

# Visualization Methods of Information Regarding Academic Publications, Research Topics, and Authors <sup>†</sup>

Hojung (Ashley) Kwon

Computational Media, Arts and Cultures, Department of Art, Art History, and Visual Studies, Duke University, Durham, NC 27708, USA; hojung.kwon@duke.edu

<sup>†</sup> Presented at the Conference on Theoretical and Foundational Problems in Information Studies, IS4SI Summit 2021, Online, 12–19 September 2021.

**Abstract:** Displaying search results in a vertical list, existing academic search engines do not reveal deeper insight into searched topics such as their connections with other topics. To address this issue, this paper proposes two interactive information visualization interfaces where users can discover networks of authors in computer science and interdisciplinary connections among research topics in philosophy, art, and history. The first interface presents collaboration relationships among authors and the authors' publications, using author name and publication title data extracted from more than two million papers published in the ACM journal until 2016. This interface, to our knowledge, is the first search engine that was built with a game engine, which sheds light on a novel use of a game engine as a search tool. The second method uses a web interface to show connections among seemingly unrelated concepts. Being the first digital implementation of Bill Seaman's poly-association, the interface possesses potentials as a widely available tool that can be used for discovering unexpected interdisciplinary relationships among research topics.

**Keywords:** information visualization; game engine; poly-association



**Citation:** Kwon, H. Visualization Methods of Information Regarding Academic Publications, Research Topics, and Authors. *Proceedings* **2022**, *81*, 154. <https://doi.org/10.3390/proceedings2022081154>

Academic Editor: Mark Burgin

Published: 5 July 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Existing academic search platforms, including Google Scholar [1] and JSTOR [2] allow users to search for publications using keywords, which include journals' names, authors' names, and research topics. However, showing papers relevant to the searched keywords in a list format, these search platforms do not provide further insights into the keywords such as other relevant keywords or influential authors from the field related to the keywords. Meanwhile, such insights are crucial in understanding the searched topics and research trends in disciplines where those topics are relevant.

In this paper, we introduce two digital interfaces that can provide deeper insight into publicly available information that we often find in existing academic search engines such as publication titles, authors' names, and research fields. In the materials and methods section, we describe the two interfaces' functionalities and underlying structures, as well as our motivations and inspirations behind designing those interfaces. In the results section, we describe data visualizations we produced with the interfaces. Finally, in the discussion section, we evaluate potentials and limitations of the two interfaces.

Although the research discussed in this paper was undertaken as part of a team (The Insight Engine 2.0 team), I am here speaking about much of my own research that has been happening in conversation with Bill Seaman and the other participating members.

## 2. Materials and Methods

### 2.1. Visualization of Author and Publication Relationships Using a Game Engine

Knowledge about a specific author's research interests and networks of authors who often publish papers together is helpful in understanding research topics and trends in a

field. However, most existing academic search engines often cannot intuitively visualize this information. Therefore, using a user interface designed with the Unity game engine [3] and programs in C# and Python, we developed a search platform that visualizes collaboration relationships among published authors and each author's number of publications.

### 2.1.1. Dataset

We used an open-source citation network dataset, which included title, publication year, and author names of more than 2 million papers that were published in the ACM journal until 2016. We modified the format of the dataset's entries so that each entry is a line in a txt file that contains comma-separated values of a publication title and the names of its authors. Using the txt file, we generated two JSON files, each of which had a set of unique keys and values associated with each key. One of the JSON files had the authors' names as keys and titles of all articles each author published as values. The other JSON file had as its key the authors' names. Each key had as its values names of other authors that the author indicated by the key collaborated with in at least one publication.

### 2.1.2. Unity User Interface for Visualizing Collaboration Relationships and Publications

To intuitively visualize academic collaboration relationships among authors in computer science and each author's number of publications in the ACM journal, we adapted the familiar form of a forest. We derive our inspiration from the 1992 book of Humberto R. Maturana and Francisco J. Varela, *The Tree of Knowledge: The Biological Roots of Human Understanding* [4], where the authors visualize in a diagram topics in disciplines including biology and behavioral sciences and show connections among those topics. As in Figure 1a, our Unity game engine interface includes a window where users can enter the name of an author. Once the user clicks the "Submit" button, the interface generates a forest similar to that in Figure 1b, where the searched author is represented as a tree at the center of the forest and all other authors that collaborated with the author are represented as trees distributed in a circular pattern surrounding the center tree.

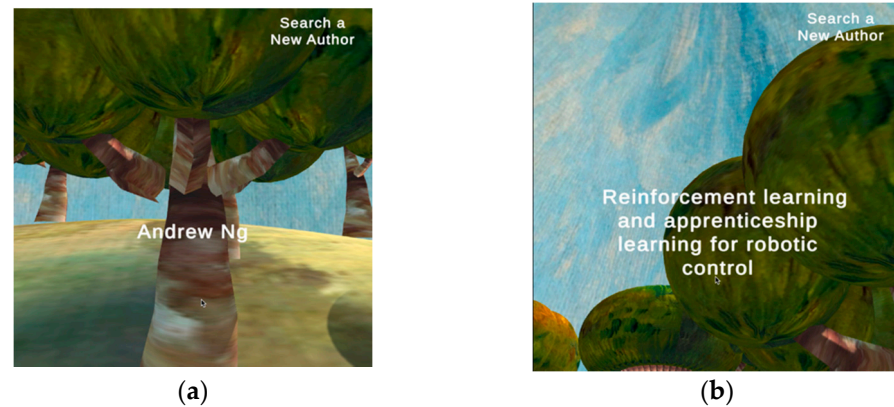


**Figure 1.** User interface of the forest model captured from the aerial view mode: (a) When the user launches the interface, it shows an input field where the user can enter an author's name and click the "Submit" button below the input field to send the name to the server; (b) The model visualizes authors as trees and their publications recorded in the dataset as leaf clusters.

The distance between a peripheral tree and the center tree is determined based on the number of papers an author represented by the peripheral tree published with the searched author. Let  $n$  be the number of joint publications between the author  $C$  that the center tree represents and the author  $P$  that the peripheral tree represents. Let  $m$  be the maximum number of joint publications  $C$  had with one author. We used the formula below to derive the distance  $r$  between the trees that represent  $C$  and  $P$ .

$$r = 40 \cdot (m + 2 - n) \quad (1)$$

The user can switch the visualization mode back and forth between the aerial view mode shown in Figure 1 and the exploration mode shown in Figure 2 using the 1 and 2 keys of a keyboard. In the exploration mode, the user can click the trunk of a tree to visualize the name of the author that the tree represents as in Figure 2a and the tree's leaf cluster to visualize the author's paper that the cluster represents as in Figure 2b.



**Figure 2.** User interface of the forest model captured from the explore mode where the user can use the arrow keys on a keyboard to explore the generated forest and a mouse or a trackpad to change the viewpoint: (a) By clicking the trunk of a tree, the user can visualize the name of the author represented by the tree; and (b) by clicking a leaf cluster of a tree, the user can visualize the title of the publication that the cluster represents.

### 2.1.3. Communication between the Unity User Interface and Backend Programs

We used FastAPI [5], a Python library that supports communications between a server and electronic devices, and Unity Web Request [6], a class in UnityEngine's Networking library, to build an interaction pipeline between the Unity user interface and a Python program. Once the user submits the name of an author, the Unity user interface's backend program in C# stores the name of the author in a JSON file and uses Unity Web Request's UploadHandler struct and SendWebRequest method to send the name to the Python program. Upon receiving the name, the Python program derives the names of all authors that collaborated with the searched author on at least one paper and all publications by the searched author and collaborating authors. Using FastAPI's put method, the program then stores this information in a JSON file and sends it back to the C# program, which visualizes the information in the Unity user interface.

## 2.2. Visualization of Poly-Association

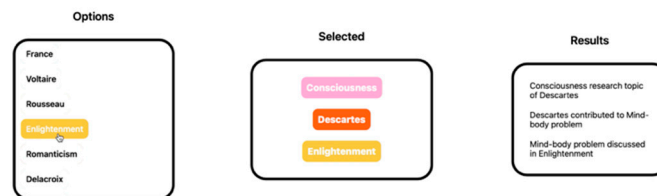
Arthur Koestler, in his 1964 book, *The Act of Creation* [7], claims that finding connections between two seemingly unrelated contexts can lead to new discoveries or inventions. Deriving inspirations from Koestler's concept of bisociation, Bill Seaman coined the term poly-association. Poly-association reveals connections between two or more contexts. Discovering instances of bisociations and poly-associations is a key to insight generation and creative problem solving. However, since these methods have been discussed in academic literature, there has not been any digital interface that aids the process of finding bisociation and poly-association. Therefore, we initially developed a model project. We created a webpage where users can visualize relationships among two or more topics related to 18th and 19th century French history, art, and philosophy. The notion is to abstract this system into the parameters and needs of the Insight Engine 2.0 project.

### 2.2.1. Poly-Association User Interface

The poly-association user interface uses three interactive windows to visualize connections between two or more concepts. When the user selects a concept in the "Options" window as in Figure 3, the concept appears in the "Selected" window in the middle of the

webpage. The user can view connections between two or more selected concepts in the “Results” window.

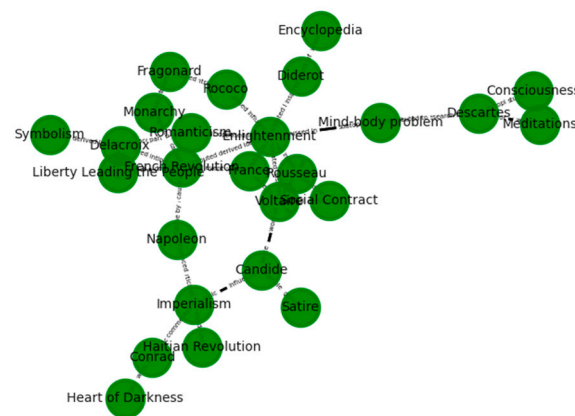
#### Poly-Association Ashley Kwon, Dev Seth



**Figure 3.** User interface of the poly-association interface: the user can select concepts in the “Options” window on the left-hand side of the webpage and view relationships among those concepts in the “Results” window on the right-hand side. All selected concepts are added to the “Selected” window in the middle. By clicking on a concept in the “Selected” window, the user can remove the concept from the list of selected concepts.

#### 2.2.2. Knowledge Graph Structure

The poly-association web interface has an underlying knowledge graph structure that its backend program uses to derive relationships among chosen concepts. As in Figure 4, the graph is composed of nodes, each of which represents a concept in art, history, or philosophy, and directed edges between two nodes, each of which describes how the two concepts represented by the nodes are related to each other. One node in the graph corresponds to one entry in the Options window in Figure 3.



**Figure 4.** Knowledge graph that the backend program of the poly-association interface uses to visualize topics in history, art, and philosophy, and relationships among those topics: In this graph, each node represents a concept and directed edges between two nodes, all of which have the weight of 1, describe how those two nodes are related to each other.

To generate entries into the “Results” section, the backend program of the webpage first finds every possible pair of nodes among selected nodes. Then, using Python’s networkx library for graph-related algorithms and visualizations, the program finds the shortest path between the two nodes in each of the pairs. All edges have the weight of 1. Therefore, the shortest path between the two nodes would be that involving the smallest amount of intermediate nodes and directed edges.

### 3. Results

With the game engine-based information visualization model, we were able to generate unique forest images for more than 1 million authors who published their work in the ACM journal. Meanwhile, the poly-association interface, containing 26 individual concepts

and possible number of selections ranging from 2 to 26, enabled us to discover numerous connections among these topics represented in the graph.

## 4. Discussion

### 4.1. Potentials of the Visualization Models

The game engine-based visualization model presents author collaboration and publication information using the form of a forest, which is familiar to most users. The model relies on simple metaphors such as an author as a tree and the author's publications as the tree's leaf clusters. As metaphors are not specific to a particular discipline, the model can be used to represent author networks and publications in various disciplines. The functionalities of the poly-association interface are intuitive and are also not associated with a specific discipline. These characteristics enable the interface to represent diverse interdisciplinary information.

### 4.2. Limitations of the Visualization Models

The database of the game engine-based visualization model currently only holds publication and author data from the ACM journal, which publishes papers in topics relevant to computer science and engineering. This database will be expanded to include papers in other disciplines such as those in humanities and enable the model to visualize interdisciplinary collaborations too. The poly-association interface also drew information from a relatively small graph network that we manually generated. We can address this issue by associating this interface with larger knowledge graph networks such as that created by Diffbot [8].

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The programs used to generate the tree models can be found through this link: <https://duke.box.com/s/vs67zw0ly9lrhmad74buh7xjx4lj55n>; The ACM publication data used to generate the tree models can be found through this link: <https://www.aminer.org/citation>.

**Conflicts of Interest:** The author declares no conflict of interest.

## References

1. Google Scholar. Available online: <https://scholar.google.com> (accessed on 28 November 2021).
2. JSTOR. Available online: <https://www.jstor.org> (accessed on 28 November 2021).
3. Unity. Available online: <https://unity.com> (accessed on 29 November 2021).
4. Maturana, H.; Varela, F. *The Tree of Knowledge: The Biological Roots of Human Understanding*; Shambhala: Boulder, CO, USA, 1992; pp. 11–14.
5. FastAPI. Available online: <https://fastapi.tiangolo.com> (accessed on 29 November 2021).
6. UnityWebRequest. Available online: <https://docs.unity3d.com/ScriptReference/Networking.UnityWebRequest.html> (accessed on 29 November 2021).
7. Koestler, A. *The Act of Creation*; Macmillan: New York, NY, USA, 1964.
8. Diffbot. Available online: <https://www.diffbot.com> (accessed on 30 November 2021).