Computer-Assisted Keyword and Document Set Discovery from Unstructured Text[*]

Gary King[†] Patrick Lam[‡] Margaret E. Roberts[§]

Abstract

The (unheralded) first step in many applications of automated text analysis involves selecting keywords to choose documents from a large text corpus for further study. Although all substantive results depend on this choice, researchers usually pick keywords in ad hoc ways that are far from optimal and usually biased. Paradoxically, this often means that the validity of the most sophisticated text analysis methods depend in practice on the inadequate keyword counting or matching methods they are designed to replace. Improved methods of keyword selection would also be valuable in many other areas, such as following conversations that rapidly innovate language to evade authorities, seek political advantage, or express creativity; generic web searching; eDiscovery; look-alike modeling; intelligence analysis; and sentiment and topic analysis. We develop a computer-assisted (as opposed to fully automated) statistical approach that suggests keywords from available text without needing structured data as inputs. This framing poses the statistical problem in a new way, which leads to a widely applicable algorithm. Our specific approach is based on training classifiers, extracting information from (rather than correcting) their mistakes, and summarizing results with Boolean search strings. We illustrate how the technique works with analyses of English texts about the Boston Marathon Bombings, Chinese social media posts designed to evade censorship, among others.

Word count: 8,885

[*] Our thanks to Dan Gilbert, Burt Monroe, Brandon Stewart, Dustin Tingley, and the participants at the Society for Political Methodology conference for helpful suggestions.

[†] Institute for Quantitative Social Science, 1737 Cambridge Street, Harvard University, Cambridge MA 02138; GaryKing.org, King@Harvard.edu, (617) 500-7570.

[‡] Institute for Quantitative Social Science, 1737 Cambridge Street, Harvard University, Cambridge, MA 02138; PatrickLam.org

[§] Assistant Professor, Department of Political Science, University of California, San Diego, Social Sciences Building 301, 9500 Gilman Drive, #0521, La Jolla, CA 92093-0521; (617) 302-6020; meroberts@ucsd.edu, MargaretRoberts.net

1 Introduction

Boolean keyword search of textual documents is a generic task used in numerous methods and application areas. Sometimes researchers seek one or a small number of the most relevant documents, a use case we call *fact finding* and for which the Google and Bing search engines were designed. For example, to find the capital of Montana, a weather forecast, or the latest news about the president, the user only wants one site (or a small number of sites) returned. In the second *collecting* use case, which we focus on, researchers do not try to find the needle in the haystack, at least at first; instead, they seek all documents that describe a particular literature, topic, person, sentiment, event, or concept.

Collecting is presently performed by attempting to think of all keywords that represent a specific concept, and selecting documents that mention one or more of these. Although this selection process is known to be a "near-impossible task" for a human being (Hayes and Weinstein 1990), many applications require it. For example, applications of sophisticated methods of automated text analysis, designed to get around simplistic keyword matching and counting methods, are often preceded by selecting keywords to narrow all available documents to a manageable set for further analysis. Similarly, search engines are optimized for fact finding, but regularly used for collecting, even though they are suboptimal for this alternative purpose. Indeed, as we discuss in Section 3, human brains have well-studied inhibitory processes that, although adaptive for other reasons, explicitly prevent us from recalling keywords when needed for the task of collecting.[1]

The problem of keyword discovery is easier when structured data is available to supplement the raw text, such as search query logs (e.g., Google's AdWords Keyword Tool, or Overture's Keyword Selection Tool), databases of meta-tags, or web logs (Chen, Xue and Yu 2008), and a large literature of methods of "keyword expansion or suggestion" has arisen to exploit such information. In this paper, we develop methods for the wide array of problems for which raw text is the sole, or most important, source of information. To avoid requiring a human user having to think of all relevant keywords, we introduce methods of

[1] Some algorithms have been proposed and implemented on search engines to provide assistance for collecting, but the approaches are based on methods of fully automated cluster analysis that perform poorly on most general problems (Grimmer and King 2011).

computer-assisted keyword discovery. Our key motivating principle is, although humans are terrible at recalling words from memory, they are very good at judging whether words given to them are appropriate representations of a given concept.

We begin by describing some of the application areas to which our methodology may provide some assistance (Section 2). We then conduct an experiment that illustrates the remarkable unreliability of human users in selecting appropriate keywords (Section 3). We then define the statistical problem we seek to solve, along with our notation (Section 4). We then present our algorithm (Section 5), several ways of evaluating it (Section 6), and a illustrate of how it works in practice (Section 7). Lastly, we discuss related prior literature (Section 8) and conclude (Section 9). The appendices give details on algorithm robustness and how to build queries for much larger data sets.

2 Application areas

Algorithms that meet the requirements of the statistical problem as framed in Section 4 suggest many new areas of application. We list some here, all of which the algorithm we introduce below may help advance. Some of these areas overlap to a degree, but we present them separately to highlight the different areas form which the use of this algorithm may arise.

Conversational Drift Political scientists, lobby groups, newspapers, interested citizens, and others often follow social media discussions on a chosen topic but risk losing the thread of the conversation, and the bulk of the discussion, when changes occur in how others refer to the topic. Some of these wording changes are playful or creative flourishes; others represent political moves to influence the debate or frame the issues. For example, what was once called “gay marriage” is now frequently referred to by supporters as “marriage equality”. Progressive groups try to change the discussion of abortion policy from “pro-choice” and “pro-life” where the division is approximately balanced to “reproductive rights,” where they have a large majority. Conservatives try to influence the debate by relabeling “late term abortion” as “partial birth abortion,” which is much less popular. As these examples show, selecting an incomplete set of keywords can result in severe selection bias, because of their correlation with the opinions of interest.

Evading the Censors Internet censorship exists in almost all countries to some degree. Governments and social media firms that operate within their jurisdictions use techniques, such as keyword-based blocking, content filtering, and search filtering, to monitor and selectively prune certain types of online content (Yang 2009). Even in developed countries, commercial firms routinely "moderate" product review forums, and governments require the removal of "illegal" material such as child pornography. In response to these information controls, netizens continually try to evade censorship with alternative phrasings. For example, immediately after the Chinese government arrested artist-dissident Ai Weiwei, many social media websites began censoring the Chinese word for Ai Weiwei (King, Pan and Roberts 2013); soon after, netizens responded by referring to the same person as "AWW," and the Chinese word for "love", which in Chinese sounds like the "ai" in "Ai Weiwei". Other creative censorship avoidance techniques involve using homographs and homophones.

Starting Point for Statistical Analyses of Text Most methods of automated text analysis assume the existence of a set of documents in a well-defined corpus in order to begin their analysis. They then spend most of their effort on applying sophisticated statistical, machine learning, linguistic, or data analytic methods to this given corpus. In practice, this corpus is defined in one of a variety of ways but keyword searching is a common approach; e.g., Eshbaugh-Soha (2010), Gentzkow and Shapiro (2010), Hopkins and King (2010), Ho and Quinn (2008), King, Pan and Roberts (2013), Puglisi and Snyder (2011). In this common situation, our algorithm should help improve the inputs to, and thus the results from, any one of these sophisticated approaches. The same issue applies for simple analysis methods, such as keyword counting.

Intuitive and Infinitely Improvable Classification Because statistical classifiers are typically far from perfect (Hand 2006), ordinary users who find individual documents misclassified may question the veracity of the whole approach. Moreover, since most classifiers optimize a global function of the data set, even sophisticated users may find of value hybrid approaches for adding human effort and knowledge to improve classification at the level of smaller numbers of documents. In this situation, keyword-based classifiers are sometimes more useful because the reasons for mistakes, even if there are more of them, are readily understandable and easily fixable (by adding or removing keywords from the selection list) for a human user (Letham et al. 2015). Keyword classifiers are also much faster than statistical classifiers and can be improved to any higher level of accuracy, with sufficient effort, by continual refinement of the Boolean query.

Online Advertising Academics recruiting study participants often bid for ad space next to searches for chosen keywords (Antoun et al. 2015), just as firms do in advertising campaigns. This is common with Google Adwords, Bing Ads, Facebook, etc. These systems, and other existing approaches, suggest new keywords to those spending advertising dollars by mining information from structured data such as web searches, weblogs from specific websites, or other ad purchases. Our approach can supplement these existing approaches by mining keywords relevant to the population of interest from raw unstructured text found in research documents, literature reviews, or information in private companies such as customer call logs, product reviews, websites, or a diverse array of other sources. Whereas keywords (or more general Boolean searches) for advertising on search engines can be mined from search engine query logs, or website logs, keywords that identify rarely visited pages, or for advertising on social media sites, can only be mined from the unstructured text.

Long Tail Search Modern search engines work best when prior searches and the resulting structured metadata on user behavior (such as clicking on one of the web sites offered or not) are available to continuously improve search results. However, in some areas, such metadata is inadequate or unavailable, and keywords must be discovered from the text alone. These include (1) traditional search with unique or unusual search terms (the “long tail”), (2) searching on social media, where most searches are for posts that just appeared or are just about to appear, and so have few previous visits, and (3) enterprise search for (confidential or proprietary) documents that have rarely if ever been searched for before. In these situations, it may be useful to switch from the present fully automated searching to computer-assisted searching using our technology.

Consider social search. During the Boston Marathon bombings, many followed the conversation on Twitter by searching for *#BostonBombings*, but at some point the social media Boston authors expressed community spirit by switching to *#BostonStrong* and out-of-towners used *#PrayForBoston*. Since guessing these new keywords is nearly impossible, those who didn't notice the switch lost the thread of the conversation.

3 The Unreliability of Human Keyword Selection

Human beings, unaided by computers, seem to have no problem coming up with *some* keywords to enter into search engines (even if not the optimal ones). Everyone is accustomed to doing Google searches, after all. However, as we demonstrate in this section for the more complicated task of choosing a *set* of keywords for the task of collection, even expert human users perform extremely poorly and are highly *unreliable* at this task. That is, two human users familiar with the subject area, given the same task, usually select keyword lists that overlap very little, and the list from each is a very small subset of those they would each recognize as useful after the fact. The unreliability is exacerbated by the fact that users may not even be aware of many of the keywords that could be used to select a set of documents. And attempting to find keywords by reading large numbers of documents is likely to be logistically infeasible in a reasonable amount of time.

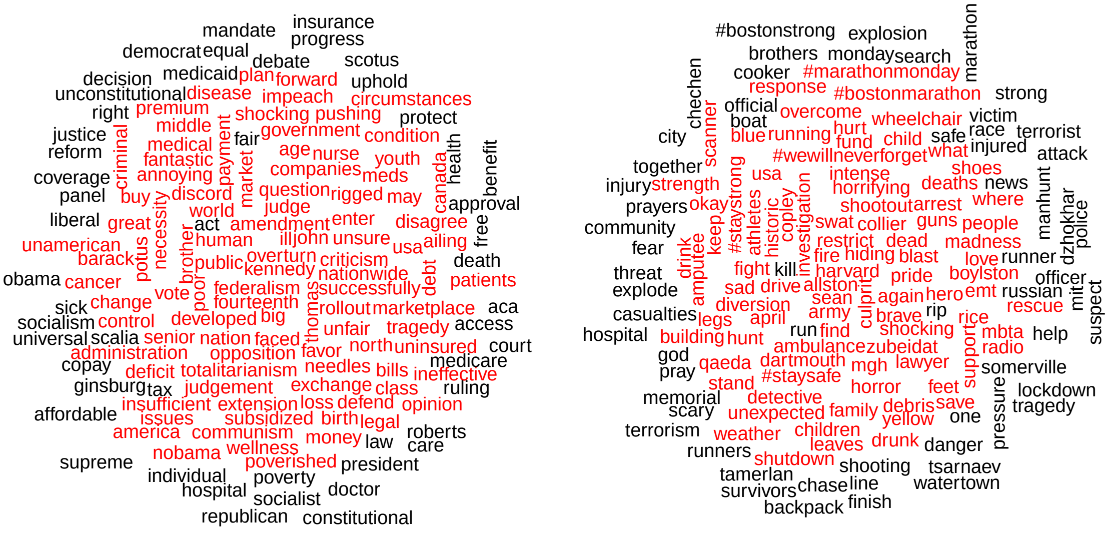
Here, we first demonstrate this surprising result with a simple experiment. Second, because this result is counterintuitive ex ante, we briefly summarize the well-developed psychological literature that can be used to explain results like this. And finally, we show the severe statistical bias (or extra ex ante variance) that can result from selecting documents with inadequate keyword lists.

Experiment For our experiment, we asked 43 relatively sophisticated individuals (mostly undergraduate political science majors at a highly selective college) to recall keywords with this prompt:

> We have 10,000 twitter posts, each containing the word "healthcare," from the time period surrounding the Supreme Court decision on Obamacare. Please list any keywords which come to mind that will select posts in this set related to Obamacare and will not select posts unrelated to Obamacare.

We also gave our subjects access to a sample of the posts and asked them not to consult other sources. We repeated the experiment again with an example about the Boston Marathon bombings.

The median number of words selected by our respondents were 8 for the Obamacare example and 7 for the experiment about the Boston Marathon bombings. In Figure 1, we summarize our results with word clouds of the specific keywords selected. Keywords selected by one respondent and not by anyone else are colored red (or gray if reading black and white). The position of any one word within the cloud is arbitrary.



(a) Obamacare

(b) Boston Bombings

Figure 1: The Unreliability of Human Keyword Selection. Word clouds of keywords selected by human users; those selected by one and only respondent are in red (or gray if printed in black and white). The position of each word within the cloud is arbitrary.

The results clearly demonstrate the remarkably high level of unreliability of our human keyword selectors. In the Obamacare example, 149 unique words were recalled by at least one of our 43 respondents. Yet, for 66% of those words, every single one of the remaining 42 respondents, when given the chance, failed to recall the same word (Figure 1, red or gray words in the left panel). In the Boston Marathon bombing example 59% failed (right panel). The level of unreliability was so high that no two users recalled the same entire keyword list.

This extreme level of unreliability is not due to our research subjects being unaware of some of the words. Indeed, after the fact, it is easy to see from Figure 1 that almost all the words recalled are recognizably related to the Boston bombings or Obamacare, respectively. In other words, although humans perform extremely poorly at recall, they are excellent at remembering.

Psychological Foundation The counterintuitive result from our experiment is related to, and can be explained by, psychological research on "inhibitory processes" (and in particular "part-list cuing"). The well supported finding, from many experiments, is that revealing one word to the research subject facilitates remembering others, but the cue provided by revealing more than a few words strongly *inhibits* recall of the rest of the set, even though you would recognize them if revealed (Roediger and Neely 1982, Bauml 2008).

Why our brains would be constructed to stop us from remembering needed information deserves at least some speculation. One way to think about this is imagining memory as a network diagram with concepts represented as nodes, and connections between concepts represented as edges. Without inhibitory processes, activating any one concept by recall would activate all concepts connected to it, and all those connected to those, etc. (Orange activates apple, apple activates banana, banana activates slip, slip activates. . . .). Millions of concepts would come flowing into your comparatively tiny short term, working memory and, unable to handle it all, you would likely be overwhelmed and perhaps unable to think at all. So either working memory would need to be much bigger, which doesn't seem to be on offer, or inhibitory processes are necessary.[2]

Consequences for Statistical Bias As is well known, the choice of a data selection rule, such as defined by the choice of keywords, is only guaranteed to avoid bias if it is independent of the variables used to analyze the chosen document set. Obviously, this is a strong assumption, unlikely to hold in many appliations, especially when using unreliable (i.e., human-only) methods of keyword selection. In other words, different keyword lists generate different document sets which, in turn, can lead to dramatically different inferences, substantive conclusions, and biases.

We now demonstrate these biases in an analysis of the data from our Boston Marathon bombings experiment. We study the well known tendency for communities suffering a tragedy to turn public discourse from the obvious negative events into positive expressions based on solidarity, community spirit, and individual heroics. To do this, we use a simple, but still very common, analysis measure (Nielsen 2011). The idea is to code each word in a social media post as having negative (-1), neutral (0) or positive ($+1$) sentiment (based on a fixed dictionary designed for Twitter), and to sum all the words in a post to give the final sentiment for that tweet. We use this method to compute the average sentiment of all tweets retrieved by each of the 43 keyword lists from our 43 subjects. The point estimates (dots) along with 95% confidence intervals (horizontal lines) for each appear in Figure 2, sorted from negative to positive sentiment.

The results vividly demonstrate the substantial effect the choice of a keyword list has on the sentiment of the document sets chosen by different research subjects given the identical prompt. Choosing some of the lists (on the bottom left) would lead a researcher to the conclusion that social media discourse was extremely negative during the month following the Boston Marathon bombing. If, instead, one were to choose other keyword

[2] We can make this strange result somewhat more plausible by turning on an inhibitory process in your brain right now: Think of your bank password. Now think of your previous bank password. Assuming you listen to your bank and don't rotate them, now think of your bank password before that. Likely you can't remember that one, but if someone showed it to you, we think you would agree that it would be easy for you to recognize it as correct. If so, then we have shown that the memory of that third password exists in your brain, even though something is causing you to not be able to access it. An example of inhibitory processes at work may even be the feeling that a thought you're having trouble remembering is "on the tip of your tongue": it is stored in your brain but you cannot access it.

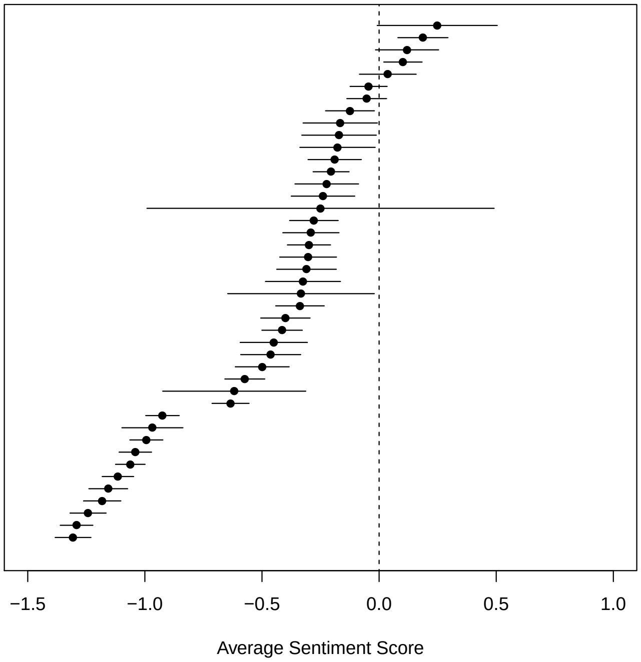


Figure 2: Average Sentiment of 43 Document Sets, each selected by a different keyword list, with point estimates (as dots) and 95% confidence intervals (horizontal lines).

sets (which appear in the middle of the graph), a researcher could report "evidence" that sentiment was only slightly negative. Alternatively, a researcher who used one of the keyword lists from the top right would be led to the conclusion that sentiment was relatively positive (by selecting documents that reflected expressions of community spirit). As is evident, almost *any* substantive conclusion can be drawn from these data by changing choice of the keyword list. This example clearly demonstrates the value of paying far more attention to how keyword lists are selected than has been the case in the literature.

4 Defining the Statistical Problem

Notation We define the *reference set*, R, to be a set of textual documents, all of which are examples of a single chosen concept of interest (e.g., a topic, sentiment, idea, person, organization, event, etc.). This set is defined narrowly so that the probability of documents being included that do not represent this concept are negligible. The reference set need not be a random or representative sample of all documents about the concept of interest (if such a process could even be defined), and may even reflect a subset of emphases or aspects of the concept (as was common for individual humans in Section 3).

Also define the *search set*, S, as a set of documents that does not overlap the reference set, $R \cap S = \varnothing$. Our goal is to identify a *target set* T, which is a subset of the search set ($T \in S$), containing documents with new examples of the concept defining documents in the reference set. Ultimately, we are interested in $T \cup R$ but, since we have R, the statistical task is to find T.

In practice, the reference set may be defined by choosing individual documents by hand, selecting an existing corpus, or using all available documents that contain text matching a specific Boolean query Q_R (defined as a string containing user-defined keywords and Boolean operators, AND, OR, NOT, such that $R = \{d : Q_R\}$, for any document d under consideration). The search set can be defined as all web sites on the internet (after removing documents in R), all available documents, a different selected existing corpus, or documents that match a Boolean query, Q_S (such that $S = \{d : Q_S\}$). The elements of a Boolean query are "keywords."[3]

An Unsupervised Statistical Problem The statistical task of finding T is "unsupervised" in that the concept defining the reference and target sets may be broadened by the human user on the fly as part of the process of discovery (rather than, as in "supervised" analyses, T being a fixed quantity to be estimated). We thus seek to identify the target set T by first finding K_T, the set of all keywords in T ranked by likely relationships with the concept. We then use human input in specific ways to craft query Q_T, intended to retrieve T from S. Depending on application, users may also be interested in the set of all keywords in the reference set K_R, the target and reference sets together $T \cup R$, a query that returns both the reference and target sets together Q_{RT}, or all of the above.

Our algorithm is computer-assisted (human powered) rather than fully automated. The

[3] The simplest version of keywords are unigrams, but they could also include higher-order n-grams, phrases, or any type of Boolean query. Common steps in automated text analysis, such as making all letters lower case or stemming, can broaden the words that a single keyword will match (e.g., "consist" would then match "consist" as well as "Consist," "consistency," "Consisted," "CONSISTING," etc.). Other standard text analytic pre-processing steps would remove words from the possible list of keywords, such as by removing stopwords or other very common words or very short words.

more common fully automated approaches to document retrieval (such as spam filters) use statistical or machine learning classifiers that are viewed as a black box to the user. By restricting ourselves to a simple Boolean search, defined by a set of interpretable keywords, we empower users to control, understand, and continually improve the retrieval process.

Another reason for the choice of a human powered approach is that the concept which the documents in the reference set share, and for which we seek a target set, is not a well-defined mathematical entity. Human language and conceptual definitions are rarely so unambiguous. For example, any two nonidentical documents could be regarded as the same (they are both documents), completely unrelated (since whatever difference they have may be crucial), or anything in between. Only additional information about the context (available to the person but not available solely from the data set) can informatively resolve this indeterminacy. To take a simple example, suppose one element of K_R is the keyword “sandy”. Should the target set include documents related to a hurricane that devastated the New Jersey, a congresswoman from Florida, a congressman from Michigan, a cookie made with chopped pecans, a type of beach, a hair color, a five letter word, or something else? To make matters worse, it could easily be the case that documents in the reference set represent two of seven of these examples, but two others in the search set are of interest to the human user. Of course, a user can always define the reference set more precisely to avoid this problem, but the nature of language means that some ambiguity will always remain. Thus, we use human input, with information from the text presented to the human user in a manner that is easily and quickly understood, to break this indeterminacy and grow the reference set in the desired direction.

5 Algorithm

The algorithm first partitions S into two groups by classifying whether a document belongs in sets T or its complement, $S \setminus T$. It mines S for all keywords K_S, and then ranks keywords by how well they discriminate between T and $S \setminus T$. This results in two lists of keywords K_T and $K_{S \setminus T}$ ranked in order of how well they discriminate each set from the other. The keyword lists themselves are often of interest to users who would like keyword recommendations for various uses. For document retrieval, the user would iterate through the two lists to produce a query Q_T that, when combined with the reference query Q_R to form Q_{RT}, best retrieves their desired document set of interest.

Table 1 gives a brief overview of the specific steps in our proposed algorithm.

5.1 Incrementally Defining R and S

The simplest application of our algorithm has R and S defined at the outset, but alternatives are often easier in practice. For example, one may begin with a large document set and without any immediately obvious distinction between the two sets. This situation is common with large, continuously streaming, or even ill defined data, such as based on the entire internet, all social media posts, or all documents narrowed by a set of very broad keywords, etc. In this situation, we can define S and R adaptively, as part of the algorithm (e.g., D'Orazio et al. 2014).

Consider the following alternative adaptive strategy. The user begins by defining R narrowly based on one simple keyword search, as a subset of the existing corpus. We then add an intermediate step to the algorithm which involves mining and displaying a list of keywords found in R, K_R, ranked by a simple statistic such as document frequency or tf-idf. The user then examines elements of K_R (aside from those used to define the set) and chooses some keywords to define Q_S, which in turn generates a definition for S, so that we can run the rest of the algorithm. The user can then continue to add keywords from K_R into the final desired query Q_{RT}. In this workflow, S can be neither pre-defined nor retrieved ex ante. This step also mitigates the issue of how to define a search set in large data sets that do not fit into memory all at once or may not even be able to be retrieved all at once. It also leverages additional information from R in the form of keywords likely to identify additional aspects of the concept and keywords the user may not have thought of for defining both R and S.

1. Define a reference set R and search set S (following Section 4).
2. Using a diverse set of classifiers, partition all documents in S into two groups: T and $S \setminus T$, as follows:
    (a) Define a training set by drawing a random sample from R and S.
    (b) Fit one or more classifiers to the training set using as the outcome whether each document is in R or S.
    (c) Use parameters from classifiers fit to the training set to estimate the predicted probability of R membership for each document in S. (Of course, every document *is* in S, and so the prediction mistakes can be highly informative.)
    (d) Aggregate predicted probabilities or classifications into a single score (indicating probability of membership in T) for each document in S.
    (e) Partition S into T and $S \setminus T$ based on the score for each document and a user-chosen threshold.
3. Find keywords that best classify documents into either T or $S \setminus T$, as follows:
    (a) Generate a set of potential keywords by mining S for all words that occur above a chosen frequency threshold, K_S.
    (b) Decide whether each keyword $k \in K_S$ characterizes T or $S \setminus T$ better, by comparing the proportion of documents containing k in T with the proportion of documents containing k in $S \setminus T$.
    (c) Rank keywords characterizing T by a statistical likelihood score that measures how well the keyword discriminates T from $S \setminus T$. Do the analogous ranking for keywords characterizing $S \setminus T$.
4. Present keywords in two lists to the user, to iterate and choose words of interest or for use in building a document retrieval query.
5. If sufficient computational power is available, rerun steps 1–4 every time the user makes a measurable decision, such as adding a keyword to Q_T to improve the lists of keywords to consider.

Table 1: *The Keyword Algorithm:* A simple version of our algorithm, used in illustrations below. The algorithm also has numerous possible extensions, such as generating phrases or higher order n-grams, clustering the documents in various different ways, redefining the reference set after the user chooses a keyword, and iterating between user input and the algorithm.

5.2 Partitioning S into T and $S \setminus T$

To partition S into T and $S \setminus T$, we first we define a "training" set by sampling from S and R. We can repeat this step with different random subsettings to increase the diversity of keyword candidates that are surfaced. (Exemplars can substitute for random sampling as well.) Since R is typically much smaller than S and our test set for our classifiers is all of S, we often use the entire R set and a sample of S as our training set.

Next, we fit classifiers to the training set, using each document's actual membership in R or S as the outcome variable. As predictors, we use any element of the text of the documents, as well as any available metadata. Any set of statistical, machine learning, or data analytic classifiers can be used, but we recommend using as large and diverse a set of methods as is convenient and computationally feasible (e.g., Bishop 1995, Hastie, Tibshirani and Friedman 2009, Kulkarni, Lugosi and Venkatesh 1998, Schapire and Freund 2012).

After fitting the classifiers, we use the estimated parameters to generate predicted probabilities of R membership for all documents in S. Of course, all the search set documents in fact fall within S, but our interest is in learning from the *mistakes* these classifiers make.

Although we do not need to transform the probabilities into discrete classification decisions for subsequent steps in the algorithm, we provide intuition into these mistakes by doing this now. Table 2 portrays the results for one example classifier, with the originally defined truth in rows and potential classifier decisions in columns. We will typically be interested in documents from the search set, (mis)classified into the reference set, $\{R|S\}$. The idea is to exploit these mistakes since documents in this set will reveal similarities to the reference set, and so likely contain new keywords we can harvest to better represent the concept of interest.[4]

Once we have predicted probabilities of R membership for each document in S from

[4] Other groups defined by the classifier in Table 2 may also be useful. For example, the documents $\{S|S\}$ contains keywords in the search set, classified into the search set and so could be useful for identifying keywords to avoid when defining a topic of interest; in a Boolean query, these could be used with NOT. Similarly, the documents $\{R|R\}$ can reveal keywords that select documents in the reference group. These can be used to refine the definition of the reference or search data sets. We also use these documents for model checking and for tuning in our classifiers.

		Classified	
		Search	Reference
Truth	Search	$\{S\vert S\}$	$\{R\vert S\}$
	Reference	$\{S\vert R\}$	$\{R\vert R\}$

Table 2: Classification sets, where $\{a|b\}$ is the set of documents in set b classified into set a; S is the search set, and R is the reference set.

the classifiers, we need to turn these into a single T membership "score" for the purpose of grouping documents. For a single classifier, the predicted probability of R membership from S is the predicted probability of T membership. In most situations, we recommend the use of multiple classifiers, so that we can extract their different "opinions" about in which set individual documents belong. The different classifiers will typically pick up on different aspects of the concept, and thus highlight different keywords for the user to choose from. To ensure that this diversity of opinion is reflected in our keyword lists, we aggregate the probabilities across classifiers for a single document by taking the *maximum* probability across the classifiers as the membership score (i.e., rather than the usual approach of using the average or plurality vote). We then use this score to group documents into T and $S \setminus T$. Our simple aggregation rule thus boils down to placing all documents with at least one classifier "vote."

5.3 Discovering Keywords to Classify Documents

After partitioning S into our estimated target set T and non-target set $S \setminus T$, we must find and rank keywords that best discriminate T and $S \setminus T$. We do this in three steps: (a) mine all keywords from S (perhaps limiting our list to those that meet thresholds such as a minimum document frequency of 5 documents), (b) sort them into those that predict each of the two sets, and (c) rank them by degree of discriminatory power.

Step (a) is accomplished by merely identifying all unique keywords in S. This is a simple step for our computer algorithm, but important in practice since a human that thinks of a word not in any documents in S will be useless, no matter how compelling the word seems to be.

For Step (b), we use the proportion of documents in which each keyword appears at least once. For example, if a keyword appears in 5 out of 10 T documents and 15 out of 50 $S \setminus T$ documents, we put that keyword into the T list since it appears in 50% of T documents and 30% of $S \setminus T$ documents, despite the fact that it appears in 10 more $S \setminus T$ documents on an absolute scale. Keywords that appear in both sets with equal document proportions can be placed in either or both lists.

In Step (c), we rank the keywords within lists, according to how well they discriminate the two sets. Although different scoring metrics could be used to accomplish this task, we find that a metric based on the following likelihood approach is quite effective (see Letham et al. 2013). For document $d \in S$ at any point in using the algorithm, let y_d equal 1 if $d \in T$ and 0 if $d \in S \setminus T$. For each keyword k in either list, denote $n_{k,T}$ and $n_{-k,T}$ as the number of documents in T that do and do not match k, respectively, and $n_{k,S\setminus T}$, and $n_{-k,S\setminus T}$ as the number of documents in set $S \setminus T$ that do and do not match k, respectively. Also define the marginal totals so that $n_{k,S} = n_{k,T} + n_{k,S\setminus T}$ and $n_{-k,S} = n_{-k,T} + n_{-k,S\setminus T}$ denote the total number of documents in S that do and do not contain k, respectively, and $N_T = n_{k,T} + n_{-k,T}$ and $N_{S\setminus T} = n_{k,S\setminus T} + n_{-k,S\setminus T}$ denote the number of documents in T and $S \setminus T$, respectively.

This then leads to a convenient likelihood function for the model we use to distinguish T from $S \setminus T$:

$$p(y_1, ...y_n \mid \theta_k, \theta_{-k}, k) = \text{Bin}(n_{k,T}, n_{k,S\setminus T} \mid n_{k,S}, \theta_k) \times \text{Bin}(n_{-k,T}, n_{-k,S\setminus T} \mid n_{-k,S}, \theta_{-k})$$

where θ_k and θ_{-k} are probability parameters with priors

$$\begin{aligned} \theta_k &\sim \text{Beta}(\alpha_T, \alpha_{S\setminus T}) \\ \theta_{-k} &\sim \text{Beta}(\alpha_T, \alpha_{S\setminus T}) \end{aligned}$$

with $\alpha_T = \alpha_{S\setminus T} = 1$ in our implementation. We want to then rank the keywords by how best they "fit" the actual distribution of documents into T and $S \setminus T$ by calculating their scores from the likelihood function. Since the probability parameters θ_k and θ_{-k} are not of interest, we marginalize over them to get:

$$\begin{aligned} p(y_1, \ldots, y_n \mid \alpha_T, \alpha_{S\setminus T}, k) \;\propto\; & \frac{\Gamma(n_{k,T} + \alpha_T)\Gamma(n_{k,S\setminus T} + \alpha_{S\setminus T})}{\Gamma(n_{k,T} + n_{k,S\setminus T} + \alpha_T + \alpha_{S\setminus T})} \times \\ & \frac{\Gamma(N_T - n_{k,T} + \alpha_T)\Gamma(N_{S\setminus T} - n_{k,S\setminus T} + \alpha_{S\setminus T})}{\Gamma(N_T - n_{k,T} + N_{S\setminus T} - n_{k,S\setminus T} + \alpha_T + \alpha_{S\setminus T})} \end{aligned}$$

We then calculate the value of the likelihood function for each keyword in each list, and rank them all from highest to lowest likelihood.

5.4 Human Input and Human-Computer Iteration

Our final step, prior to iterating, involves using human input to choose items from the two keyword lists and to build queries Q_T and Q_{RT}. Following Section 3, we optimize so humans do what they are good at and computerize what they are not. We present all the keywords, so the humans do not need to recall anything, along with computerized rankings to organize best guesses about what may be of interest to them. Then, the humans can use their detailed contextual knowledge, unavailable to our algorithm, to find different eddies of conversation, and meanings of concepts of interest not previously recalled. This process of evaluating a list of words is of course considerably faster and much more reliable than asking humans to pull keywords out of thin air or thinner memories.

The algorithm is unsupervised so that human users can easily refine, improve, or totally redefine the concept of interest, as the keyword lists inspire them to think of new perspectives on the same material. Users may also discover new directions that cause them to begin again with a completely new reference set, or to add to the existing reference set or reference query Q_R.

At this point, the user can iterate with the algorithm in various ways to continue to adjust the partition of S into T and $S \backslash T$ and to refine or redefine their concepts of interest. One way to iterate can be to simply update the reference query with the new selected words and rerun the algorithm. Another is for the user to designate specific keywords or documents of interest or not of interest, which gives algorithm more information to update the definitions of T and $S \setminus T$.

6 Evaluations

For our evaluations, we require a ground truth and a data set with documents properly coded to the concept of interest. Of course, the version of keyword selection we are studying is an unsupervised task, and so the concept initially chosen in real applications is not necessarily well-defined, may differ from user to user or application to application, and can be refined or changed altogether while using the algorithm; indeed, the ability of the user to make these changes is an important strength of the algorithm in practice.

Thus, to make ourselves vulnerable to being proven wrong, we evaluate distinct parts of the algorithm in specifically designed experiments. For example, we consider a limited case with a specific and fixed concept of interest. To do this, we leverage the usage of Twitter hastags as an explicit way users code their own concepts. The 4/15/2013 Boston Marathon Bombings example used in Section 3 was defined this way, with the hashtag *#bostonbombings*. We then construct a data set composed of three different sets of tweets. As the reference set, we use 5,909 English language tweets that contain the hashtag *#bostonbombings* but not the word *boston* posted April 15-18, 2013. The target set T we hope the algorithm will identify contains 4,291 tweets during the same time period that contains both *#bostonbombings* and *boston*. We created the $S \setminus T$ portion of the search set with the 9,892 tweets that were posted April 12-13, 2013, before the bombings, that contain the word *boston* but not *#bostonbombings*. The especially useful feature of these data is that the bombings were a surprise event that no one on social media was aware of ahead of time, which makes the demarcation between T and $S \setminus T$ much clearer than it would normally be.

The task of identifying the target set is of course straightforward with the keywords *#bostonbombings* and *boston*, and so solely for this experiment we remove them from the text of each of the tweets before our analysis. We also do not use metadata indicating the date or time of the tweet. This is therefore an artificial example, but one constructed to make it possible to evaluate. The goal is for human users selecting keywords with our algorithm to be more accurate, reliable, fast, and creative than working on their own without it. Although this is the relevant goal for a single human user, it is a trivially easy standard for our algorithm to meet. To see this, consider a crippled version of our algorithm with keyword lists ordered *randomly*. Since we showed in Section 3 that humans are usually incapable of recalling more than a small fraction of relevant keywords, but are very good at recognizing important keywords put before them, even randomly ordered keyword lists would still provide a great deal of help.

We thus seek to evaluate only the quantitative features of our algorithm here, and so run the algorithm once without iteration, and also without any human input or interaction. To simplify the analysis, and to make replication of our results easier with fewer computational resources, we degrade our approach further by using only two fast classifiers (Naive Bayes and Logit). The estimated target set is designated as any document that receives at least one classifier vote, with probability above 0.5. We also preprocess the documents in standard ways, by stemming, and removing punctuation, stop words, numbers, and words with fewer than three characters.

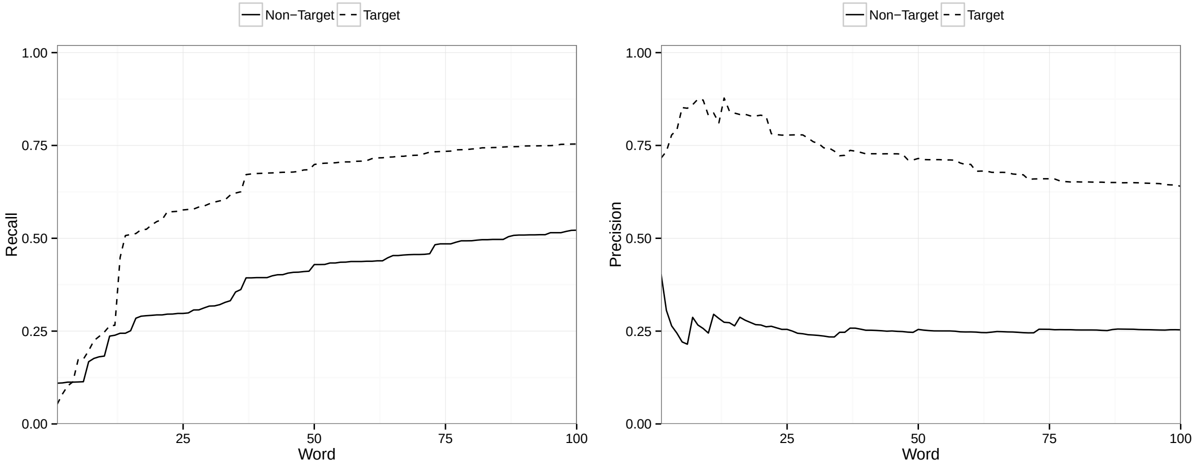
Qualitative Summary We evaluate this analysis in three ways, beginning in this section with the qualitative summary in Table 3. This table lists the top 25 (stemmed) keywords from the target T and non-target $S \setminus T$ keyword lists produced by a single run of the algorithm, without human input. We can evaluate the algorithm informally by merely looking at the words and seeing what readers recognize. It appears that most of the target keywords are closely related to the bombing incident (*#prayforboston*, *thought[s]*, *prayer*, *fbi*, *arrest*, *bomb*, *inure*, *attack*, *victim*, *terrorist*, etc.). A few words are clearly related but may be too imprecise to be useful to keywords to select documents (*cnn*, *sad*, etc.). Most non-target keywords do a good job of finding events related to Boston that are unrelated to the bombings, largely related to sports teams (*celtic*, *game*, *miami*, *heat*, *red sox*, *bruin*, *win*, *fan*, etc.). They also include a few words that were apparently misclassified and so should be in the target set (*tsarnaev*). The word *bostonmarathon* in the target set and *marathon* in the non-target set do not clearly discriminate posts related or unrelated to the bombings on their own to necessarily be useful — although interestingly the algorithm discovered a pattern difficult for humans, that social media posts happened to use the former word to describe the bombings and the latter to describe the sporting event.[5]

[5] Liam Payne was a 19-year-old singer inappropriately stopped by authorities in an underage establishment, and the subject of many social media posts. The word "rip" was, before removing punctuation and stemming, "R.I.P.," which means rest in peace.

Target Keywords	Non-target Keywords
peopl, thought, prayforboston, prayer, fbi,	marathon, celtic, game, miami, weekend
affect, arrest, cnn, pray, video,	heat, tsarnaev, new, play, red
obama, made, bomb, bostonmarathon, heart,	watertown, open, back, sox, job
injur, attack, releas, victim, terrorist,	mom, tonight, win, fan, monday
sad, news, sick, rip, investig	bruin, reaction, liam, tomorrow, payn

Table 3: Top 25 keywords in the Boston Bombings validation example from the target T and non-target $S \backslash T$ search set lists produced by a single non-iterative run of the algorithm, without human input.

Grouping and Ranking Keywords Second, we more formally evaluate the likelihood model used in our algorithm to group and rank keywords. Ideally, the target set list should have keywords that perform well on both recall and precision at the top and the non-target set list should have keywords that perform poorly on both recall and precision.[6] Figure 3 reports the cumulative recall and precision for the first 100 keywords in each list (introduced one at a time from left to right in both graphs). The cumulative recall (left graph) and precision (right graph) are running estimates as we add more and more terms into an "OR" Boolean query.



(a) Recall

(b) Precision

Figure 3: Cumulative Recall and Precision

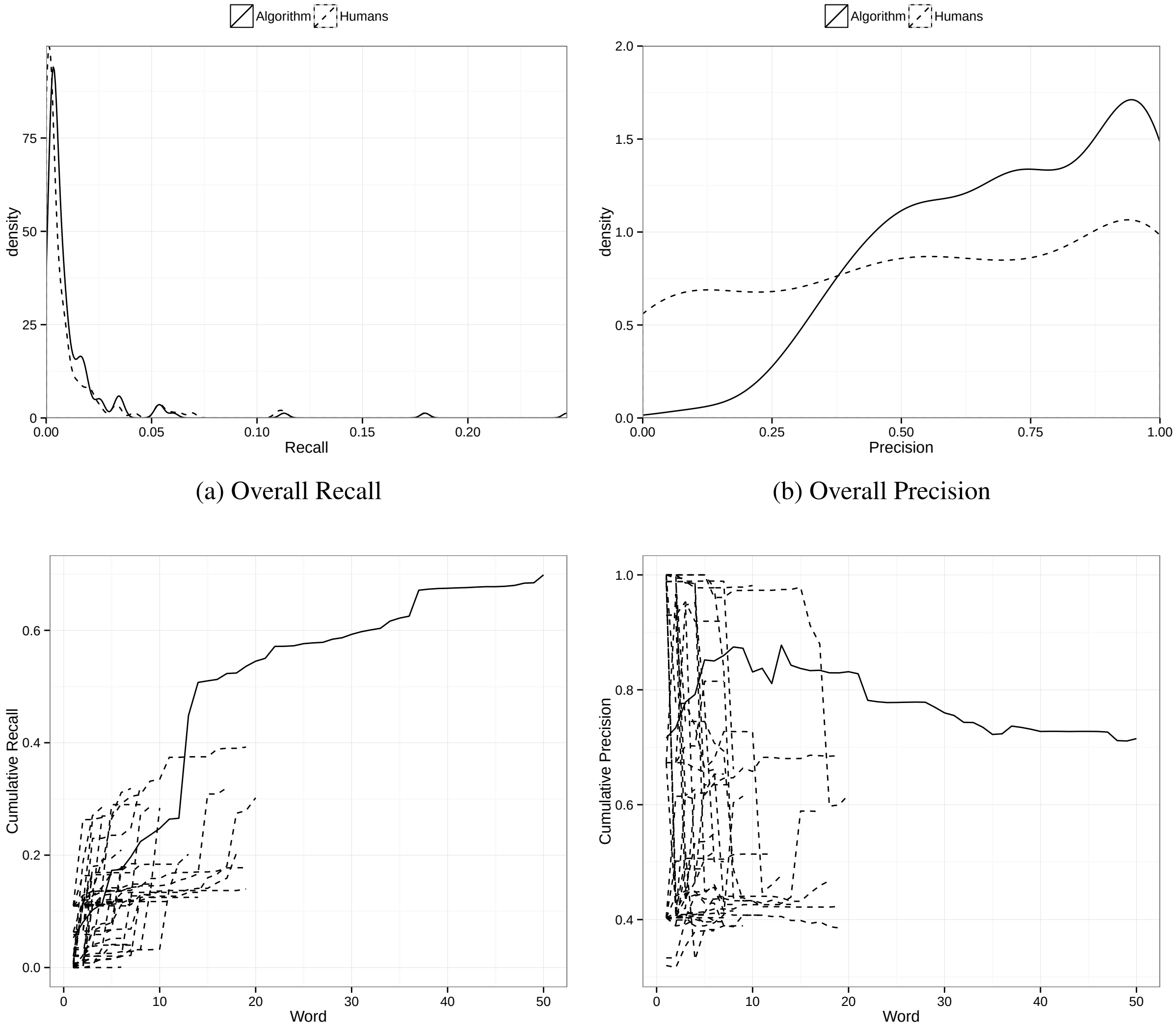
[6] The two common metrics we use are from the information retrieval literature. They include *precision*, the proportion of retrieved documents from each keyword that contains documents of interest, and *recall*, the proportion of all documents of interest that are retrieved by the keyword.

The key result in Figure 3 is that the target set line (in teal) is usually well above the non-target set line (in red) for both recall and precision. In other words, our algorithm is doing a good job separating the two lists, which provides quantitative confirmation of the qualitative impression from the words in Table 3.

By definition, cumulative recall increases as we add more keywords. The fact that recall is not consistently zero for the non-target set list speaks to both the need for human input, as well as the nature of human language where the same keywords can often be used in social media posts with the opposite meanings — describing concepts of interest and not of interest. The general downward trend of the cumulative precision for the target set list shows that the general ordering of the keywords is also valid, with more precise words near the top of the list.

Comparison to Human Users For our final evaluation, we compare this single non-iterative run of our algorithm (with no human in the loop) with a purely human approach. We do this in two ways. First, we compare the top 145 words from our target set keyword list with the 145 unique keywords that the 43 undergraduates in our Boston Bombings experiment came up with in the experiment described in Section 3. For this evaluation, we are therefore comparing the effort of 43 minds versus one single run of the computer algorithm without any human input. This is not a real comparison of course, since in practice researchers are unlikely to be able to hire 43 research assistants and would be able to use some human input to improve the algorithm, but it gives a useful baseline comparison.

Panels (a) and (b) in Figure 4 give density estimates for the overall precision and recall of the 145 words chosen by humans (in teal) compared to the top 145 words from the target set list from our algorithm (red). The results show that recall of the algorithm is approximately the same as the collective work of 43 humans. Put differently, both the one-step algorithm and the humans come up with keywords of about the same quality. Of course, we constrained the algorithm to the same number of words as the 43 humans when, of course, our algorithm would produce *many* more than the 145 words shown in the graph.



(a) Overall Recall

(b) Overall Precision

(c) Individual Recall

(d) Individual Precision

Figure 4: Comparing Recall and Precision for the Algorithm versus 43 Human Users. Panels (a) and (b): The distribution of recall and precision for the 145 words from the humans and the top 145 words from the algorithm. Panels (c) and (d): Cumulative recall and precision for each human (in black) versus first 50 target set keywords of algorithm (in red). Human keywords are in the order that humans thought of them.

To get a sense of the quality of the individual words in this comparison, we see from Panel (b) that the precision of the algorithm's words are generally much higher than the precision of words from the humans. When restricted to 145 words, the algorithm produces the same level of recall as the effort of 43 different humans combined, but the words chosen by the algorithm contain much less noise and are therefore of substantially higher quality than human only approaches.

Finally, we consider a more realistic comparison of a (still crippled) one-step version of our algorithm without human input to one human research assistant at a time. Individual humans choose only about 7-8 words, with no one of our 43 individuals choosing more than 20. Panels (c) and (d) of Figure 4 give cumulative recall and precision for our algorithm out to 50 words (although it could of course keep going) compared to each of our 43 human users. Individual undergraduate cumulative recall appears as separate black lines in Panel (c). The algorithm's cumulative recall is better than most of the human users until about 12 words are recalled, at which point the algorithm's performance soars well beyond any one of the human users. After 20 words, the human users obviously have nothing to offer. The algorithm's precision (panel d) is also better than most of the human users in the entire range of human recalled words, but then continues out to 50 words in the graph without losing much precision in the process.

Although our algorithm is clearly better than individual human users, using the algorithm with human input as designed has the potential to be much better than either alone.

7 Detecting the Language of Censorship Evasion

In what became known as the "Wang Lijun incident" in China, police chief of Chongqing Wang Lijun was abruptly demoted from his job on February 2, 2012. Rumors began circulating that Wang had fell out of favor of his boss, party chief of Chongqing and popular political leader Bo Xilai. On February 6, 2012, Wang Lijun went to the U.S. Consulate in Chengdu, possibly to seek asylum, but after the consulate became surrounded by police, Wang agreed to leave the consulate and was detained by the Chinese government. During this time, rumors about how the incident, perceived as treason by many in China, would affect the political prospects of Bo Xilai spread virally across social media, culminating in Bo's March 15 dismissal from his post. It was later revealed that Wang had fled to the consulate because he had confronted Bo that Bo and his wife, Gu Kailai, were connected to the murder of a British businessman Neil Heywood, who had died in November 2011 in Chongqing. In dramatic trials of Wang, Gu, and Bo that followed, all were convicted to lengthy prison sentences.

The Wang Lijun incident and Bo Xilai scandal were some of the most dramatic and important political events to occur in China in decades. Bo Xilai, son of a famous revolutionary Bo Yibo, had gained widespread popular support in Chongqing for his crackdown on crime and promotion of Maoist culture. He was also an ambitious politician who was hoping to be promoted to higher leadership roles within the Party. Because of the scale and drama involved in the scandal, the Bo Xilai scandal was of tremendous public interest and widely discussed, but at the same time highly censored.

Social media posts that used the names "Bo Xilai," "Gu Kailai," and "Wang Lijun" were censored across much of the social media landscape by automated filters programmed in many social media web sites. At the same time, social media users, who know about these filters, tried to write posts using creative re-phrasings and neologisms so their posts would slip past the filters but still be understandable to general readers. Amidst this linguistic arm's race between government-controlled computers and the Chinese people, researchers trying to understand this scandal have to scramble to keep up with these novel words and re-phrasings. Missing even one may cause them to lose the thread of the conversation, bias their inferences, or make finding posts of interest difficult or impossible. We show how our algorithm can be used by researchers to find these words and the posts of interest.

We began with words widely known to be used to evade censorship for the reference set and those that were more commonly used to describe the scandal in the search set. Examples of a few of the words we discovered appear in the first column of Table 4. For example, the reference set was composed of micro-blogs that contained the word *bxl* (in English), the first letter of each syllable in Bo's name, during the first half of 2012, and the search set was the broader term to describe the scandal “Chongqing incident” (重庆事件). The target set picked up a variety of words related to the event, including words that netizens were using to evade censorship. For example, 王丽娟, a homophone for Wang Lijun, appeared within the top one hundred of the list. Bu xing le (不行了, which means “not OK”, but has the same initials as Bo Xilai, appeared within the keyword list associated with the reference set BXL. Upon reading texts with these words, we verified that both of these words were being used to evade censorship.

Keyword discovered	Reference set	Search Set	Found In	Meaning
王丽娟	bxl	重庆事件	target set	homophone for Wang Lijun (王立军)
不行了	bxl	重庆事件	reference set	Bu Xingle, has same initials as Bo Xilai
护士长	王丽娟	重庆事件	reference set	“matron”, nickname for Wang Lijun
hwd	薄熙来	gkl	target set	Abbreviation for Neil Heywood’s last name

Table 4: Words the Chinese use to Evade Government Censors

Based on the new words we found to evade censorship, we further revised the reference set and reran the algorithm to search for other keywords. For example, we used the homophone for Wang Lijun, 王丽娟 , as the reference set and again “Chongqing incident” (重庆事件) as the search set. We discovered yet another nickname for Wang Lijun, “matron” (护士长). Using Bo’s full name 薄熙来 to define the reference set and the abbreviation for Gu Kailai’s name “gxl” as the search set, we also found the abbreviation for Neil Heywood’s name in the keyword target set “hwd”.

Of course not every word on the list was being using to evade censorship, since to be effective these words need to be rare. For example, many of the words were closely indicative of the scandal but not neologisms. However, a human user knowledgeable about the region can easily pick out the words that are being used to evade the censors from this longer list. Seeing the English word “hwd” out of a list of mostly Chinese characteris automatically alerts the reader or researcher that it is being used as shorthand for another word, and knowing the context (or perusing the documents) would enable one to ascertain if it is being used to substitute for a censored word. Similar patterns emerge in the purely Chinese words as well. The power here comes from the combination of the algorithm doing the “recalling” and the human doing the recognition of what is relevant.

8 Prior Literature

Our algorithm is related to the information retrieval literature and “query expansion” methods, including algorithms that add or reweight keywords within search queries to retrieve a more representative set of documents (Xu and Croft (1996), Rocchio (1971), or Carpineto and Romano (2012) for a review). Our approach differs in two important ways. First, most query expansion methods retrieve new keywords by stemming the original keyword, looking for synonyms or co-occurrences, or finding related terms within the corpus defined by the original keyword (Schütze and Pedersen (1997) Bai et al. (2005)). In contrast, our approach finds related keywords in external corpora that do not include the original keyword. For example, thesauri will not reveal novel hastags or many of the terms in log tail search (Section 2) or those used to evade censors (Section 7).

While some query expansion methods use large external corpora, such as Wikipedia, to enhance keyword retrieval (Weerkamp, Balog and de Rijke 2012), our method allows the user to define the external corpus without any structured data aside from the sets R and S. We thus rely on the user’s expertise to define the search and reference sets from which new, related keywords will be generated.

Second, current query expansion methods often try to limit “topic drift”, or are concerned with identifying keywords that are too general (Mitra, Singhal and Buckley 1998). As a result, most of those methods implicitly focus on maximizing the precision of the documents retrieved (making sure the documents retrieved are all relevant), whereas we focus on both precision and recall (making sure to retrieve as many of the relevant documents as possible). Our method intentionally suggests both general and specific keywords and includes topic drift, not as a problem to be fixed but, at times, as the subject of the study. We instead rely on the user interaction phase of our model to refine the keyword suggestions and avoid topic drift outside the user’s interest.

Finally, most query expansion methods rely on probabilistic models of the lexical properties of text (e.g. Voorhees (1994), Carpineto and Romano (2004)). Our approach uses ensembles of document classifiers to first group documents that may be of interest to the user. (A related approach is search results clustering (SRC), except with user-specifed corpora of documents; see Carpineto et al. (2009) for a review.) It then retrieves keywords that are likely to appear in this document group, but unlikely to appear in the rest of the search data set. Despite the differences between our approach and the current query expansion methods, our approach is actually a more general framework that can incorporate many of the existing methods, as we describe in a later section.

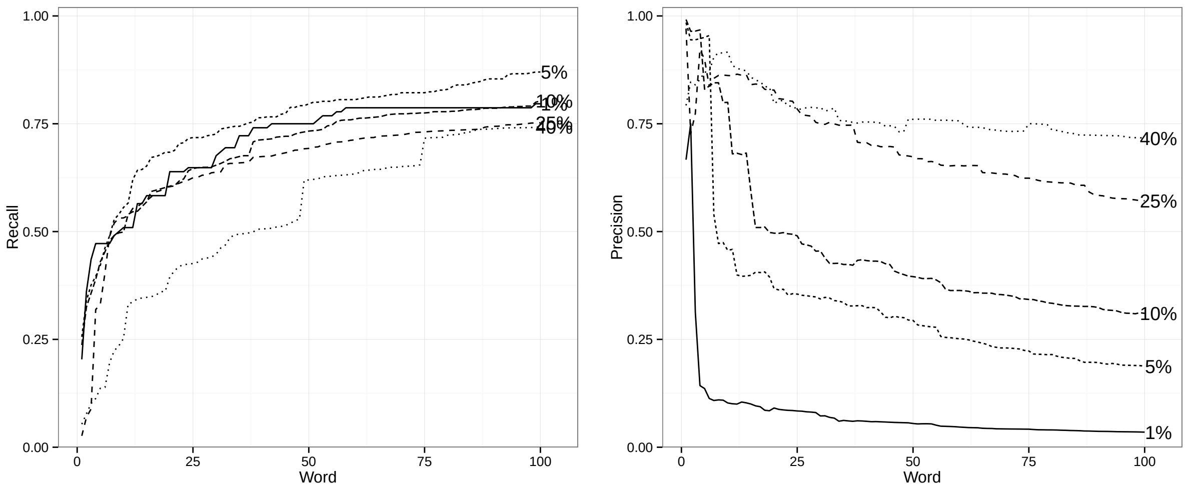
9 Concluding Remarks

The human led, computer-assisted, iterative algorithm we propose here learns from the mistakes made by automated classifiers, as well as the decisions of users in interacting with the system. In applications, it regularly produces lists of keywords that are intuitive, as well as those which would have been unlikely to have been thought of by a user working in isolation. Compared to a team of 43 human users, our algorithm has the same recall but far better precision; the algorithm also dominates individual human users on many dimensions. The algorithm discovers keywords, and associated document sets, by mining unstructured text, defined by the user, without requiring structured data. The resulting statistical framework and methods open up a range of applications for further analyses. In addition to the examples in English and Chinese, this algorithm has been useful in detecting Arabic dialects (Smith 2016), and we see no reason why it would not work on all human languages, but this would of course need to be studied further.

A Appendix: Robustness to Target Set Size

We study here how robust our keyword list discovery is as the target set size declines as a percentage of the search set. In Section 6, the (true) target set size was about 30% of the entire search set. We now test a variety of target set proportions from 1% to 40%. We create these samples by setting the search set size to 10,000, and then randomly drawing from the coded target and non-target sets to control the overall proportions.

Figure 5 gives cumulative recall and precision for different target set sizes. Clearly, the general trends from Figure 3 continue to hold. In addition, recall goes up and is higher for smaller target set proportions, which makes sense since fewer documents of interest need to be retrieved. Precision also follows the same downward sloping trend — higher for target sets that are a larger proportion of the search set. With more documents of interest and thus less noise in the search set, fewer high quality keywords exist, and more pertinent information can be found with each retrieval relative to noise. Note that for very small target sets, the precision drops off fast after only a few words, which suggests smaller target sets in general will have many fewer words that are of high quality. In practice, human users may choose to respond to this situation by broadening the concept of interest if that is an option, or, to find a needle in a large textual haystack, using small numbers of words to search document sets.



(a) Recall

(b) Precision

Figure 5: Cumulative Recall and Precision of Target Set Keywords for Different Target Set Percentages.

B Appendix: Building Queries for Large Data Sets

In Section 6, our validation example assumed a single data set that was divided into a reference and search set. The workflow in a single small data set is relatively simple. We first separate the reference set from the search set, run our algorithm, and then retrieve a list of target set keywords and non-target set keywords. The user can then use the keywords for various applications, one of which is building a comprehensive Boolean query Q_{RT} to retrieve a set of documents of interest. Q_{RT} can be built in this setup by simply taking the initial reference query Q_R and adding target set keywords with OR operators and/or non-target set keywords with the NOT and OR operators. For example, a query for the Boston Bombings example (assuming that the entire data set is given) could be "*#bostonbombings* OR (*suspect* OR *fbi* OR *#prayforboston*) AND NOT (*sox* OR *celtics* OR *bruins*)" where the words correspond to words from the reference query, target keyword list, and non-target keyword list respectively.

For large or potentially infinite data sources such as social media, the workflow described above is not feasible for a couple of reasons. In cases where the data set is large but finite, processing and running the algorithm over the entire data set as a search set may be infeasible computationally. For data sets of infinite size, there is no single search set that can be defined to run the algorithm. The user must define the search set manually via Boolean query or other means, a decision which then highly affects the results. We describe in more detail here a workflow alluded to in Section 5, where the definition of the search set may be incorporated into the workflow and the algorithm run multiple times to define the comprehensive query Q_{RT}. Consider the following steps to the workflow:

1. Define reference set R.

2. Mine R for keywords K_R to expand the query or use any other query expansion method available.

3. Choose one or more words from the query expansion to add to the query by either

    (a) adding the query expansion words to the comprehensive query Q_{RT} as is.

    (b) using the query expansion words to define a search set S, running our algorithm, and then adding additional words from our algorithm to refine the query expansion words for the comprehensive query Q_{RT}.

4. Repeat step 3 multiple times.

We demonstrate this workflow in an example of gathering relevant tweets about the Paris terrorist attacks on November 13, 2015. We collected a set of tweets between November 13 and November 15 with the hashtags *#parisattacks* as our initial reference set. We show here a very simplified version of the workflow for how to collect keywords and use our algorithm to develop a comprehensive Boolean query to retrieve tweets about the Paris attacks.

1. Use *#parisattacks* to define a reference set. (Q_{RT}: *#parisattacks*)

2. Use a simple query expansion method by simply mining the entire reference set for keywords and rank them according to their document frequency. Then scan the top 100 words for ideas about expanding the query. We can alternatively include any other query expansion method in the literature here.

3. See the word *#prayforparis* in the expansion list. Through substantive knowledge, recognize that all tweets returned by *#prayforparis* are like to be relevant so simply add it to the query. (Q_{RT}: *#parisattacks* OR *#prayforparis*)

4. See the word *paris* in the expansion list. We would like to add it to the query but not all documents retrieved by *paris* will be relevant, so we need to use the algorithm to subset further. Define and retrieve a search set with *paris* but excluding *#parisattacks* or *#prayforparis*.

5. Run the algorithm on the newly defined search set and look at the top 100 words in each list. See words that will help retrieve relevant posts from target set list (e.g. *prayer*, *raid*, *abaaoud*, *mastermind*) and words that indicate non-relevant posts from non-target set list (e.g. *climate*, *change*, *conference*). Add to comprehensive query. (Q_{RT}: *#parisattacks* OR *#prayforparis* OR (*paris* AND (*prayer* OR *raid* OR *abaaoud* OR *mastermind*) AND NOT (*climate* OR *change* OR *conference*)))

6. From the expansion list in step 2, see the word *france* and use it as a search set for investigation. Repeat the algorithm with the new search set and find words that separate *france* into relevant and irrelevant posts. (Q_{RT}: *#parisattacks* OR

#prayforparis OR (*paris* AND (*prayer* OR *raid* OR *abaaoud* OR *mastermind*) AND NOT (*climate* OR *change* OR *conference*)) OR (*france* AND (*suspect* OR *victim* OR *attack* OR *terrorist*) AND NOT (*air* OR *england* OR *russia* OR *benzema*))

7. Repeat until satisfied.

Through this workflow that involves both human and algorithmic expertise, the user can work through a large or infinite set of documents and retrieve the relevant documents of interest by building long and comprehensive queries.

References

Antoun, Christopher, Chan Zhang, Frederick G Conrad and Michael F Schober. 2015. "Comparisons of Online Recruitment Strategies for Convenience Samples Craigslist, Google AdWords, Facebook, and Amazon Mechanical Turk." *Field Methods* pp. 1–16.

Bai, Jing, Dawei Song, Peter Bruza, Jian-Yun Nie and Guihong Cao. 2005. Query Expansion Using Term Relationships in Language Models for Information Retrieval. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*. pp. 688–695.

Bauml, Karl-Heinz. 2008. Inhibitory Processes. In *Learning and memory: A comprehensive reference, Volume 2: Cognitive psychology of memory*, ed. Roediger H.L. Oxford: Elsevier pp. 195–220.

Bishop, Christopher M. 1995. *Neural Networks for Pattern Recognition*. Oxford: Oxford University Press.

Carpineto, Claudio and Giovanni Romano. 2004. "Exploiting the Potential of Concept Lattices for Information Retrieval with CREDO." *Journal of Universal Computer Science* 10(8):985–1013.

Carpineto, Claudio and Giovanni Romano. 2012. "A Survey of Automatic Query Expansion in Information Retrieval." *ACM Computing Surveys (CSUR)* 44(1):1.

Carpineto, Claudio, Stanislaw Osiński, Giovanni Romano and Dawid Weiss. 2009. "A Survey of Web Clustering Engines." *ACM Computing Surveys (CSUR)* 41(3):17.

Chen, Yifan, Gui-Rong Xue and Yong Yu. 2008. Advertising Keyword Suggestion Based on Concept Hierarchy. In *Proceedings of the International Conference on Web Search and Web Data Mining*. pp. 251–260.

D'Orazio, Vito, Steven T Landis, Glenn Palmer and Philip Schrodt. 2014. "Separating the Wheat from the Chaff: Applications of Automated Document Classification Using Support Vector Machines." *Political analysis* 22(2):224–242.

Eshbaugh-Soha, Matthew. 2010. "The Tone of Local Presidential News Coverage." *Political Communication* 27(2):121–140.

Gentzkow, Matthew and Jesse M Shapiro. 2010. "What Drives Media Slant? Evidence from US Daily Newspapers." *Econometrica* 78(1):35–71.

Grimmer, Justin and Gary King. 2011. "General Purpose Computer-Assisted Clustering and Conceptualization." *Proceedings of the National Academy of Sciences* 108(7):2643–2650. http://gking.harvard.edu/files/abs/discov-abs.shtml.

Hand, David J. 2006. "Classifier Technology and the Illusion of Progress." *Statistical Science* 21(1):1–14.

Hastie, Trevor, Robert Tibshirani and Jerome Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Ed.* New York: Springer.

Hayes, Philip J and Steven P Weinstein. 1990. CONSTRUE/TIS: A System for Content-Based Indexing of a Database of News Stories. In *IAAI*. Vol. 90 pp. 49–64.

Ho, Daniel E and Kevin M Quinn. 2008. "Measuring Explicit Political Positions of Media." *Quarterly Journal of Political Science* 3(4):353–377.

Hopkins, Daniel and Gary King. 2010. "A Method of Automated Nonparametric Content Analysis for Social Science." *American Journal of Political Science* 54(1):229–247. http://gking.harvard.edu/files/abs/words-abs.shtml.

King, Gary, Jennifer Pan and Margaret E. Roberts. 2013. "How Censorship in China Allows Government Criticism but Silences Collective Expression." *American Political Science Review* 107:1–18. http://j.mp/LdVXqN.

Kulkarni, Sanjeev R, Gábor Lugosi and Santosh S. Venkatesh. 1998. "Learning Pattern Classification-A Survey." *IEEE Transactions on Information Theory* 44(6):2178–2206.

Letham, Benjamin, Cynthia Rudin, Tyler H McCormick and David Madigan. 2013. "Interpretable Classifiers Using Rules and Bayesian Analysis: Building a Better Stroke Prediction Model.".

Letham, Benjamin, Cynthia Rudin, Tyler H McCormick, David Madigan et al. 2015. "Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model." *The Annals of Applied Statistics* 9(3):1350–1371.

Mitra, Mandar, Amit Singhal and Chris Buckley. 1998. Improving Automatic Query Expansion. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 206–214.

Nielsen, Finn Årup. 2011. "A new ANEW: Evaluation of a word list for sentiment analysis in microblogs." *CoRR* abs/1103.2903.
URL: *http://arxiv.org/abs/1103.2903*

Puglisi, Riccardo and James M Snyder. 2011. "Newspaper Coverage of Political Scandals." *The Journal of Politics* 73(3):931–950.

Rocchio, Joseph John. 1971. *Relevance Feedback in Information Retrieval*. Prentice-Hall, Englewood Cliffs NJ.

Roediger, Henry L and James H Neely. 1982. "Retrieval Blocks in Episodic and Semantic Memory." *Canadian Journal of Psychology/Revue canadienne de psychologie* 36(2):213–242.

Schapire, Robert E and Yoav Freund. 2012. *Boosting: Foundations and Algorithms*. The MIT Press.

Schütze, Hinrich and Jan O Pedersen. 1997. "A Cooccurrence-Based Thesaurus and Two Applications to Information Retrieval." *Information Processing & Management* 33(3):307–318.

Smith, Evann. 2016. Mass Mobilization in the Middle East: Form Perception, and Language PhD thesis Harvard University Cambridge MA: .

Voorhees, Ellen M. 1994. Query Expansion Using Lexical-Semantic Relations. In *Pro-*

ceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.

Weerkamp, Wouter, Krisztian Balog and Maarten de Rijke. 2012. "Exploiting External Collections for Query Expansion." *ACM Transactions on the Web (TWEB)* 6(4):18.

Xu, Jinxi and W Bruce Croft. 1996. Query Expansion Using Local and Global Document Analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 4–11.

Yang, Guobin. 2009. *The Power of the Internet in China: Citizen Activism Online*. New York: Columbia University Press.