Statistical Context Analysis and Search Quality

R. T. Oehrle & E. A. Johnson

dick.oehrle@ey.com eric.johnson2@ey.com

Ernst & Young LLP[1]

Applications of machine learning techniques to eDiscovery, information governance, and other related forms of classification and categorization continue to improve in performance and efficiency, but traditional search-based methods of exploration and classification—searches for keywords, key phrases, proximity searches, regular expression searches, and their Boolean combinations—have not disappeared. Part of their persistence lies in their suitability for specialized tasks such as identification of privileged or hot documents. Even in the presence of machine learning techniques to relevance filtering and information governance classification, search-based methods often play a role—as in the initial identification of seed sets used for initial training. In fact, it's hardly surprising to see search-based methods used to support machine learning techniques and technologically assisted review. In this paper, we look at a way in which statistical techniques can be used to increase search quality, focusing on increasing precision using automated and semi-automated tools to analyze the contexts immediately surrounding search-term matches.[2]

Standardly, search quality is assessed at the document level, depending on two properties: whether a document is relevant to the goal of the search and whether the document is returned by the search. Precision—the ratio of relevant documents returned to all documents returned—is a measure of correctness. If we know (or can estimate) the number of relevant documents in the search universe, we can define the corresponding measure of completeness known as recall as the ratio of relevant documents returned to the known or estimated total of relevant documents in the search universe.

Büttcher et al. describe a Boolean query designed to address TREC topic 426—a topic created for TREC 1999 involving 'the use of dogs worldwide for law enforcement purposes' [1, p. 25]. The proposed query [1, p. 67] takes the form:

(("law" AND "enforcement") OR "police") AND ("dog" OR "dogs").

This query was run on the TREC45 collection, returning 881 of the roughly 500,000 documents in the collection. The authors comment:

[1]This material in this paper represents the views of the authors and should not be attributed institutionally or in any other way to EY. We received substantial benefits from two sets of reviewer comments, which we hereby acknowledge with thanks.

[2]A reviewer notes affinities with the work of Xu and Croft [4], who investigate the use of what they call 'local context analysis' for query expansion, primarily with the goal of increasing recall. For Xu and Croft, a 'global technique requires some corpuswide statistics that take a considerable amount of computer resources to compute', while a 'local technique processes a small number of top-ranked documents retrieved for a query to expand that query. A local technique may use some global statistics such as the document frequency of a term, but such statistics must be cheap to obtain. The source of expansion terms is the set of top-ranked documents.' [4, p. 80]. Xu and Croft also discuss a refinement of this approach in which individual documents are replaced by 'passages': 'fixed-length text windows, typically 300 words long' [4, p. 89]. Our target in this paper is the analysis of the immediate context around (and including) a simple search term match. This might seem to be a minimal kind of passage, but our goal is not to find new terms locally related to given terms, but rather to replace simple queries with their immediately contextualized phrasal counterparts, which can be analyzed (in a way that the simple queries cannot) in a supervised learning setting in a way that leads to improved performance, as discussed in the exposition below.

> According to official NIST judgments, there are 202 relevant documents in the TREC45 test collection for topic 426. Our query returns 167 of these documents, giving a precision of 0.190 and a recall of 0.827. A user may find this result acceptable... [1, p. 67]

If we don't find this result acceptable, we can try to improve the query.

On the recall side, we can explore (by sampling or other methods) the documents not returned by the search, with the goal of expanding the search (manually or in some automated way) with additional high-precision terms designed to match the missing documents and those sufficiently like them to be probably responsive. On the precision side, we might sample the non-relevant documents returned and attempt to alter the query in such a way that it returns all or almost all the relevant items, but fewer—perhaps many fewer—non-relevant items. The critical part of this attempt involves the analysis of search matches. Ideally, we would like our search query to be matched in all and only relevant documents. Practically, this means that we want to find appropriate matches in relevant and non-returned documents and add these matches to the query; at the same time, we want to find inappropriate matches and prevent them from being matched by the query. How can we apply automated tools and techniques to increase the speed and quality of this process? One way is to focus on the immediate context surrounding a search match (sometimes referred to in the literature as 'key words in context' or 'kwic'), consisting of the match, with additional tokens preceding and following the match.[3]

It is often possible to score search matches as high-value or low-value indicators of relevance, based solely on their immediate context. Below, we display two contexts for (case-normalized) matches of the search term *privileged* in the Enron EDRM dataset.

1007 this email may be confidential and/or privileged. This email is intended to be reviewed
26 Subject: PRIVILEGED & CONFIDENTIAL ATTORNEY-CLIENT COMMUNICATION ...

The first is highly likely to be a boilerplate disclaimer—and thus, since such disclaimers are often attached automatically to outgoing electronic mail, a low-value indicator of potential privilege. The second is a much more high value indicator of potential privilege: it occurs in the subject line, it is upper case, it isn't weakened by the replacement of the conjunction with a disjunction or weakened by the presence of *may*. This is the first step to being able to distinguish disclaimers from non-disclaimers automatically (at the match level), rather than manually (at the document level). Given enough context, boilerplate disclaimers are easily and reliably identifiable. If we could ignore the boilerplate matches, we could also avoid reviewing as potentially privileged documents whose only indicators of privilege are boilerplate disclaimers, resulting in significant savings in time and expense. The same properties often hold (in our experience) much more broadly: natural language expressions—and especially natural language search terms—are ambiguous, but the numerical imbalances between matches involving one interpretation and matches involving another can be quite consequential. For example, searching for the term "hidden" in a case involving fraud may (and no doubt will) return a large number of documents with extracted text of the form "START HIDDEN CONTENT". After discovering unexpectected imbalances between intended matches and unintended matches, it may be possible to improve query results by removing the unwanted matches and keeping the intended matches.

In one form, this is a process of semi-automated manual context analysis: the context matches are extracted and grouped automatically into context types, and perhaps automatically sorting the types by

[3]This definition makes sense for simple keyword and phrasal searches and even proximity searches if the distance between the components of the match is small; when the distance between components is large or unconstrained (as in searching for a Boolean conjunction of queries), it is possible to extend this definition in one way or another, but with uncertain gains. We focus here on contexts immediately surrounding simple keyword and phrasal searches.

decreasing number of matches, but the review is manual. In the general case, this has important consequences. First, every document returned by a search must contain at least one search match, but it can contain many more matches. Second, as the context of a match is increased in size (say from a five-token window on the match to a seven-token window on the match), the number of match contexts grows steeply in general. This increase directly impacts the complexity of any manual intervention. (Note that boilerplate disclaimers are atypical in this respect: since they are often applied automatically or as part of a template, particular instances don't vary from one occurrence to the next, so that the result is many occurrences of very long and easily identifiable contexts.) Finally, although it's often easy to discriminate high-value from low-value search matches, the sheer number of cases can make the discriminable cases hard to find in a manual analysis. For these reasons, in the manual setting, the initially attractive goal—so simple and clear in some cases—of improving precision by removing unwanted matches (somehow) becomes more and more difficult and costly to attain.

Yet it may be possible to overcome these disadvantages of manual context analysis using statistical sampling to rank contexts automatically. Consider the following scenario:

- start with a set of simple keyword queries
- run the queries against the search universe
- extract a relatively simple context around each match (two tokens on each side, say)
- turn the contexts into queries (*context-queries*) and run them against the data (returning the same documents as the original queries)
- review a randomly selected representative sample of sufficient size
- assign two properties to each context query with a match occurring in the sample:
  - the *nr-ratio*: the ratio of non-relevant matches to total matches (that is, 1.0 if all the matches occur in documents judged non-relevant; 0.0 if all the matches occur in documents judged relevant; otherwise, between 0.0 and 1.0)
  - the *mass*: the number of matches that occur in the sample data.
- select two parameters: an nr-ratio threshold and a mass threshold
- remove any context query whose properties exceed both thresholds
- re-run the reduced set of context queries
- *prediction*: if the thresholds are sufficiently high, precision should improve and recall should not significantly decrease.

Note that at the document level, the context-queries replicate the original simple keyword queries exactly: both sets of queries return the same set of documents, because they return the same set of matches. But at the query level, the set of context-queries is in general much more finely grained than the original query set. If our only tuning mechanism is the inclusion or omission of a query component, the set of original keyword queries can be tuned only by retaining or omitting one of the original queries, but the number of ways the context-query can in general be tuned by retention or omission for each of the contexts the original query

contains matches in. If a particular context-query only returns non-relevant documents, then removing this query from the set cannot adversely affect recall and precision cannot decrease.[4]

Note as well that we require two parameters. Some measure of the balance between R and NR is obviously required if we wish to remove NR documents. But this measure by itself is not sufficient: if a particular context query returns only one document in the sample set and this document is non-relevant, then the nr-ratio associated with the query relative to this sample is 1.0, but the evidence is slender. If the query hits multiple documents and they're all non-relevant, the evidence is stronger.[5]

Two Exploratory Experiments. We tested this protocol using materials derived from TREC 2011 task 401.[6] This task is based on available Enron data and the TREC 2011 organizers have made available a set of 24,071 coding decisions on a superset of the items deemed relevant.[7] We selected a query (query ISITrFAM) from one of the participating teams, the Indian Statistical Institute. We restricted the population to the set of 24,071 documents for which we have coding decisions. (Thus, our results diverge from those reported in the TREC 2011 overview, which is based on rankings defined over the entire Enron EDRM data.) The query is given originally as an Indri query. We translated it into the Boolean query below:

(enrononline OR eol) AND (tagg OR jarnold OR orig OR dollar OR bankruptcy OR may01 OR kiodex OR active OR nature)

We ran the two conjuncts as independent disjunctive searches and intersected the results, with the results below:

R	*NR*	*Both*	*Total*	*Total - Both*	*Precision*	*Recall*	*F1*
2151	306	129	2586	2457	2151/2457 = 0.875	2151/7333 = 0.293	0.439

We then retrieved contexts, each containing the original query match along with two tokens on either side, for *all* the matches[8] in the 2,586 documents: 569,951 context match occurrences, based on 21,118 distinct context match types. We turned each distinct context match type into a phrasal query (containing the left context, the original query match, and the right context), ran the set of queries against the 24,071 documents in the T401 universe, intersected the results with the results of the first conjunct of the query above and obtained the same 2,586 documents in return.

We next ran two experiments in which we drew random samples of the results of the right conjunct query[9] and assigned an *nr-ratio* and a *mass* to each of the context queries. In each case, we then selected threshold values for the *nr-ratio* and the *mass*. We removed every query exceeding both thresholds and

[4]It isn't true that precision is bound to improve, because the set of non-relevant documents returned by this query might also be returned by other context queries. And if the context-query returns mostly, but not only, non-relevant documents, removing it opens the door to possible loss of relevant documents as well.

[5]A reviewer noted a striking parallel between our two parameters and the two parameters associated with *pseudo-relevance* or *blind relevance feedback*: selecting the k most relevant documents corresponds to our selection of the highest nr-ratio, properties of term weighting within this population corresponds to our mass. For discussion, see [1, §8.6.2] and [3, §9.1.6]. The parallels are worth exploring further, allowing for the difference that the goal of blind relevance feedback is to increase recall, while our goal is to increase precision.

[6]See the comprehensive overview [2] and the resources available at `http://trec-legal.umiacs.umd.edu/#2011`.

[7]Three coding values are found in the version of the list used here: *R* (7333), *NR* (16490), and *Both* (248), which we believe stems from clashing judgments for different duplicates. We use the number of documents coded *R* as the basis of recall estimates below. We ignore documents coded as *Both*.

[8]Actually, not quite all. For technical reasons, we had to remove eleven documents—evidently dupes or near dupes—which each contained 45,174 search matches.

[9]That is, without intersecting the results with the left conjunct results. This population consists of 2,852 documents—2,218 R, 491 NR, and 143 Both. The higher NR ratio of this population offers a possible training advantage over the 2,586 document universe.

re-ran the remaining set of context-queries, in each case intersecting the results with the results of the first conjunct of the original query. We then removed the randomly selected "training" data from the results, in order to ascertain how the new queries performed on the remaining "test" data.

Experiment 1. In the first experiment, the "training" data consisted of 500 randomly selected documents from the results of the context queries based on the right conjunct, consisting of 388 *R*, 84 *NR*, and 28 *Both*.[10] Each context query is assigned a 3- or 4-gram identifier and each identifier of a context query with a match in the "training" data is assigned an *nr-ratio* between 0 and 1 inclusive[11] and a positive integer mass, as illustrated below:

identifier	*nr-ratio*	*mass*
0gbz	1.0	15
uhb	1.0	11
0gdc	1.0	8
0boi	1.0	8
udj	1.0	6
...		
nzk	1.0	3
wtf	1.0	2
ham	1.0	2
...udo	1.0	1
meh	0.769	13
eqb	0.75	12
cdt	0.667	3
...		
ctv	0.5	4
...dne	0.0	8
...		

We selected the *nr-ratio* to be 0.75 or greater and the *mass* threshold to be three or greater.[12] There are 14 context queries meeting these two criteria. We removed them from the context query set, re-ran the query, intersected it with the left conjunct results, and removed the training sample from the results. The remaining results constitute our test data for this experiment. We also removed the training data from the coded results, leaving 23,571 coded items, with 6,945 *R*, 16,406 *NR*, and 220 *Both*. We used the 6,945 *R* documents in our recall calculations below.

R	*NR*	*Both*	*Total*	*Total - Both*	*Precision*	*Recall*	*F1*
1760	211	104	2075	1971	1760/1971 = 0.893	1760/6945 = 0.253	0.395

The results for the original query on this same data set are shown below:

R	*NR*	*Both*	*Total*	*Total - Both*	*Precision*	*Recall*	*F1*
1772	260	104	2136	2032	1772/2032 = 0.872	1772/6945 = 0.255	0.395

[10] It is perhaps worth emphasizing that it is documents that are sampled, not individual match occurrences or types of match occurrences, because documents are the bearers of review decisions, not match occurrences or types of match occurrences

[11] Our nr-ratio is the same as the usual notion of precision, but with the scale reversed: high nr-ratio is low precision and low nr-ratio is high precision.

[12] The parameter selection was done manually, based on inspection of the range of values and the rapid fall-off of mass below the 0.75 nr-ratio threshold.

After training and context query modification, the number of *NR* hits goes from 260 to 211—a reduction of about 18.8%—and overall precision goes from the already fairly high 0.872 to even higher 0.893. But along the way, we also lose 12 *R* documents, so recall falls from 0.255 to 0.253. The F1 score remains roughly the same.

Experiment 2. Obviously, results might improve with a larger training set. In particular, with more data, it might be possible to set higher thresholds for both *nr-ratio* and *mass*. Accordingly, we drew a second random sample from our original context-query results (for the second conjunct of the original query), increasing the size from 500 to 750 (578 *R*, 127 *NR*, 45 *Both*). After assigning values for *nr-ratio* and *mass* to each of the context queries with matches in this larger training set, we manually selected higher thresholds: 0.9 or higher for *nr-ratio* and 4 or higher for *mass*. There are 13 context-queries satisfying both criteria. As above, we re-ran the reduced set of context queries and evaluated the results after removing the training data from the relevant populations. (This leaves 23,321 coded items, 16,363 *NR*, 6,755 *R* (down from 6,945 in the earlier test universe), and 203 *Both*. We used the 6,755 remaining *R* items in the recall calculations below.)

The table below displays the results of the newly trained query.

R	*NR*	*Both*	*Total*	*Total - Both*	*Precision*	*Recall*	*F1*
1586	183	87	1856	1769	1586/1769 = 0.897	1586/6755 = 0.235	0.372

In comparison, when the original query results are evaluated on this test set, we have the results shown below:

R	*NR*	*Both*	*Total*	*Total - Both*	*Precision*	*Recall*	*F1*
1588	233	87	1908	1821	1588/1821 = 0.872	1588/6755 = 0.235	0.370

In this case, the trained query removes 50 of 233 *NR* items (over 21%), with a loss of only two responsive documents (less than 0.126%).

Discussion. These two experiments are exploratory in nature, rather than definitive. In both experiments, it's clear that the results are very far from what would be expected if a random sample of documents (61 documents in the first case, 52 in the second) were selected from the original set and removed to form the second set. But the exact number of documents in each case is dependent on the queries and the distribution of their matching contexts across documents. A different approach to evaluating significance, and perhaps a less straightforward one, is to compare the results obtained from the results of randomly selecting 10 or 11 or 12 context queries from the entire set of context queries, omitting them from the context query set and obtaining new results for comparison. (The fact that the nr-ratio and mass parameters have been set manually outside the experimental set-up doesn't make the comparison entirely clean and easy.) Additional questions arise with respect to the data selected (especially its representativeness), the query selected, and the sample sizes selected.

To address these questions in future work, we plan to fix in some way the method of determining the *nr-ratio* and *mass* thresholds and then to run a series of experiments, using Monte Carlo methods, to determine whether the protocol described here converges favorably under a variety of parametric settings, on a broader, more representative class of data.[13] We believe that the exploratory experiments reported above justify further and more definitive experimentation.

[13] A reviewer suggested applying the techniques described here to the TREC Legal CDIP collection, for which topics, associated Boolean queries, and partial relevance judgments are available.

References

[1] Stefan Büttcher, Charles L. A. Clarke, and Gordon V. Cormack. *Information Retrieval: Implementing and Evaluating Search Engines*. MIT Press, Cambridge, Massachusetts, 2010.

[2] Maura R. Grossman, Gordon V. Cormack, Bruce Hedin, and Douglas W. Oard. Overview of the TREC 2011 Legal Track. In *NIST Special Publication: SP 500-296: The Twentieth Text REtrieval Conference (TREC 2011) Proceedings*, 2011. http://trec.nist.gov/pubs/trec20/t20.proceedings.html.

[3] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, 2008.

[4] Jinxi Xu and W. Bruce Croft. Improving the effectiveness of information retrieval with local context analysis. *ACM Transactions on Information Systems*, 18(1):79–112, 2000.