2.3. The evolution of search engines and their application to Justice: opportunities and challenges. A technical perspective

Paolo Ferragina – Università degli Studi di Pisa & Scuola Superiore Sant'Anna

Summary: 1. Premise – 2. Algorithm versus AI definitions – 3. The history of Web Search Engines – 4. The role of AI in modern Search Engines – 5. Applications of search engine technology in Justice

Abstract: Inspired by Lessig, this contribution considers that, in order to understand the "age of cyberspace" we currently live in it is necessary to dig into the nature of all the impressive advancements that Computer Science has achieved in the last twenty years: mostly based upon search engines and AI. This chapter is divided into four main parts that deal with the definition of Algorithms and AI (often considered synonyms!), the history of web search engines (it is argued this "storytelling" will allow the reader to better understand the "opportunities and challenges" in their application to Justice), the recent role of AI in their design (which is a mix of ML/AI-based techniques for Natural Language Processing, Knowledge Graphs, and Generative AI), and conclude with a discussion on the value "that could be added" to legal search engines with all these algorithmic and AI technologies. Overall, this discussion will argue that more research and software development is still necessary in order to make the searching and mining of legal document collections easier, faster, more accurate, more "intelligent," and serendipitous in offering hints and views on legal arguments.

1. Premise

Writing about Search Engines and their long journey from Digital Libraries to Web giants like Google and Bing is much too long a story to be presented in only a few pages. Nonetheless, a glimpse of their main underlying technologies and evolution is warranted because this "storytelling" allows the interested reader to better understand the "opportunities and challenges" in their application to Justice, particularly in light of the bursting developments that have recently affected Artificial Intelligence (AI) and Machine Learning (ML), which are two new key ingredients of modern search technology.

Motivation for the content and structure of this chapter is taken from the following excerpt of the book "Code and other Laws of Cyberspace," written in 1999 by Prof. Lawrence Lessig,[40] who wrote: "Ours is the age of cyberspace. It, too, has a regulator. […] This regulator is code—the software and hardware that make cyberspace as it is. […] In a host of ways that one cannot begin to see unless one begins to understand the nature of this code, the code of cyberspace regulates." Inspired by Lessig, I also believe that, with the aim of understanding the "age of cyberspace" we currently live in, even more now than in 1999, we must to dig into the nature of "code" and begin to acquaint ourselves with two ubiquitous terms: Algorithms and AI.

2. Algorithm versus AI definitions

The Oxford English Dictionary states that an Algorithm is, informally, "a process, or set of rules, usually one expressed in algebraic notation, now used especially in computing, machine translation, and linguistics." The modern meaning for Algorithm is quite like that

[40] Roy L. Furman, Professor of Law at Harvard Law School and the former director of the Edmond J. Safra Center for Ethics at Harvard University.

of the terms "method," "procedure," "routine," except that the word Algorithm in Computer Science connotes something more precisely described. The recognized definition worldwide for the word Algorithm is due to Donald E. Knuth, Professor Emeritus at Stanford, who stated at the end of the 1960s that "an Algorithm is a finite, definite, effective procedure, with some output."[41] Although these five features may be intuitively clear, their significance is so dense that we need to look at some of them in more detail, as this investigation will lead us to better understanding the difference the terms algorithm and AI. We restrict our attention to:

- Definite: "each step of an algorithm must be precisely defined; the actions to be carried out must be rigorously and unambiguously specified for each case." This means that anyone reading the algorithm's description will interpret it in a precise way and nothing will be left to personal choice. This unambiguity is currently guaranteed by using one of many programming languages such as C/C++, Java, or Python.
- Input-Output: the behavior of the algorithm is not unique, but depends on the data given as input to be processed, which produces an output that constitutes the answer returned by the algorithm for those inputs. The mapping between inputs and outputs is precisely defined by the problem the algorithm must solve and must be "guaranteed" for all possible inputs. This is the so-called correctness of the algorithm.

On the other hand, there are many definitions of Artificial Intelligence that, according to some statements published by the European Parliament, either "refer to systems that display intelligent behaviour by analyzing their environment and taking actions – with some degree of autonomy – to achieve specific goals"[42] or "is the ability of a machine to display human-like capabilities such as reasoning, learning, planning and creativity."[43]

Among others, I prefer instead the definition given by David L. Parnas[44] in 2017, who described the AI approach as "heuristic programming." This definition is different from that given above for Algorithm because a heuristic program is one that "does not always get the right answer." Heuristic programs are based on rules that hang on experience, but are not supported by hand-written code or theory. Typically, "heuristic" is not a desirable attribute of software, but has been used effectively in recent years in more and more contexts, i.e. natural language understanding, audio and video processing, chat, text generation and translation, where finding a mathematically precise definition of the problem to be solved is difficult (if not impossible!). This approach gained popularity thanks to impressive advancements in the field of Machine Learning, which is another approach to the creation of Artificial Intelligence by constructing programs that "learn" from examples.

---

[41] D. E. KNUTH, *The Art of Computer Programming*, vol. 1-4, Addison-Wesley, 2023.

[42] COMMUNICATION FROM THE COMMISSION TO THE EUROPEAN PARLIAMENT, THE EUROPEAN COUNCIL, THE COUNCIL, THE EUROPEAN ECONOMIC AND SOCIAL COMMITTEE AND THE COMMITTEE OF THE REGIONS, *Coordinated Plan on Artificial Intelligence*, COM(2018) 795 final, December 7, 2018, available at: https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:52018DC0795&rid=3#:~:text=Artificial%20Intelligence%20refers%20to%20systems,autonomy%20%E2%80%94%20to%20achieve%20specific%20goals

[43] EUROPEAN PARLIAMENT, *What is Artificial Intelligence and how is it used?* in *News*, September 4, 2020, available at: https://www.europarl.europa.eu/news/en/headlines/society/20200827STO85804/what-is-artificial-intelligence-and-how-is-it-used

[44] D. LORGE PARNAS, *The Real Risks of Artificial Intelligence*, Communications of the ACM, October 2017, 60, 10, 27, doi: 10.1145/3132724.

This seems easier than designing an efficient algorithm that solves the actual problem and then coding for it. Nevertheless, these ML-based programs may be untrustworthy because they often exhibit weaknesses like incomplete or biased experiences, which are plugged into the system as "examples" and then turned into wrong answers by the learning process. In fact, ML solutions may fail when given unusual or untrained inputs, i.e. situations.

Let's now focus our attention on one of the most ubiquitous and sophisticated software tools currently available on every personal device, deploying the most advanced algorithms and AI/ML techniques ever designed by academia and industry: the Search Engine. Its simplicity of use has driven most users to consider it a trivial software to build.

Let's now briefly summarize the history of search engines over the last 30 years, and then "re-map" it onto the case of search tools for Justice. Sometimes a parallel example helps understand what can and cannot be done.

3. The history of Web Search Engines

The history of search engines, as we know now, is strongly linked to the history of the Web, which was born in 1991. The context at that time was very different from the present: there were only a few internauts and the Web consisted of only a few million well-maintained and reliable documents, i.e. pages, belonging to government or university sites. Those search engines, with names like Wanderer and Aliweb which are all now but forgotten, were based on extremely elementary algorithms for searching for user-specified keywords through meta-information that the authors of the pages manually associated to them. The search proceeded on that meta-information by using a linear scan, which was efficient only because of the limited number of existing Web pages.

The sudden growth in size of the Web made these approaches completely inefficient and new search engines were born, with perhaps more familiar names such as AltaVista, Lycos, Excite, and Yahoo!. These search engines introduced a set of criteria that could be used for sorting the results of a search for the first time, since they were growing more and more numerous given the growth of the Web. The concept of the relevance of results emerged which was addressed by means of two primary approaches: first, the Boolean retrieval model and, then, the more powerful Vector-space model. The former was primarily derived from the DataBase setting: a query consisted of a set of logical criteria for retrieving documents. The criteria specified the presence, and possibly the proximity, of indicated terms for documents to be responsive. The relevance measure ranked results in terms of how completely the Boolean criteria for the query were satisfied. However, terms were given "the same" weight, although this did not reflect their frequency or discriminative power, i.e. articles *versus* nouns. In the Vector-space approach to relevance, documents and queries were represented as vectors in a multi-dimensional space in which dimensions corresponded to terms and each vector component denoted the frequency of the corresponding term within the document and across the document collection (the so-called TF-IDF score). This representation surpassed the term-agnostic limitations of the Boolean retrieval model, and allowed for computing the similarity of documents/queries in purely mathematical terms by means of the scalar product of vectors. The final retrieval results achieved by Altavista and its competitors, at that time, were excellent and depended heavily on the fact that documents available on the Web were of high quality.

Around the year 1997, use of the Web in the business sphere and knowledge of how search engines worked paved the way for malicious practices aimed at influencing the ranking

of search results, a practice now known as "spamming." This heavily penalized the performance of search engines, making them often unusable for queries that contained frequent terms of interest to Web users. Countermeasures therefore became necessary as it soon became clear that Web page content alone was insufficient for determining their relevance to users' queries.

The subsequent biennium marked the beginning of the third generation of search engines and coincided with Google's birth, with its famous PageRank algorithm crucially based on the interconnections between Web pages, i.e. their hyperlinks. This generation of search engines, to which Ask Jeeves, Yahoo!, and Bing also belong, dominated the Web search scenario during the following decade. In the initial version of Google, the relevance of a page depended on its content, as in AltaVista, but also on the relevance of other pages pointing to it as well as which text, so called "anchor-text," surrounded those hyperlinks. This "centrality" measure was named PageRank, it was recursive in nature, and has proven to be one of the most important and persistent measures used for determining the relevance of a "node" in a network, whether it be on the Web, a social network, or a set of posts on Instagram, Facebook, or Twitter. Even now, whenever there is a network to analyze, a descendant algorithm of PageRank is typically one of the first options to consider. The third generation of search engines thus combined the textual information contained in Web pages and in anchor-texts with general information on the structure of the Web graph. This approach was so effective at answering user queries that second-generation search engines soon disappeared within a short time.

But, as is often the case in the world of Web search engines, the mechanisms for determining page relevance were quickly threatened by new spamming techniques, the most famous of them termed Google bombing.[45]

We are now living in the age of fourth generation search engines, in which there is world-class engagement between the two giant protagonists, Bing and Google, plus a multitude of others at the national level – such as Baidu in China and Yandex in Russia –providing specific contents (i.e. products, publications, users, maps), or claims of "semantic" searches, i.e. DuckDuckGo, and the most recent AI-based versions of Bing and Google, that interpret users' questions and carry out an in-depth analysis of document content. This latest generation is marked by an improvement in the efficiency and effectiveness of the search technology to "understand" the user query and document collection.

4. The role of AI in modern Search Engines

The above-mentioned capacity to "understand" hinges on ML/AI-based techniques for Natural Language Processing and Understanding (NLP/U), Knowledge Graphs (KG), and the latest advancements in Generative AI, such as ChaptGPT. The first techniques are used to process input texts, identify keywords or entities (possibly formed by multi-words) and then perhaps assign roles to those tokens and sentences, with their eventual "meaning." Here, "meaning" can refer to their Part of Speech – the so-called PoS, i.e. subject, verb – or, more interestingly, the corresponding concept in a Knowledge Base, such as Wikipedia or DBpedia. The former case fits into the classical realm of Computational Linguistics, which dates back to the 1960s, but with a revamped interest and more effective algorithms thanks

[45] https://en.wikipedia.org/wiki/Google_bombing

to the recent progress of AI/ML-tools. The latter was born in 2012 with Google's introduction of the first very-large Knowledge Graph, also known as a Semantic Network,[46] which represents a network of real-world entities – i.e. persons, events, objects, or concepts – and models the relationship between them thanks to links with (possibly many) associated types. There are now many known and freely available Knowledge Graphs. The key idea of their use in Search Engines is to disambiguate terms in indexed pages or query keywords by linking these terms to the proper corresponding nodes in the KG, i.e. Wikipedia pages. This represents not only a new way of mapping terms to concepts, but also a manner of empowering machines to extract interrelated concepts by percolating the KG starting from those nodes. By way of example, the query "Leonardo painted the Mona Lisa" clearly refers to the scientist and artist Leonardo da Vinci. So, the search engine, with the help of a KG, i.e. Wikipedia, connects "Leonardo" to the node representing "Leonardo da Vinci," i.e. https://en.wikipedia.org/wiki/Leonardo_da_Vinci. After which point, traversing the adjacent nodes in the KG, the search engine could discover the cities of Vinci and Florence, their nation Italy, or other information related to the famous scientist, such as the fact that he lived during the Renaissance. In some sense the KG expands the notion of ontologies to other kinds of entities and allows software engineers to develop highly sophisticated techniques for semantically annotating texts to support more intelligent and concept-based searches.

It goes without saying that the size and quality of the KG is crucial for the concreteness and completeness of these concept-based searches and reasoning, which requires very sophisticated AI techniques and algorithms to process and digest large volumes of (often unstructured) texts from which that Knowledge is extracted and interconnected to form these Graphs.[47] This approach gives algorithms the power to reason about the significance of terms and texts, along with finding similarities that go much beyond the (syntactic) sharing of terms.

More recently, this "understanding capacity" has been further extended with the advent of Transformers[48] and other sophisticated ML tools that, by processing billions of texts, extract mathematical representations of keywords capturing, in a sense, their "semantics" from their co-occurrence with other words in those large textual collections. The most notable ML-techniques in this setting are the original GPT (Generative Pre-trained Transformer)[49] and BERT (Bidirectional Encoder Representations from Transformers).[50] The former has been mainly used to generate human-like text beginning from questions or phrases, so-called prompts; the latter has been mainly used to build tools for some important end-to-end applications, such as entity recognition and document classification, thanks to novel and effective vector-based representations of token/words or sentences, which bring with themselves useful context-based (semantic?) information. Note that these approaches are orthogonal to previous ones and thus can be – and indeed have been – used to mine KGs

[46] https://en.wikipedia.org/wiki/Knowledge_graph

[47] See X. L. DONG, E. GABRILOVICH, G. HEITZ ET AL., *Knowledge Vault: A Web-Scale Approach to Probabilistic Knowledge Fusion* in *Proceedings of* the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 2014, New York, USA, 601.

[48] https://en.wikipedia.org/wiki/Transformer_(machine_learning_model)

[49] T. WOLF, L. DEBUT ET AL., *Transformers: State-of-the-Art Natural Language Processing*, in *Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020, 38, doi:10.18653/v1/2020.emnlp-demos.6.

[50] *Open Sourcing BERT: State-of-the-Art Pre-training for Natural Language Processing*, in *Google AI Blog*, November 2, 2018.

or reason more efficiently than them (though it is too technical to discuss the algorithmic details here).

5. Applications of search engine technology in Justice

We are now ready to dig into the "value-added" or "value that could be added" to legal search engines with respect to Web search engines, in light of the technological achievements just outlined above.[51] The two most known and notable commercial search systems for legal documents in the United States are Westlaw Next (WN) and Lexis Advance (LA), which offer many of the features commented on below and that should be compared with the features offered by, for example, the Italian *Italgiure* system, currently available for the personnel of Italian Courts.[52] All those systems, in one way or the other, have followed the evolution of Web search engines, though they have not yet achieved the same level of sophistication and efficacy, for many reasons that are also intrinsic to the nature of legal documents and user needs.

The first issue to deal with here is the composition of a user query. In the legal context, more than in the classic Web search, syntactic searches are not enough to match the needs of (legal) users. Concept-based retrieval is essential, and is becoming more and more mandatory as the size of digitalized legal document collections has increased in the last few years. Additionally, the use of techniques such as query auto-completion or query expansion could turn out to be very effective in empowering legal users to design queries that are better composed. In the former, a "dictionary" of potential queries is fundamental upon which a user's query is matched for its completion. Web search engines use dictionaries drawn from many sources, the most important of which are query-logs, built by the search engine during its use. In the latter, legal ontologies or, better, legally-centered Knowledge Graphs should be built to properly interpret and then expand the "syntactic" queries posed by users with additional meaningful terms that solve polysemic or synonym issues present in them.

In this context, a crucial role could also be played by so-called user relevance feedback, which would have a key role in "personalizing" the ranked list of search results via expert-generated annotations. The key algorithmic idea would be to flag some relevant results, in a sort of human-in-the-loop feedback system, that helps the machine learn a model embodying those judgments in a way applicable to new documents. Re-ranking might employ evidence derived from those expert-generated annotations, frequency information in the text of documents, citation networks and document popularity from previous queries. The (re-)ranking function could also be optimized by using ML to determine the weights to ascribe to those different features.

It is evident that the interpretation of user queries is strictly tied with the analysis of legal sentences. This action may occur at different levels of granularity: from classic and simple Part-of-Speech tagging to Entity linking discussed above, up to ultimately adding more semantic information about the role of the sentences in legal arguments. These "annotations" could be exploited by the search engine to rank the results depending on the "role" played by the searched terms into the indexed sentences, but also to reason about the concepts

[51] See K. ASHLEY, *Artificial Intelligence and Legal Analytics: New Tools for Law Practice in the Digital Age*, Cambridge, 2017.

[52] https://www.italgiure.giustizia.it/

involved in those results or to visualize them better by deploying representation schemes that surpass the usual linear list.

In fact, the third key issue to be mentioned is related to the visualization of query results. Web search engines have habituated us to the linear list of so-called "ten blue links," with their associated snippets of a few lines and searched keywords in bold. This choice leaves it up to the users to read the list of resulting documents and decide which of them is relevant and refine the query based on what was found or not. Although poor from an effectiveness perspective, the linear list has become the de facto visualization standard, mainly dictated by the small screens of our mobile devices, the simple needs of Web users, i.e. navigational, transactional, and informational, and by their short patience (a few seconds…).

In the legal context, the scenario is much different: users may spend time looking carefully through the search results as their needs are more conceptual, they are also possibly open to serendipitous suggestions, and legal-document collections are not as big as the Web. These characteristics should lead search-engine designers to afford for a more sophisticated processing of documents and queries while, on the other hand, they should be advised that legal language is more "complicated," so that polysemy/synonym issues, as well as the role of words/sentences in those documents, are taken into account. These issues were already discussed in the previous section, with respect to their impact on the (re-)ranking of search results. Browsing a plain (linear) list of hundreds or thousands of search results (possibly sorted by date or other simplistic criteria) isn't humanly feasible and can't be performed with the right level of attention even for legal users. As such, it should be declined. In this respect, the legally-centered Knowledge Graphs mentioned above as well as the citation networks, based on articles, entities, or concepts mentioned in the resulting documents, could be deployed to arrange search results in the form of *graphs* that could be far more visually effective than the plain list and allow users to extract a fast and meaningful glimpse from them. Moreover, graphs are browsable in several conceptual dimensions, thus resulting in a greater flexibility than linear lists to adapt to the search needs of users and to accompany them more efficiently and effectively in retrieving what they were searching for or to discover "new" concepts or arguments that are useful for their work.

The fourth and last issue is dedicated to the impressive progress of generative AI, and ML in general, that is often succinctly summarized these days by the tool "ChatGPT." This family of tools is referred to this way because Search and Generative AI will progressively converge. Generative AI went mainstream in 2022 with ChatGPT and Dall-E, offered by OpenAI, after the seminal work done by Google with Transformers. However, Generative AI still needs a lot of effort to check the factuality and groundedness of its generated phrases, by preventing "hallucinations" (I prefer "rambling") in texts that look correct, but which are, in fact, not. Web search, on the other hand, could help with fact checking algorithms and with Web references provided together with AI-generated text.

But there is another way to "effectively merge" these two approaches, as some tools/apps are already pursuing in various contexts: namely, to use ChatGPT for reasoning about a large part of search results. That is, deploying the power of ChatGPT or similar tools to produce humanly-readable summaries of "significant parts" of search results, to answer specific questions regarding them, to extract argument-related information, or finally to order the results in a manner that is tailored to the problem a user seeks to address. This would avoid the fatigue of reading hundreds of results snippets and generate outputs that are currently not possible with current textual searches. In this scenario, it can be argued that search engines for legal documents could benefit the most by the mixing of Generative AI, classic

AI/ML, and algorithms, thus making the most advanced search assistant one could think of available to their users.

Let us conclude this chapter by mentioning that, in the last year, we have investigated some of these challenges and issues within the context of Italian legal documents, thanks to the support of a PNRR-PON project, named "*Giustizia Agile,*"[53] that has seen fruitful collaboration between the Court of Pisa and several Departments of the University of Pisa. Its goal was the study, design, implementation, and experimentation with a software platform for the analysis, indexing, search, and visualization of Italian legal documents, by following some of the methods suggested above. During this project, three main gold standards were constructed semi-automatically – for NER, document classification, and keyword extraction – that could be adopted for further studies with Italian legal documents, and a preliminary experiment on keyword extraction via Generative AI was proposed. Overall, the study demonstrated that a lot of research and software development is still necessary in order to bring Italian legal search engines up to the task of matching the (not so futuristic) vision discussed in the paragraphs above, in which the most advanced AI and algorithms blend together to make the searching and mining of legal document collections easier, faster, more accurate, more intelligent, and possibly serendipitous in offering hints and views to legal arguments for the daily work of court officials, judges, and lawyers.

In the end, the opportunities for applying "intelligent" search engines as well as interesting and powerful AI/ML and algorithmic techniques to the justice sector already abound these days both in industry and academia. Advancements in legal search engines are sure to surprise us in the not-so distant future.

[53] A special thanks to all colleagues and students who collaborated on implementing the software platform we designed for the PNRR-PON project, "Giustizia Agile:" Leonardo Calàmita, Piero Cossu, Matteo De Francesco, Chiara De Nigris, Alessandro Lenci, Giacomo Mariani, Giovanna Marotta, Lucia Pàssaro, Erika Pistolesi, Mattia Proietti, and Giacomo Vaiani. A warm thanks also goes to the colleagues (especially, PIs Benedetta Galgani and Giuseppe Campanelli), researchers, and fellows of the Departments of Law and Management Engineering, as well as to the officials and judges of the Courts of Pisa, Lucca, and Livorno, who contributed to shaping and driving the success of this project. My final warmest thanks and gratitude goes to Maria Giuliana Civinini, who was the real "[search] engine" of collaboration that accompanied the intense study and software design behind this project.