In-dex

Varun Iyer · 2014 words

Abstract

This paper introduces in-dex, a tool that creates cross-reference links to important phrases within documents. The first section contextualizes the tool's function and purpose. The second section describes the tool itself. The third section discusses the implementation and technical aspects of in-dex.

1 Background

Most written works in English are meant to be read top to bottom. Authors must wrangle complex ideas into linear form, so that they can be understood in series by the reader.

Some works stand as exceptions to this rule — for example, branching narrative stories, or reference works that are meant to be read one entry at a time as necessary.

Digital works push the envelope by generating connections between different pieces of text in ways that the author did not explicitly specify. For example, Zoe Quinn's Depression Quest guides the user through a narrative that changes depending on the player's choices, and has many different possible ways to play through it, not all of which were explicitly laid out by the author.[1]

As another example, Wikipedia creates cross-references between different articles depending on the words used in a given article. This connects articles together into graphs that can be explored by a person, jumping from link to link, or by software that analyzes the relationships between articles.[2] These connections allow readers to read the text of Wikipedia in non-linear, unintended ways.

Code is also a unique kind of written artifact that is often read 'out of order.' Code is not read in order of writing, in the order it falls on the page, or in order of execution. Rather, it is read in a reverse-climax order, from broadest and boldest strokes down towards the details that a specific programmer is interested in.[3]

I was inspired to make in-dex by the way that indexing works in traditional non-fiction works and unintentional cross-referencing and network creation on digital platforms like Wikipedia. In-dex identifies important, key phrases in a traditional, linear text and then hyperlinks together successive uses of the phrase. This creates in-text cross references. By clicking a hyperlink, a user can jump to the previous or next use of a key phrase. The aim of these connections is to allow a user to explore a document in a non-linear way.

This non-linear exploration is a common strategy used to build an understanding of an area that cannot be read cover-to-cover, so to speak. For example, the U.S. legal corpus is far too large for single person to comprehend. If a person seeks to understand an area of the law, they have two choices. First, they could refer to a casebook, a representative set of documents curated and organized by an expert in the field.[4] Or, they could identify cases

1. Zoe Quinn, "Depression Quest," 2013, accessed June 3, 2022, depressionquest.com.

2. Luyu Wang et al., "WikiGraphs: a wikipedia - knowledge graph paired dataset."

3. Kevin Brock, *Rhetorical Code Studies: Discovering Arguments in and around Code* (University of Michigan Press, 2019), 140.

4. Library Innovation Lab, "Open Casebook," 2022, accessed June 4, 2022, opencasebook.org.

they think are important and read other works that the original case relies on heavily and future works that rely on the original case — constructing a casebook of sorts by using the information already available within the text. This process is similar to the 'relevant content' suggestions generated by social media platforms.[5]

I have been exploring academic and legal content suggestion recently through a project called Lexcaliber.[6] In-dex applies network exploration of text to navigation in a single document, rather than in a body of works.

A user who jumps from phrase to phrase explores a document by the ways that it touches on a particular topic, potentially revealing relationships or narratives that are more difficult to uncover when reading a document in its intended fashion. I have found in-dex especially helpful in exploring complex documents such as court opinions, as they touch on many different issues in different parts of the document. Cross-referencing works particularly well in allowing reading one issue at a time.

2 Description

The functionality of in-dex is currently as follows: the in-dex program is run with a simple HTML file (for example, the kind produced by Firefox reader) as an input. The program writes an indexed HTML file that contains links between key phrases in the original HTML file. This file can then be opened in any browser and the links can be explored.

The text of such a file is picture in Fig. 1, with in-dex's output pictured in Fig. 2. In-dex inserts small arrows linking important terms together — in this case, 'installation of pen register', 'search of petitioner's residence,' and 'Court of Appeals of Maryland,' among other terms. The up arrow carries the user to the previous use of the phrase, while the down arrow carries the user to the next use of the phrase.

The interface of in-dex is minimal, leaving the text of the document unobstructed and still dominating the uesr's view.

3 Tech

While having a relatively simple functionality, the working of in-dex is somewhat complicated, involving six distinct steps. I'll describe the steps and then explore the most important steps in greater detail.

1. Tokenize raw text into constituent sentences and words, and exclude stop words.

2. Count occurrences of phrases bundles of 3-4 sentences.

3. Identify phrases that occur in multiple bundles and multiple times in each bundle (suggesting a paragraph that deals with a given topic).

5. Chantat Eksombatchai et al., *Pixie: A System for Recommending 3+ Billion Items to 200+ Million Users in Real-Time*, 2017, https://doi.org/10.48550/ARXIV.1711.07601, https://arxiv.org/abs/1711.07601.

6. Faiz Surani and Varun Iyer, "Lexcaliber," 2022, accessed June 4, 2022, github.com/lexeme-dev/core.

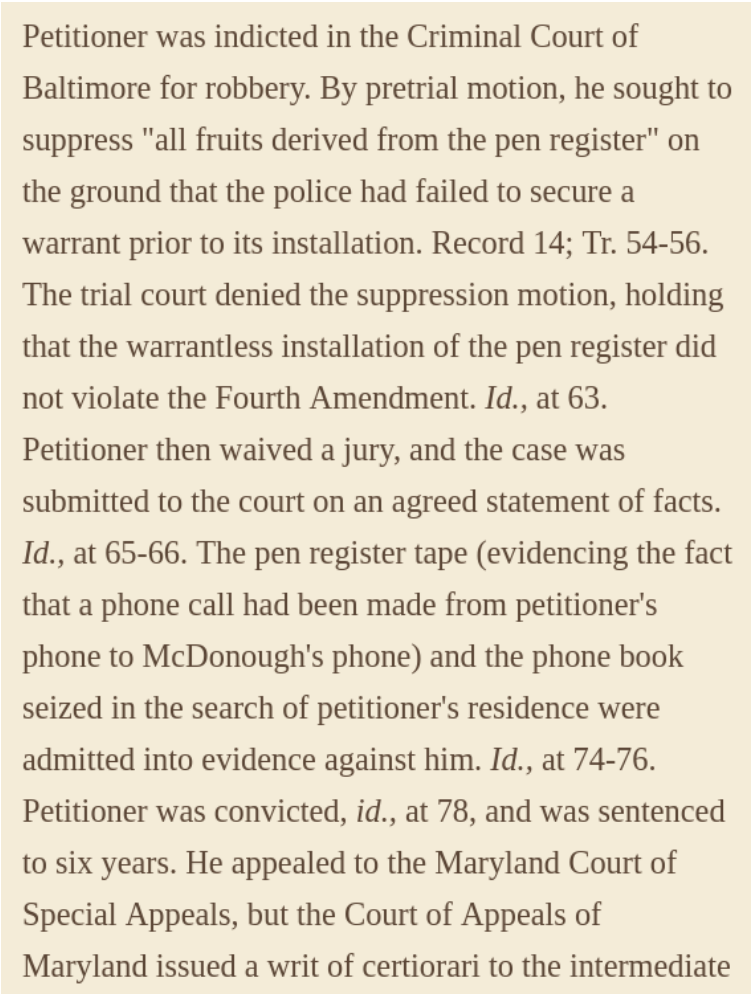
Petitioner was indicted in the Criminal Court of Baltimore for robbery. By pretrial motion, he sought to suppress "all fruits derived from the pen register" on the ground that the police had failed to secure a warrant prior to its installation. Record 14; Tr. 54-56. The trial court denied the suppression motion, holding that the warrantless installation of the pen register did not violate the Fourth Amendment. *Id.*, at 63. Petitioner then waived a jury, and the case was submitted to the court on an agreed statement of facts. *Id.*, at 65-66. The pen register tape (evidencing the fact that a phone call had been made from petitioner's phone to McDonough's phone) and the phone book seized in the search of petitioner's residence were admitted into evidence against him. *Id.*, at 74-76. Petitioner was convicted, *id.*, at 78, and was sentenced to six years. He appealed to the Maryland Court of Special Appeals, but the Court of Appeals of Maryland issued a writ of certiorari to the intermediate

Figure 1: Original text of *Smith v. Maryland*[7]

4. Fuzzy-find all occurrences of such key phrases in the document.
5. Insert HTML markup to identify these phrases.
6. Insert links that jump from one phrase occurrence to the next.

After identifying the words present in the document, in-dex must determine which concepts are relevant or important to the document. Finding a heuristic to approximate significance is a non-trivial problem, and required a few different attempts to identify a solution that seemed to work.

A first attempt relied on sent2vec, an unsupervised natural sentence vectorizer closely based on the famous word2vec. A vectorizer like sent2vec uses machine learning or traditional techniques to numerically represent non-quantitative data.[10] While this method had some success in identifying semantically similar sentences, it was not very effective at determining which of these relationships are significant or helpful.

10. epfml, "sent2vec," 2021, accessed June 4, 2022, github.com/epfml/sent2vec.

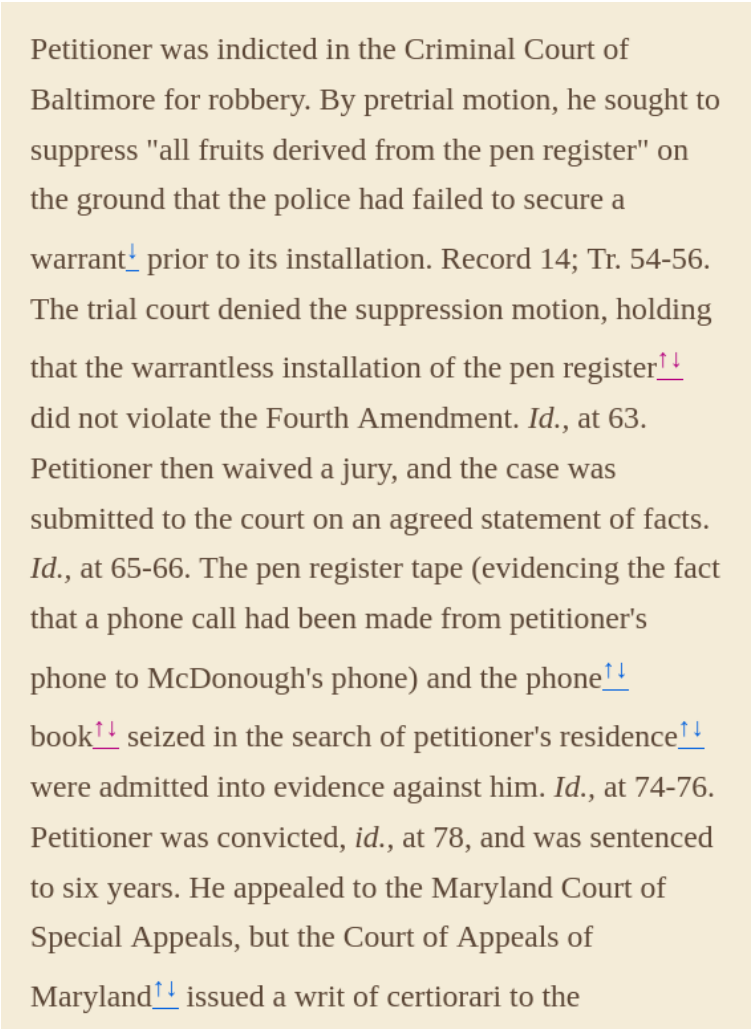
Petitioner was indicted in the Criminal Court of Baltimore for robbery. By pretrial motion, he sought to suppress "all fruits derived from the pen register" on the ground that the police had failed to secure a warrant prior to its installation. Record 14; Tr. 54-56. The trial court denied the suppression motion, holding that the warrantless installation of the pen register did not violate the Fourth Amendment. *Id.*, at 63. Petitioner then waived a jury, and the case was submitted to the court on an agreed statement of facts. *Id.*, at 65-66. The pen register tape (evidencing the fact that a phone call had been made from petitioner's phone to McDonough's phone) and the phone book seized in the search of petitioner's residence were admitted into evidence against him. *Id.*, at 74-76. Petitioner was convicted, *id.*, at 78, and was sentenced to six years. He appealed to the Maryland Court of Special Appeals, but the Court of Appeals of Maryland issued a writ of certiorari to the

Figure 2: *Smith v. Maryland*[9] after markup by in-dex; note the small arrow links after several terms.

Ultimately, the most effective method was a unique approach combining two different techniques, n-gram analysis and tf-idf. N-gram analysis is a method of text analysis that takes sequences of words rather than individual words as its fundamental unit of analysis. Looking at pairs or three words used adjacent to each other proved effective because identical, repeated phrasing is a good indicator that the same concept or topic is being discussed. Tf-idf is a technique used to identify relevant or interesting objects by analyzing the frequency of their occurence.[11] Terms that occur many times in one document (or paragraph, or set of sentences) but that do not occur in many documents are likely good descriptors of the unique content of that document. I applied the tf-idf counting method to series of three words, which proved to be effective at identifying recurring themes or topics unique to a few different parts of a text.

Figuring out how to represent the connections between different sections proved to

11. Wikipedia Contributors, "tf-idf," 2022, accessed June 4, 2022, wikipedia.org/wiki/Tf-idf.

be quite difficult; ultimately, I decided that displaying all possible connections was too visually cluttering and functionally confusing. Instead, the tool currently allows the user to jump to the previous use of a phrase (by clicking the 'up' arrow link) or the next use (by clicking the 'down' arrow link). Ultimately, in-dex still relies on the linear orientation of the original document, but allows different ways to move across it.

4 Conclusion

Digital technology and repositories of text open up new possibilities for how that text can be presented, read, shared, and explored. In-dex is a small proof-of-concept for a tool that allows users to navigate complex documents in non-linear ways. However, the true benefit of a tool like in-dex will likely be revealed more clearly when dealing with groups of documents rather than cross-referencing a single document. While in-dexing a single document is an interesting curiosity, in-dexing a set of complex documents may allow readers to identify connections in text that would have otherwise gone unnoticed. I hope to expand in-dex to fulfill this purpose in the near future.

References

Brock, Kevin. *Rhetorical Code Studies: Discovering Arguments in and around Code*. University of Michigan Press, 2019.

Contributors, Wikipedia. "tf-idf," 2022. Accessed June 4, 2022. wikipedia.org/wiki/Tf-idf.

Eksombatchai, Chantat, Pranav Jindal, Jerry Zitao Liu, Yuchen Liu, Rahul Sharma, Charles Sugnet, Mark Ulrich, and Jure Leskovec. *Pixie: A System for Recommending 3+ Billion Items to 200+ Million Users in Real-Time*, 2017. https://doi.org/10.48550/ARXIV.1711.07601. https://arxiv.org/abs/1711.07601.

epfml. "sent2vec," 2021. Accessed June 4, 2022. github.com/epfml/sent2vec.

Lab, Library Innovation. "Open Casebook," 2022. Accessed June 4, 2022. opencasebook.org.

Quinn, Zoe. "Depression Quest," 2013. Accessed June 3, 2022. depressionquest.com.

Surani, Faiz, and Varun Iyer. "Lexcaliber," 2022. Accessed June 4, 2022. github.com/lexeme-dev/core.

United States, The Supreme Court of the. "Smith v. Maryland," 1976.

Wang, Luyu, Yujia Li, Ozlem Aslan, and Oriol Vinyals. "WikiGraphs: a wikipedia - knowledge graph paired dataset."