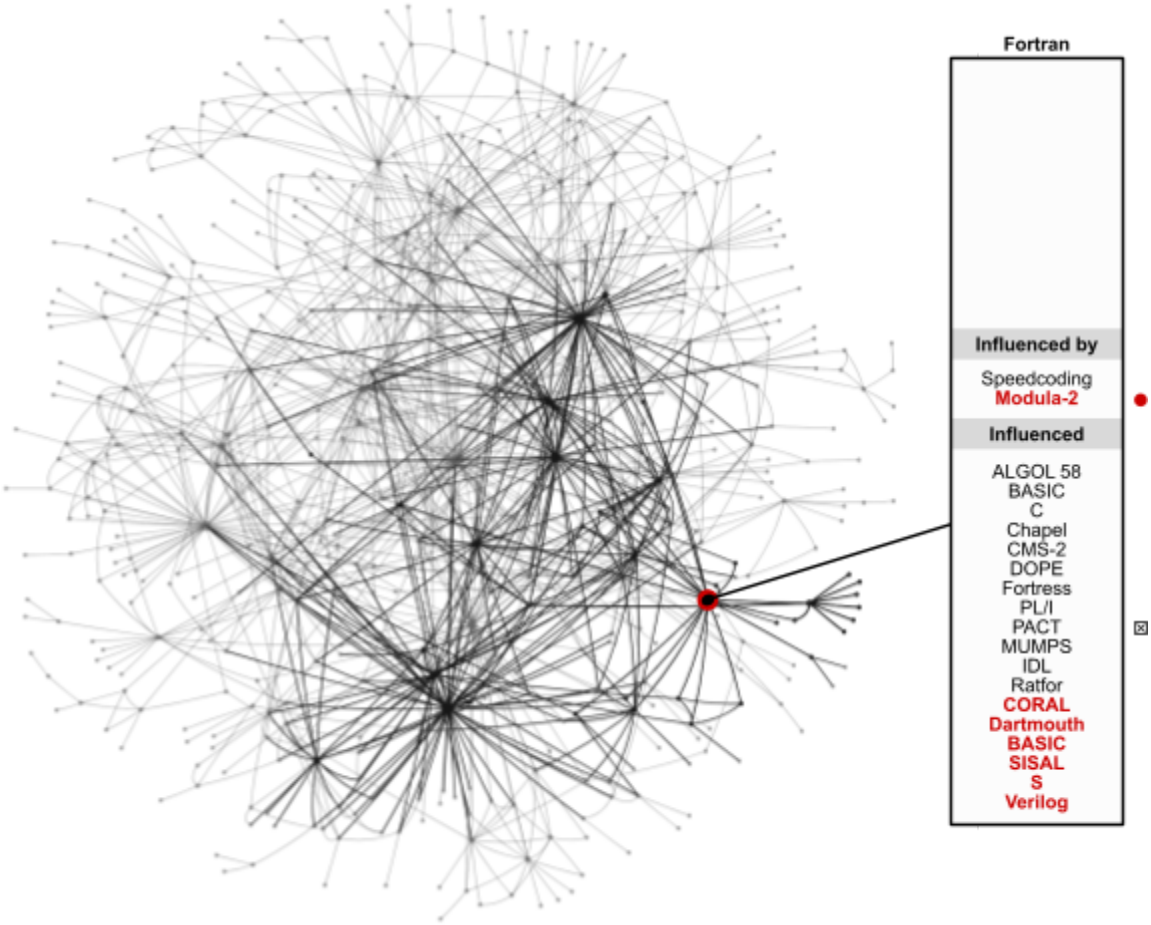

Figure 5. The Infobox Interlinker visualization system, using the Fortran infobox as a real world example. In this draft prototype all recursively scraped pages are shown as a node-link graph. The design philosophy of this visualization is underpinned by the goal, which is to remove as many red highlights as possible. Markers and highlights indicate errors to be fixed, the human editor is tasked with filling missing hyperlinks in infobox content so that semantic interlinking is achieved. In this example only one page remains, Fortran, highlighted by the red node in the node-link graph. A pop-up of the page's infobox with highlighted missing hyperlinks helps the editor with editing tasks.

To explain nuance, Fortran is a good example to showcase the need for a visualization to aid a human editor. Let's start with the "Influenced" parameter, where there are six page links missing and highlighted red. Each one of these connects to a page and the editor has no reason not to add them directly. This is the simplest case, so straightforward it could be automated. However, we don't actually want automation here, because there is nuance. For example, one of these suggested pages may not have an infobox, the visualization highlights this with a marker (crossed box). In this example, the page for PACT does not have an infobox. Any page that doesn't have an infobox behaves as a leaf in the "Influenced" directed graph, in this case PACT is an example, an example of the visualization revealing a missing infobox in the list of pages already listed in Fortran. In this example none of the discovered pages had an infobox missing, in the case any did the crossed box marker works the same way. Because no infobox means no content was scraped, the crossed box glyphs overrides any other marker glyph.

Nuance doesn't stop there, there can be more complex scenarios. Continuing with Fortran as an illustrative example, the vis system reveals that there is only one link missing in the "Influenced by" parameter, which is Modula-2. Turns out this link redirects to a subsection of the main Fortran page, Fortran#Fortran_90. When this kind of subsection redirect is present, a red dot marker highlights this for the editor. In this scenario the human editor has three choices. On one hand, the editor can do nothing, which means that Modula-2 will not be added to the Fortran infobox, and Modula-2 will keep redirecting to the Fortran 90 subsection. On the other hand, the editor can add the Modula-2 link, which establishes a two-way connection that mends the severance between pages, but is semantically misleading because Modula-2 (from 1978) influenced the newer Fortran 90 (from 1991) - not the original Fortran (from 1957). Finally, the third option is the most involved, which is creating a new page for Fortran 90, with its own infobox, and using the infobox interlinker to link them all technically and semantically.

MILESTONES

A total of 80 hours allocated to 14 weeks is 5.7 hours a week.

- 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
    - Continue work on scraper and analysis.
    - 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]
    - Devise visual encodings for new situations uncovered by data analysis.
    - Finalize the scraping and analysis logic
    - Prototype visualization system
- 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
    - Evaluate prototype with real-time Wikipedia infobox editing exercise.
- 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.

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.