INSIDE THE BLACK BOX OF SEARCH ALGORITHMS

BY SUSAN NEVELOW MART, JOE BREDA, ED WALTERS, TITO SIERRA & DR. KHALID AL-KOFAHI

A behind-the-scenes look at the algorithms that rank results in Bloomberg Law, Fastcase, Lexis Advance, and Westlaw.
Algorithmic accountability means different things in different contexts. In legal search, accountability means sufficient transparency to establish trust between the legal database providers and the researcher, and it means providing sufficient help in understanding how algorithms affect search results to ensure that legal researchers can be effective in their search strategies. Satisficing is a compelling strategy for online legal researchers used to determinate responses to their general queries in Google or other search engines. Algorithmic accountability is one way of enabling researchers to move beyond a single search and its top results. During the 2018 AALL Annual Meeting program “Inside the Black Box,” Fastcase and Lexis Advance lifted the hood on their algorithms to help legal researchers better understand their unique algorithms and why results vary so much for the same search in the same jurisdictional database across platforms. In this article, Bloomberg Law and Westlaw join Fastcase and Lexis Advance to continue that conversation.

An Overview of Basic Search Relevance Factors
Although there are many ways search is implemented in legal databases, there are some common ingredients in how search relevance is calculated. Knowing about these ingredients can be helpful in understanding why searches work the way they do.

TF-IDF!
Most full-text search engines, including legal search engines, use query-document term similarity as a foundational element in relevance ranking. The most commonly used approach for calculating query-document term similarity is TF-IDF (term frequency/inverse document frequency). Term frequency is a count of the presence of the user’s search...
terms in matching search documents, with the assumption that documents that include the user’s queries more frequently are more relevant to the query. To avoid having commonly occurring terms in the documents dominate ranking, the scoring contribution for specific terms is typically discounted by the overall frequency of the word across all searchable documents (inverse document frequency).

**A Rose by Any Other Name**
Anyone who has constructed a complicated Boolean search has probably employed something that looks like this: “… AND (suit OR claim OR action OR petition)…” Modern search engines will expand the user’s query to include potential synonyms for the terms provided by the user. A more sophisticated variation of this approach will expand search terms or provide suggested terms using algorithms developed over the corpus that identify relationships between words in the corpus.

**Terms of a Feather Flock Together**
Another potential input that affects search results ranking is how close the query terms appear in a document (“proximity”). Consider a search for “financial disclosure”—one would expect a document containing these terms in that exact order would be more relevant than a document where “financial” and “disclosure” appear in entirely different sections. Proximity of terms is particularly important for legal phrases that appear frequently in legal texts.

**Rank and File**
Legal search engines frequently provide additional inputs that determine the ranking of results. When ranking court opinions, for example, a decision’s precedential value can impact its relevance. Matching text found in the majority portion of a recent U.S. Supreme Court decision will likely (but, admittedly, not always) be more relevant than a dated circuit court dissent. The frequency with which a given document is cited by other documents is also valuable in ranking.

**Algorithmic Insights, Implementation Guidance, and Usage Tips from Search Database Providers**

**Bloomberg Law**
The goal of any search system is to get users to the answer they seek as fast as possible. Bloomberg Law employs standard techniques to optimize the relevancy of natural language results. Our search system parses user queries to identify legal phrases and entities and then prioritizes documents containing these items. In measuring the relevance of matching documents, we also consider the proximity and distribution of matching keywords. To ensure that the most relevant documents are appearing at the top of our search results, subject-matter experts periodically evaluate the results of queries and provide explicit feedback that is used to fine-tune our scoring algorithms.
Bloomberg Law is also implementing context-specific search experiences that help the user more easily identify content specific to their practice area or topic of research. For example, a user using Bloomberg Tax would receive search results customized to that practice area, based on the search system’s understanding of the terminology in that area of law. Our empirical research shows that attributes of relevancy vary by context, so by offering a specialized search experience based on the user’s context, Bloomberg Law can offer better, more relevant results.

It is important to remember that legal information systems offer a great deal more than a searchable content database. We also use global search fields to help users discover functionality and navigate the platform. When a user enters a query in the “GO Bar” found at the top of the Bloomberg Law website, the search system examines the entered terms and generates a list of suggestions, based on degree of match and popularity, to help users discover useful platform features.

Fastcase

Many services are trying to make legal research more “Google-like.” Fastcase is trying to empower deeper research, and its tools are designed to be less Google-like. Fastcase’s search engine uses 16 different factors to rank search results, including TF-IDF keyword relevance, proximity, authoritativeness (citation counts), recency, and the aggregate history of more than 100 million searches on the system.

Fastcase’s ranking algorithm automatically gives a different weight to each of these factors to bring the most authoritative results to the top of the list. One problem with all algorithms is that they are optimized for the best overall results, but no search engine can be optimized for every kind of search. For example, the researcher may want to look at a subset of documents that are rarely cited or may want to privilege documents from the 1980s and 1990s.

Fastcase exposes the levers and allows power-users in Advanced Search to customize their own relevance algorithm for the particular research task. There are a few other unique features for power users inside the “black box” at Fastcase. Researchers frequently do not read past the first page of results. Fastcase’s search results are “infinite scroll,” so that researchers can read seamlessly without paging through the application.

Fastcase’s citation analysis algorithms allow researchers to re-sort search results to bring the results most cited by other documents in the search results to the top of the list, and it can even find cases that should be in the results but aren’t. The algorithms identify cases frequently cited by the search results that don’t contain some search terms but are nevertheless important to the research query.

Lexis Advance

To deliver high-quality search results for all users, search tasks, and search methods, the Lexis Advance search engine
utilizes a suite of algorithms to identify the user’s search intent, select the most relevant matching documents for this intent, and apply the best possible relevance ranking to these results.

Identifying user search intent is a critical first step in delivering high-quality search results. This step involves parsing the user’s query to identify search intent. For users that prefer more control, Lexis Advance supports an extensive set of search commands, including Boolean operators, proximity operators, wildcards, and quoted phrases. For queries without search commands, Lexis Advance applies natural language processing to identify user search intent. This includes identifying specific legal citations, legal topics, jurisdiction names, semantic concepts, named entities, and other signifiers of intent. Term and phrase equivalents are also added to increase recall of relevant documents.

The next step is returning search results aligned with the user’s intent. For Boolean searches this is largely a mechanical process of honoring the search, matching criteria specified by the user. For natural language searches, an algorithm is applied to determine what terms in the user’s query are required in matching documents.

These results are then ranked for relevance. Relevance ranking on Lexis Advance uses a combination of factors, including term frequency in document, term proximity in document, and many content type-specific relevance weighting factors. For example, citation activity is used for case law content search, and publication date weighting is used for news content search. LexisNexis optimizes relevance ranking for each content type.

LexisNexis continually invests in its search technology, increasingly utilizing natural language processing and machine learning algorithms to deliver relevant search results. Prior to releasing an algorithm update, LexisNexis employs a comprehensive testing process to ensure that search changes are both impactful and a measurable improvement to using industry-standard search quality measures. The significant investment in search testing and evaluation ensures that the Lexis Advance search quality gets better and better with every release.

**Westlaw**

Westlaw relies on a set of vertical search engines. Each one is tuned to one or more content types. For example, citations, key numbers, and treatment history play a larger role in the search engine for case law documents than those for statutes and regulations, while
document structure plays a larger role for regulations and statutes engines.

Unlike horizontal search engines, which mostly rely on query-document term similarity, vertical search approaches allow us to codify how we think about a problem and represent it in a way suitable for algorithmic computations. For example, a decision is often not only meant to resolve the dispute at hand but can also be used to resolve similar disputes in the future. The ramifications of a decision are not fully contained in the decision’s words and phrases; rather, the decision can only be fully understood after the fact, as other courts apply it and interpret it. A search engine, therefore, must capture this ‘meaning’ that did not happen yet, or for older decisions, a meaning that is distributed across many decisions (i.e., the citation network of a case).

To solve this problem, Westlaw uses machine learning algorithms called learning to rank, which combines a diverse set of factors in its ranking function and represents the challenge as an optimization problem. In the case of Westlaw, it minimizes the number of pairwise inversions, where a lower-quality result is ranked before a higher-quality result. To ensure this is not a popularity contest, where highly cited decisions dominate the results, Westlaw algorithms use normalized features to level the playing fields. In addition, we use a number of stratified sampling strategies to ensure the long tail of the law is represented in both the training and the evaluation of the algorithms.

Recently, Westlaw Edge extended those capabilities through a set of proprietary natural language processing algorithms that aim to “understand the meaning of a query” and, when appropriate, provide answer-like results.

**The Researchers’ Challenges with Any Algorithm**

Boolean (“terms and connectors”) searching is satisfying in its simplicity. Although there are minor variations in the way Boolean searches are processed in each database, the results of a Boolean search may be cleanly divided into two buckets: those with a precise literal match to the search query are returned as results, and those without a precise literal match are not.

The operation of such a search is easy to understand, and results can be verified with certainty. Another benefit is that the intent of the search is known to the researcher. A Boolean algorithm is simple: The primary search syntaxes at work are proximity and field limiters. Skill is needed: A Boolean search presupposes the researcher is able to effectively translate a concept into an often complex search query that captures all of the concept’s possible expressions within an increasingly large corpus of data.

Semantic or “algorithmic” search approaches search results differently. Rather than deterministically dividing results into matches and non-matches, modern search engines take a probabilistic approach. Effectively, the entire corpus is ranked from “most likely to be responsive” to “least likely to be responsive.” The benefit is that multiple search syntaxes and formulas are applied to the query. But the algorithm is probabilistically guessing at intent. Therefore, skill is needed: The researcher still needs to formulate a sufficiently targeted search to help the algorithms do their work.

Serious thinking about the legal problem is the best first step for every search. Both types of searches set algorithms to work to bridge the gap between the researcher’s query and the documents in the system. The human researcher still plays a vital role in priming the algorithm that closes that gap.

**READ**

Susan Nevelow Mart’s article “Every Algorithm Has a Point,” from the September/October 2017 issue of *AALL Spectrum* at bit.ly/SO17Algorithm.