Machine-Assisted Issue Spotting for Self-Represented Litigant Portals

An Introduction to Spot—a multi-label text classifier[1]

David Colarusso
School of Law
Suffolk University
Boston, MA
dcolarusso@suffolk.edu

Motivation. If you know the name of what you are looking for, a web search or well-curated site menu are likely to get you what you need. Would-be litigants, however, are often unaware of the labels attached to situations by the legal system. Consequently, they may not recognize the problems they face as legal problems or fail to have access to the term of art needed to power a productive search. No one says, "I am suffering from a *constructive eviction*," rather they know their landlord has made it impossible for them to live in their home. Spot is a tool that attempts to meet people where they are, translating non-lawyer expressions of fact patterns into collections of potential issues from the Legal Issues Taxonomy. At least five state-wide portals/resource finders and one national online forum currently make use of Spot to help users discover relevant resources.[2]

In This Paper

What is Spot?

Spot is an online issue spotting service. Give Spot a non-lawyer's description of a situation, and it returns a list of likely issues from the Legal Issues Taxonomy (LIST)—formerly the National Subject Matter Index, Version 2. LIST provides the legal aid community with a standard nomenclature for talking about client needs. It includes issues like eviction, foreclosure, bankruptcy, and child support. Spot is provided free of charge to the legal aid community over an online API (Application Programming Interface). Mostly, this means it is built for use by computer programs, not people. Coders, however, can build things—like websites—on top of the API. By automating part of issue identification, developers can use Spot to help people in need of legal assistance better access available resources by identifying potential issues in the text provided to prompts such as, "What seems to be the problem?" These identified issues can then be used to surface resources to the end user. Spot works best on short fact patterns composed of a few sentences and a couple hundred words.

[1] You can learn more about Spot at the project's website: https://spot.suffolklitlab.org/. There you will find links to training data, measures of performance, and even a sample implementation you can take for a test drive.
[2] See e.g., MA Legal Resource Finder, CT Law Help, Illinois Legal Aid Online, Court Forms Online, Pine Tree Legal Assistance, and the Reddit Eviction help bot.

For example, provided with “The heat and water in my apartment aren't working,” Spot will return a number of possible LIST issues along with statements about how sure it is they are present. For the above text, it thinks the most likely issue is LIST issue HO-05-00-00-00 (Problems with living conditions), putting the odds of that issue being present at 91%, with lower and upper bounds of 0.62 and 0.93 respectively.

It is worth noting that the LIST issues Spot returns are not *legal issues* in the sense they may not necessarily have legal remedies. Rather, they are *issues* in the colloquial sense, more akin to what an attorney would call a fact pattern. For example, a particular set of facts, as described in a text, may return a *housing issue* independent from the availability of remedies. That is, it can tell you a text is likely talking about “living conditions” but not whether the text implicates a particular remedy as the answer to this question could differ from jurisdiction to jurisdiction.

Additionally, Spot is what is commonly called a multi-label classifier, not a multiclass classifier. Think checkboxes, not radio buttons. With multi-label one assumes that multiple correct answers may exist, whereas multiclass assumes only one correct answer. This means that Spot can pick up on a constellation of co-occurring issues, a must when dealing with short narrative descriptions.

How good is Spot?

That depends on what you're using it for. Think of Spot like a map. It doesn't have resolution down to the inch, but it can still be helpful. Like a map, Spot is a model, and as the George Box quote says, "All models are *wrong*, but some are *useful*." Because it's "wrong," those using Spot should recognize that its output is not the final word, and if they want to know if it's useful, they'll have to ask "compared to what?" That being said, what you probably want to know is, "how wrong is it?"

Unfortunately, you can't just look at Spot's accuracy. To understand why, let's say you want to evaluate an algorithm that predicts if there will be a snow day tomorrow. I have an algorithm with an accuracy of 98%. Impressive right? What if I told you my algorithm was, "always guess no?" My model is 98% accurate because snow days only happen 2% of the time. To know if my algorithm was any good, you need to know more. So you might ask what percentage of actual snow days did I "catch?" This is something called recall. The answer is 0%. You could also ask how often am I right when I say something is a snow day. This is something called precision. Since I didn't actually predict any snow days, I can't even calculate this number because I'd have to divide by zero. Either way, these alternative metrics make it clear that my model is no good. And none of that even takes into account that I may have a preference for false positives over false negatives or vice versa. Consider a screening test for some ailment. The idea is that it's one step in a process, a filter used to identify folks for a diagnostic test (i.e., it's not the final word). In such a case you might care more about minimizing false negatives than you do about false positives. The point is, it's complicated.

That being said, you can see detailed performance metrics for each of our active labels at https://spot.suffolklitlab.org/. These metrics measure the performance of Spot’s classifications against a number of labeled datasets, described below under the heading “Who trains Spot?” The *active labels* qualifier limits our model to only those labels for which we can make predictions better than a coin flip or always guessing yes/no.

Spot currently recognizes 105 active labels.The average weighted[3] accuracy for our active labels is currently 94.0%. For recall, it's 77.0%, and precision is 75.0%.

Those performance metrics, however, are the ones you get if you take the model as predicting the presence of a label when it states there is more than a 50% chance of it being there. If you change the API's cutoff, you can favor recall over precision, or the other way around. So API users should think carefully about their use case and what cutoff is appropriate. Context should inform where one places these cutoffs. What user experience are your users expecting? Is it better to be over inclusive and show more false positives or will this erode user trust? Is the cost of a false negative so high that you want to be over inclusive?

Who trains Spot?

Spot has been trained on more than **one million labels**. This data comes from multiple sources, including the crowdsourcing effort Learned Hands and individual API users (e.g., legal aid organizations running an online triage tool). These data include people's natural articulations of issues (e.g., "my landlord kicked me out") along with LIST labels indicating what issues might be relevant (e.g., HO-02-00-00-00: Eviction from a home). Spot works by finding patterns in these associations and attempting to match novel texts to what it has seen before. Consequently, high quality labeled data is the bedrock upon which Spot is built, and the accumulation of new data over time works to improve Spot's performance both in accuracy and coverage. Though Spot was originally trained primarily on Learned Hands data, more and more of its training data is now coming from other users. As we grow this community, we grow Spot's ability to recognize diverse statements of issues. For this reason, we ask that users of Spot share their data when possible. We will discuss the two main sources of training data below—*Learned Hands* and *User-Derived Data*.

Learned Hands Data.
Spot builds upon data from the Learned Hands online game, a partnership between the LIT Lab and Stanford's Legal Design Lab. Learned Hands aims to crowdsource the labeling of laypeople's legal questions for the training of machine learning (ML) classifiers/issue spotters. Labels are drawn from LIST. Currently, this labeling is limited to publicly available historic questions from the r/legaladvice forum on Reddit. See Stanford and Suffolk Create Game to Help Drive Access to Justice.

User-Derived Data.
In addition to the data labeled by Learned Hands, users of the API (those building tools with it) have the option to let Spot forget or remember the content of text shared with it. If Spot is given permission to remember a text, we may use it to improve the issue spotter by having humans perform their own issue spotting and using their insights to retrain the issue spotter.

Institutional users of the APII, like legal aid organizations, may also share information about a user's actions after seeing results from Spot. This can provide information about whether or not Spot was correct.

Additionally, institutional users may choose to share bundled historic data with Spot (e.g., previously collected data from a website chatbot or webform). By labeling this data before

[3] Values are weighted based on the number of affirmative examples for each issue in our dataset.

sharing, they can provide valuable training material to make Spot more responsive to their client base.

What are the costs and benefits of letting Spot remember user data?
If Spot is given permission to remember a text, or if a text is shared with our team as part of a bundle, it may be read by people on our team. *We do not sell this data to third parties and only share it with a closed group working on quality control and labeling.* If the text is labeled it may be used to help improve Spot's performance by acting as an example for training of our algorithms. In this way, sharing a text can help others with similar issues by making it easier for Spot to identify issues.

This sharing is really important for populations not represented in the Learned Hands data. Different communities talk about issues in different ways, and in order for Spot to recognize issues in a text, it needs to have seen them talked about in that way before. Consequently, sharing data with Spot is the best way to improve its performance.

How to talk about data sharing
Developers are encouraged to consider their use case carefully when deciding how to incorporate end user input regarding the remembering of texts. For most cases it will be prudent to have the end user either opt in or opt out. Rarely is it appropriate to hardcode a universal approach. Given the benefit that accrues to all users when data is shared, an opt out, as opposed to opt in, is encouraged for most use cases. You can find a more nuanced discussion of this on Spot's website.

Who benefits from sharing data with Spot?

Spot's long-term success is dependent on the creation of a virtuous cycle. That is, having it learn from the people it is helping so it may better help those facing similar issues in the future. Such continuous improvement relies on the creators of tools built on Spot and their end-user sharing data with Spot. Their willingness to share depends on them trusting Spot with their data. To help earn and maintain that trust, we plan to use trust law to make sure that the creators of tools based on Spot owe their users a legally enforceable duty. A limitation of traditional software licensing is that it creates duties only between parties to the license (e.g., the software author and those using it to deliver some service). By placing the fruits of Spot in trust, making those who build tools around them trustees, and defining end-users as beneficiaries, we can ensure that users are owed a duty and given a true say in how Spot is used.

Machine learning tools such as Spot derive much of their value from broad community involvement while impacting these same communities through their use. We have chosen a trust structure in part because of its ability to create enforceable responsibilities to this broader community. The Spot Click-Trust is an attempt to maximize access to, and so the reach of, Spot while including constraints necessary to obtain buy-in from data contributors that Spot depends on for continued improvement.

Consider that Spot can only recognize a statement of an issue that resembles those it has seen before. If Spot's training data is sourced from one population and used on a different one, there will be differences in how each group talks about issues, affecting Spot's ability to identify them. Additionally, different populations may experience issues at different rates, skewing their prevalence in training data. Our best defense against such bias is to make sure that we are using data from the populations we aim to serve. One way to do this is to create the virtuous cycle envisioned above, having users share data with Spot for future training. Such data sharing, however, is not always appropriate, and when appropriate, the choice to share such data should be based on trust. That is, those providing data to train Spot should be able to trust that their data will be used in a manner they support.

Such trust requires an understanding of how Spot will be used. It requires an understanding of what uses are permitted, what uses are prohibited, and how violations of expectations will be addressed. The Spot Click-Trust is the product of many conversations with stakeholders facilitated in large part by Duke Law's Digital Governance Design Studio which also proved invaluable in helping to draft the current trust language. A copy of the trust and more discussion will be available on the Spot website in June 2022. In short, it provides an assurance, with the enforcement mechanisms of trust law, that improvements derived from shared data will accrue only to API users working in the furtherance of access to justice and in the interest of those providing that data.

Suggestions for using Spot as part of an self-represented litigant portal

As discussed above, the utility of a tool is measured in comparison to the alternatives. One must always ask, “compared to what?” Keyword search and well-curated menus are powerful tools, and Spot is not positioned to replace these, esp. given that such tools are expected by users visiting a modern website. Spot is useful in helping users discover what it is they need when they don’t know the correct term of art to make use of such tools. Consequently, the use of Spot should be approached with a “yes, and” framing. This could look something like the landing pages of the MA Legal Resource Finder where a text prompt feeding into Spot is presented alongside a standard menu of topics. This allows users to enter a description of their issue or pick directly from the menu.

An improvement on the standard menu in cases where the proper term of art may not be known by the user is the decision tree. Consider a scenario where the menu on the landing page contains general topics like *housing*, and *consumer debt*. Once a user clicks on a high-level topic, they are presented with a set of questions designed to get at their specific needs. Each question in such a decision tree leads the user on a journey to a specific need. Of course, such journeys can be long and winding. Here Spot provides an assist to the user. If the final destination for each path in the tree along with each node, fork in the tree, is associated with a LIST item, Spot’s predictions can be used to jump users down the decision tree. That is, instead of a user working through a decision tree, a user can answer a prompt in natural language, and based on that answer, they could be brought as close to the final decision as warranted by the confidence of Spot’s predictions.

The above behavior describes at least four live Spot implementations. These include the MA Legal Resource finder along with tools from CT Law Help, Illinois Legal Aid Online, and Pine Tree Legal Assistance. At this time, these sites are not using Spot to jump very deeply into their decision trees. As Spot improves, however, this option will become more and more viable. In fact, the current implementations are actually helping to bring this vision into reality because most of them are sharing information about the final destination of users with Spot. This allows us to connect text inputs with probable issues as labeled by these destinations. Consequently, Spot can learn from these interactions and improve its ability to detect more fine-grained issues.

As it stands, users visiting one of the above sites can either answer a question like “What seems to be the problem?” or select from a set of topics. In our experience, about a third of users choose the text prompt. If the user chooses the text prompt they are presented with a selection of possible issues to confirm what they want to explore. This filtering is needed for multiple reasons. Remember, often there is a collection of co-occurring issues and there needs to be a way for users to select their path, additionally as we noted above, all models are wrong. Consequently, their output shouldn’t be the last word. For example, Spot may correctly identify that a fact pattern describing an accident implicates traffic violations and matters relating to personal injury, but the user is interested only in issues relating to personal injury. Presented with the issues Spot found, they can choose which to learn more about. After making their selection they are then transported to the appropriate point in the decision tree, jumping the line over a user who opted to simply choose a topic.

Of course, at this stage in its development, Spot may not be able to identify all of the issues addressed by a portal’s resources. The number of issues Spot can identify is growing over time, but a supplementary method of issue discovery is recommended. If implemented in the right way, such a supplement can also serve to help improve Spot. When deciding what issues to show a user after they answer the prompt, the above sites pull from the list provided by Spot and at least one other source (e.g., keyword search). So if Spot can’t identify an issue based on the text, but a traditional search can, that's a win, likewise, if a traditional search fails but Spot succeeds. Of course, there are also those cases where both methods agree. Since the above sites share the user’s final destination with Spot regardless or whether their entry into the tree was based on an issue found by Spot of the alternate search, Spot can actually learn to identify issues outside of its active labels.

For portals without the resources needed to construct a robust decision tree, it is worth recognizing that a special case of the above approach is that where there is no decision tree only a set of resources. In such cases, the user is not jumped down the decision tree because there is only one level. Rather, the list of all possible resources is filtered to include only those implicated by the search results (e.g., a combination of Spot and keyword search results).

Ultimately, for Spot to be helpful in the context of resource discovery, it is necessary to label all potential resources with issues from LIST. Once this is done, however, Spot’s insights can be leveraged to connect users with resources based on their narrative description of a fact pattern.