Summarize the main points of the document: eyecite is a Python package for high-performance extraction of legal citations from text. It can recognize a wide variety of citations commonly appearing in American legal decisions, including:

- full case (e.g., Bush v. Gore, 531 U.S. 98, 99-100 (2000))
- short case (e.g., 531 U.S., at 99)
- statutory (e.g., Mass. Gen. Laws ch. 1, § 2)
- law journal (e.g., 1 Minn. L. Rev. 1)
- supra (e.g., Bush, supra, at 100)
- id (e.g., Id., at 101)

It also offers tools for pre-processing citation-laden text, aggregating like citations, and annotating citations with custom markup.

Statement of need

Citations are the bedrock of legal writing and a frequent topic of legal research, but few open-source tools exist for extracting them from legal texts. Because of this, researchers have historically relied on proprietary citation data provided by vendors like LexisNexis and Westlaw (e.g., Black & Spriggs, 2013; Fowler et al., 2007; Spriggs & Hansford, 2000) or have used their own personal scripts to parse such data from texts ad hoc (e.g., Clark & Lauderdale, 2012; Fowler & Jeon, 2008). While this is sometimes acceptable, human authors have used a wide variety of citation formats and shorthands over centuries of caselaw – and continue to add new ones – so accurate citation extraction requires maintenance of a long list of rules and exceptions. By providing an open-source, standardized alternative to individualized and closed-source approaches, eyecite promises to increase scholarly transparency and consistency. It also promises to give researchers the extendability and flexibility to develop new methods of citation analysis that are currently not possible under the prevailing approaches.

For example, one burgeoning research agenda seeks to apply machine learning techniques to citation analysis, either to recommend relevant authorities to legal practitioners (Ho et al., 2021), model the topography of the legal search space (Dadgostari et al., 2021; Leibon et al., 2018), or automatically detect and label the semantic purpose of citations in texts (Sadeghian et al., 2018). One obvious application of eyecite would be to use it to generate empirical training data for these kinds of machine learning tasks.
Functionality

To facilitate those kinds of projects and more, eyecite exposes significant entity metadata to the user. For case citations, eyecite parses and exposes information regarding a citation’s textual position, year, normalized reporter, normalized court, volume, page, pincite page, and accompanying parenthetical text, as well as eyecite’s best guess at the names of the plaintiff and defendant of the cited case. For statutory citations, eyecite parses and exposes information regarding a citation’s textual position, year, normalized reporter, chapter, section, publisher, and accompanying parenthetical text.

![Figure 1](image)

**Figure 1:** In step (1), eyecite consumes raw, cleaned text. In step (2), it parses the text into discrete tokens using Hyperscan and its regular expression database. In step (3), it extracts meaningful metadata from those tokens, returning a unified object for each parsed citation.

Because researchers often want to parse many documents and citations at once, eyecite is designed with performance in mind: it makes use of the Hyperscan library (*Wang et al.*, 2019) to tokenize and parse its input text in a highly efficient fashion. Hyperscan was originally designed to scan network traffic against large regular expression blacklists, and it allows eyecite to simultaneously apply thousands of tuned regular expressions to match the idiosyncratic ways that courts have cited each other over centuries of caselaw. eyecite’s regular expression database has been built from over 55 million existing citations culled from the collections of the Caselaw Access Project and CourtListener, the Cardiff Index to Legal Abbreviations, the Indigo Book tables, and the LexisNexis and Westlaw databases. From these sources, eyecite has honed a test suite of real-world citation strings. Figure 1 depicts eyecite’s extraction process of a full case citation at a high level.

eyecite offers other tools as well. Because researchers are often working with imperfect input text (perhaps obtained via optical character recognition), eyecite provides tools for pre-processing and cleaning it. Additionally, it can heuristically resolve short case, supra, and id citations to their appropriate full case antecedents, and it integrates well with custom resolution logic. Finally, for practical applications, it can also “annotate” found citations with custom markup (like HTML links) and re-insert that markup into the appropriate place in the original text. This works even if the original text was pre-processed, as eyecite uses the diff-match-patch library (*Google*, 2006) to intelligently reconcile differences between the original text and the cleaned text.

State of the field

To the best of our knowledge, no open-source software offering the same functionality as eyecite exists. Other similar packages are either no longer maintained or lack the robust

---

1 We estimate that eyecite can parse typical legal text on the order of approximately 10MB/second, though this depends on the density of citations within the text.

2 We estimate that eyecite can currently recognize 99.9977% of these citations.
parsing, resolution, or annotation features of eyecite (e.g., LexPredict, 2021; Sherred, 2021; Tauberer, 2017). eyecite also benefits from being used in production by two public data projects, the Caselaw Access Project and CourtListener, to process and analyze millions of documents in their collections. At least one study has already used an earlier version of the data generated by eyecite’s underlying code (Carmichael et al., 2017).

Limitations and future work

eyecite currently only recognizes American legal citations, as it was developed to extract data from cases published by courts within the United States. It is unclear how much of its design would apply to other bodies of law, though we hope that its conceptual abstractions would be extendable to other legal contexts as well. eyecite also does not offer worst-case performance guarantees, and both the citation extraction and annotation tools use libraries that may take exponentially long on worst-case inputs. It is therefore recommended to externally impose time limits if running eyecite on potentially malicious inputs. Finally, we have not explored other parser-based or machine-learning-based alternatives to eyecite’s collection-of-regular-expression-based approach to citation extraction. However, eyecite would be a strong baseline for performance and accuracy when developing such approaches.

References


LexPredict. (2021). LexNLP: Information retrieval and extraction for real, unstructured legal


Sadeghian, A., Sundaram, L., Zhe Wang, D., Hamilton, W. F., Branting, K., & Pfeifer, C.

(2018). Automatic semantic edge labeling over legal citation graphs. Artificial Intelligence
and Law, 26, 127–144. https://doi.org/10.1007/s10506-018-9217-1

raindrum/citeurl


doi.org/10.2307/449284

unitedstates/citation


A Fast Multi-pattern Regex Matcher for Modern CPUs. Proceedings of the 16th USENIX