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

Gary King,† Patrick Lam‡ Margaret E. Roberts§

July 15, 2014

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 crucially on this choice, researchers typically pick keywords in ad hoc ways, given the lack of formal statistical methods to help. Paradoxically, this often means that the validity of the most sophisticated text analysis methods depends in practice on the inadequate keyword counting or matching methods they are designed to replace. The same ad hoc keyword selection process is also used 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 any 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 then summarizing results with Boolean search strings. We illustrate how the technique works with examples in English and Chinese.

*Our thanks to Dan Gilbert for helpful suggestions.

†Institute for Quantitative Social Science, 1737 Cambridge Street, Harvard University, Cambridge MA 02138; GKing.harvard.edu, king@harvard.edu, (617) 500-7570.

‡Institute for Quantitative Social Science, 1737 Cambridge Street, Harvard University, Cambridge, MA 02138; www.patricklam.org

§Department of Government, 1737 Cambridge Street, Harvard University, Cambridge MA 02138; scholar.harvard.edu/mroberts

1 Introduction

Boolean keyword search in textual documents is an extremely generic task used at some level by most methods of automated text analysis and in numerous areas of substantive application. In many areas, algorithms for assisting keyword selection are not available (beyond thesauri and other simple approaches). This situation is problematic since choosing all words that represent a specific concept is well known to be a "near-impossible task" for a human being (Hayes and Weinstein 1990). Indeed, many sophisticated methods of automated text analysis were designed to get around simplistic and demonstrably inadequate keyword matching and counting methods but yet turn out to require their use and assume their validity even to get started.

In some areas of application, sets of external structured data exist for mining keywords beyond the 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 them. In this paper, we develop an alternative framework for keyword suggestion that uses a much wider range of information in unstructured textual data, can be used on the fly with the human in the loop, and requires no structured data. Posing the statistical problem in the way we do suggests new methodological approaches and new areas of application, and may have the potential to contribute useful keywords and document set definitions beyond those from existing approaches when structured data are available.

We begin by reframing the statistical problem, listing some of the application areas to which our methodology may provide some assistance, introducing our specific methodology, and then offer a range of illustrations and validations.

2 Reframing the Statistical Problem

Define a *reference set*, R, as a set of documents, each of which contains text that includes at least one match for a set of chosen keywords (and Boolean operators) K_R. The reference set is intended to be defined narrowly so that it only includes documents about

a single chosen concept (topic, sentiment, etc.), such that the probability of documents being included that do not represent this concept are negligible with respect to the application at hand. Also define a *search set*, S, as another set of documents that does not overlap the reference set (i.e., none of which contain text with matches for K_R), and some of which may be about the concept. The search set may be defined by a broader set of keywords (K_S), a convenience sample, all available documents, or the entire Internet.

The goal of the procedure is to identify a *target set* T, a subset of the search set ($T \in S$) with documents that reflect the same concept defining the documents in the reference set, or equivalently to find a set of keywords (and Boolean operators) K_T that define the target set. The goal, depending on the application, may include keywords K_T, the target set T, the target and reference sets $\{T, R\}$, or all of the above. The algorithm we propose below analyzes R, S, and K_R and outputs K_T, along with the associated target set.

To identify the target set and associated keywords, we develop a computer-assisted (human empowered) algorithm, rather than a fully automated algorithm. The reason for this choice is that the "concept" which all 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. Indeed, 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 a reference set is defined by 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? 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.

3 Application areas

Algorithms that meet the requirements of the statistical problem as framed in Section 2 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; others represent major 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 also 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 have similarly tried 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, as they can be highly correlated with the opinions being studied. There is a need for a method that extracts these documents sets and defines keywords directly from text as written, rather than waiting for many people to figure out and search on these words so that the information may be harvested from weblogs and other search data.

Evading the Censors Censorship of the Internet exists in almost all countries to some degree. Governments, and social media and other firms that operate within their jurisdictions, use techniques such as keyword 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. Each of these practices is a special case of censorship.

In response, netizens who hope to continue discussion about such censored topics respond with alternative phrasings in a continuing effort to evade censorship. For example, immediately after the Chinese government arrested artist-dissident Ai Weiwei, the government 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". In this application, we can define the reference set as social media posts containing the term "Ai Weiwei" and the search set as all social media posts, or those from a specific region or about some of the topics at issue. Our algorithm then enables researchers to identify these alternative keywords, and the associated social media posts on the same topic. In this way, we make it possible to continue to follow the conversation even in the presence of creative authors.

Product Advertisements Those purchasing online advertising often bid to buy ad space next to searches for chosen keywords. For example, to sell a vacuum cleaner, one might bid for search keywords "carpet", "vacuum", "Hoover", etc. This is common with Google adwords, Bing Ads, 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 from raw unstructured text found in company documents, customer call logs, customer 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.

Legal Discovery Electronic discovery, or "e-discovery" has become a central part of a broad array of civil litigation. Once a lawsuit is filed, and standing is granted, attorneys

on both sides must agree on the documents at issue. Often the plaintiffs try to expand the document set, while defendants try to narrow the list. In the "culling" phase of document review, an initial set of keywords is identified. Then a sample of documents is manually reviewed, and each document is determined to be relevant or irrelevant. Then the set of keywords is narrowed or expanded depending on how many documents within the sample are relevant to the topic. A more immediately relevant set of keywords would not only assist in fair judgements of these cases, but also in cutting unnecessary costs in reviewing excessive numbers of documents.

Searching for Sets of Documents The way we reframe the statistical problem above suggests a common but under-appreciated distinction between two uses of modern search engines. In the first, users seek one or a small number of the most relevant web sites or documents, a use case which we might call *fact finding*. For example, to find the capital of Montana, the user only wants one site (or a small number of sites) returned. To assist with this first use case, search engine manufacturers spend a great deal of effort ranking web sites to reflect user intent. In the second *collecting* use case, users do not try to find the needle in the haystack; instead, they are try to identify all sites or documents that discuss a particular topic, product, business, person, or concept. The attempt here is an exhaustive list rather than to find the single most appropriate site. Some of the many examples of collecting include scholarly literature reviews, research of many types, vanity searches, brand monitoring, and company due diligence.

Search engines are optimized for fact finding, but are nevertheless regularly used for collecting, even though they are suboptimal for this alternative purpose. This suggests that they could be substantially improved by developing methods, such as those offered here, explicitly for identifying a document set (and associated keywords) relevant to a chosen topic. Some algorithms have been proposed and implemented on search engines to provide some assistance for collecting, but the methods based on fully automated cluster analysis work very poorly on most problems, for known theoretical reasons (Grimmer and King 2011).

To apply our algorithm, the user could define a reference set by choosing a narrowly

defined list of keywords (and verifying that the documents returned largely include those of interest). Then a very broad set of keywords could be used to define the search set, and the algorithm would then suggest relevant other keywords to help the user identify other portions of the documents that were excluded from the reference set.

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. These areas 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) 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.

Take social search, for example. The user could search *#BostonBombings* to produce a flow of social media posts. These would almost all be about the Boston Bombings, but numerous others, such as *#prayforboston* or those about the subject without a hashtag, would be missed. To use our algorithm, we would merely define the results of this first search as the reference set, and use it suggest new keyword sets and thus improve the set of posts displayed. The quality of the resulting search may thus be greatly improved. A similar situation would apply to enterprise search.

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, and then spend most of their effort on applying sophisticated statistical, machine learning, linguistic, or data analytic methods to this given corpus. In practice, this document set is defined in one of a variety of ways but keyword searching is a common approach. See for example 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.

Intuitive and Infinitely Improvable Classification Because statistical classifiers typically fail to achieve high levels of accuracy (Hand 2006), analytically unsophisticated users finding individual documents misclassified may question the veracity of the whole approach, and have little recourse to improve the result. Moreover, since classifiers usually optimize a global function for the entire data set, even sophisticated users may find of value a way of adding effort to improve classification at the level of smaller numbers of documents.

In this situation, a keyword-based classifier is sometimes more useful because the reasons for mistakes, even if there are more of them, are readily understandable. 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 keyword list. Our algorithm, which uses statistical classifiers to find a keyword-based classifier for the user, may thus offer a useful alternative approach in this situation.

4 The Unreliability of Keyword Selection by Unaided Humans

Human beings, unaided by computers, often think they are good at selecting keywords (think of yourself using the Google search engine). However, as we demonstrate in this section for the slightly more complicated task of choosing a set of keywords to select a target set from a search set, even expert human users are highly unreliable. That is, two human users familiar with the subject area, given the same task, will usually select keyword lists that overlap very little, and each will be a very small subset of those they would recognize as useful after the fact. And, of course, without reliability, validity is not even defined.

This pattern in fact is to be expected given a well documented but counterintuitive result from psychological research on "inhibitory processes" (and in particular "part-list cuing"). Many psychological experiments have been devoted to attempting to recall a set

of words about some topic (e.g., words that are easy to recognize as belonging to the list if revealed). In this situation, revealing one word facilitates remembering others, but the cue provided by revealing more than one word in the set strongly *inhibits* recall of the rest of the set (Roediger and Neely 1982, Bauml 2008). Applied to our problem, what appears to happen is that a person can often easily come up with a small number of keywords about some aspect of an event. However, the cues provided by the first few words recalled degrades the person's ability to retrieve other similar keywords, as well as those related to different aspects of the same concept.

The difficulty of recalling keywords by hand is exacerbated by the fact that a human user may not even be aware of many of the keywords that could be used to classify search documents into a target set. Users are also unlikely to be able to find many keywords within a reasonable amount of time by directly examining or reading any subset of the large numbers of documents in the search or reference sets.

With a small formal experiment, we now demonstrate that the same unreliability also applies to the task of keyword selection. To do this, we asked 43 sophisticated individuals (mostly undergraduate students) to enter keywords in a web form with this prompt: "We have 10,000 twitter posts, each containing the word 'health care', from the time period surrounding the 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 them access to a sample of the posts and asked them not to consult other sources. We repeated the example with four other substantive examples (some of which are detailed in Section 6 and Section 7).

The median number of words selected by our respondents were 8 for the Obamacare example and 7 for an example about the Boston 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. The position of any one word within the cloud is arbitrary.

The results clearly demonstrate the high level of unreliability of human keyword selectors: 66% of the unique words selected in the Obamacare example, and 59% selected

(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 printed in red. The position of each word within the cloud is arbitrary.

in the Boston bombing example, were selected by only one of the 43 respondents (see the mass of words colored in red in both panels). We also ran the same experiment with three other examples (regarding Nelson Mandela, the Nigerian schoolgirls kidnapping, and the birth of Kate Middleton's child) and found, respectively, that 66%, 76%, and 63% of the unique keywords selected were chosen by only one respondent. Across all five examples, essentially no two respondents chose the same keyword list.

5 Algorithm

The overall idea of the algorithm is to marshal diverse methods to first classify documents in the search set into either the search or reference sets. We then learn from the mistakes of these classifiers. However, instead of correcting the mistakes, we exploit the information that leads to the mistakes to identify the target set.

We first outline the specific steps required for our proposed algorithm and then provide additional detail on two of the steps. Table 1 gives a very brief summary as a guide.

1. Define a reference set R and search set S.
2. Define a training set by sampling from R and S.
3. Fit multiple diverse classifiers to the training set, predicting whether each document should be in R or S.
4. Use parameters from classifiers fit to the training set to estimate predicted probabilities of R membership for all documents in S. Combine probabilities when running parallel processes.
5. Summarize documents by the vector of opinions about each document and, on that basis, partition documents into similar clusters
6. Find keywords that best classify documents into clusters
   (a) Generate a list of potential keyword combinations/rules r from the search set S using the Apriori algorithm or similar.
   (b) For each of the set of rules, calculate the likelihood for each cluster.
   (c) Return the set of rules within each cluster ranked by the highest likelihood to the user.
7. Present keywords in clusters or lists to the user who can resolve linguistic ambiguities.
8. Update keyword lists (by redefining R and repeating steps 1–6) each time the user chooses keywords.

Table 1: *The Keyword Algorithm:* This is the simplest implementation of our algorithm, and the version we use in the illustrations below. The algorithm also has numerous possible extensions, such as generating combinations of rules, expanding clusters, and iterating between user input and the algorithm.

5.1 The Eight Steps

Our algorithm requires eight steps. First, following Section 2, we define the reference set R, the associated keywords and Boolean operators K_R, and the search set S. The search set may be defined as a preexisting set (say all web sites, or all press releases) or by subsetting via a set of keywords (K_S) that culls documents from a larger set.[1]

[1] Prior to our algorithm, users may optionally preprocess textual documents in ways that are common in the literature. This can include changing all letters to lowercase, removing punctuation, removing numbers or special characters, removing stop words, stemming words, or removing short words that consist of fewer than three letters.

Second, we define a "training" set from subsets of S and R. The subsets will usually be chosen at random, although for some applications we may use "exemplars" that best represent documents within the chosen set. We also sometimes repeat this step with a different random subsettings to increase the diversity of keyword candidates.

Third, we fit a large number of document classifiers to the training data trying to predict membership in S vs. R, using as predictors any element of the text of the documents (other than their deterministic definitions, K_R and K_S). Any set of statistical, machine learning, or data analytic classifiers can be used, but we recommend as large and diverse a set as is convenient and computationally feasible (e.g., Bishop 1995, Hastie, Tibshirani and Friedman 2009, Kulkarni, Lugosi and Venkatesh 1998, Schapire and Freund 2012).

Fourth, we use the classifier estimated in the training set to estimate the probability of each document in the search set falling into either R or 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 by doing this now. Thus, Table 2 portrays the results for one example classifier, with the originally defined truth in rows and potential classifier decisions in columns. For the applications areas given above, the researcher 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 have some similarities to the reference set (even though they are in fact in the search set), and so likely contain new keywords we can harvest to better represent the concept of interest.[2] For the rest of the algorithm, we use the continuous probability of being in each of the classes in Table 2, such as $p_{c,d}(R|S)$, for classifier c and document d.[3]

[2]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. Similarly, the documents $\{R|R\}$ contain alternative keywords that define documents within the reference group. These could be used in the next iteration as keywords to refine the definition of the search dataset, or might be specific enough to be used to expand the reference set directly.

[3]To speed up the algorithm, we approximate it and parallelize the classification step. After defining S and R, we randomly sample a small number of documents from S and R respectively. We repeat this sampling many times in parallel. We then follow steps 2-4 for each parallel process by designating a training set from the S and R samples and running the classifiers to get $p_{c,d}(R|S)$ for each S document in

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

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.

Fifth, we take advantage of the fact that different classifiers pick up on different features of the data and so can be used to represent different concepts, some of which a user may wish to choose. In this step, we group documents based on similarity with respect to the set of potentially diverse opinions expressed by the set of classifiers about the probability of (for example) a search document being in the reference set. We give more detail of this step in Section 5.2,

Sixth, we extract keywords (or more general Boolean operators) that characterize (and would help distinguish) each of the clusters we created of search set documents. Keywords provide a simple and concise definition of candidate target sets. Although humans are not good at selecting them all on their own, individual keywords are easily understandable and the set of keywords are easily improvable by human users. Section 5.3 provides more detail on this step.

Seventh, we present keywords to users organized in sets by similarity and likelihood of their being useful; this step is illustrated in context in complete examples in Section 6.

Finally, we keep the human in the loop by treating keywords surfaced thus far as suggestions, which the user can keep or reject by studying the keywords or some of the documents they select. If a keyword is deemed relevant, it can be used to expand the reference set; concluding that a keyword is not relevant can be used to shrink the search set. As the search and reference sets are updated with knowledge from the user, we rerun steps 1-7 (automatically, in the background) which continuously generates new and improved sets of keywords to offer the user. After interacting with the system in this

each parallel run. Across M parallel runs, any particular document in S may be sampled between 0 and M times. For each document in S sampled at least once in the parallel runs, we then calculate C averaged probabilities, one for each classifier, by taking the average probability $p_{c,d}(R|S)$ over the parallel runs, where each averaged probability is an average across anywhere from 1 to M values depending on how often an S document is sampled in the parallel runs. We then follow the remaining steps from the algorithm using the average probabilities.

iterative way, the user may come to a decision that effectively all documents will be found, given the time available, and thus no more keyword searching is necessary.

5.2 Step 5 Details: Leveraging Multiple Classifiers

Suppose one of the keywords defining the reference set K_R is "black." Does this mean the user interested in a color, an ethnic group, a comedian (Lewis Black), a way to prepare coffee, or a great ski run? To answer this question, we will use human input; our task now is to find a way to represent this diversity of information in a manner that may make it easier for the user.

Denote by $p_{c,d}(R|S)$ the probability that document d from the search set is estimated to be in the reference set, according to classifier c. If we were to assume that all the classifiers tapped the same underlying concept defining the reference set, then we might average $p_{c,d}(R|S)$ over classifiers for each document (i.e., assuming that they differ only due to random chance) and then mine keywords from the set of documents with the highest probabilities. Instead, we use the diverse "opinions" about what documents should be in the reference set offered by a diverse set of classifiers to help characterize the different concepts that the same keywords defining the reference set may represent.

Thus, we first summarize each document d with the set of classifier opinions about it:

$$p_{.,d}(R|S) = \{p_{1,d}(R|S), \ldots, p_{C,d}(R|S)\}$$

Then, we use a clustering method to partition the documents, with each partition intending to approximate a different concept or meaning of K_R. In principle, any number of existing clustering methods may be used at this stage. One approach we use is the partition around medoids algorithm ("pam") and its large data sampling counterpart CLARA (Clustering LARge Applications) (Kaufman and Rousseeuw 1990), but others are feasible at this stage too.

5.3 Step 6 Details: Extracting Keywords

Each of the clusters of documents within the search set created in Step 5 serves as a candidate for the target set $\{R|S\}$. At this point our algorithm would benefit from human

input, which requires that it be presented in an easily and quickly understandable manner. In fact, even if one of these clusters clearly belongs in the target set, the user would have a difficult time ascertaining this fact by having access to only a large set of documents too numerous to read.

Thus, we summarize each cluster by a simple set of keywords so that it easier to understand (and of course for some purposes, the goal is the keywords themselves). To do this, we use a method to optimally classify documents into these given clusters subject to the constraint of keyword-based classification. Even if the keywords do not perfectly represent the cluster, they will often be close and, regardless, being able to receive human input at this stage is often well worth the compromise. Moreover, the fact that the keywords are understandable by the human user means, unlike most statistical or machine learning classifiers, that the user can tweak the keywords to continuously improve the classification if desired.

To construct a keyword-based classfier, we borrow from the literatures on decision rules and association rule learning coupled with a generative model (Apte, Damerau and Weiss 1993, Cohen 1996, Cohen and Singer 1999). Our specific approach is similar to that of Letham et al. (2013), based on Bayesian List Machines (BLM).

For expository clarity, we reuse some terminology and designate one cluster of interest as $\{R|S\}$ and denote the union of all documents in all other clusters as the set $\{S|S\}$. (We will ultimately repeat this for each cluster of interest.) Then, for document d in search set S, let y_d equal 1 if $d \in \{R|S\}$ and 0 if $d \in \{S|S\}$. The goal is to perform a discrete optimization, identifying the set of keywords that best distinguish documents in $\{R|S\}$ from documents in $\{S|S\}$.

Then denote a keyword decision rule r (or Boolean search string) in the format

$$\text{foo} \Longrightarrow \{R|S\}$$

where "foo" is a chosen keyword and the notation indicates that "foo" selects the set $\{R|S\}$ because documents in this set contain the keyword "foo" with higher frequency than documents in the set $\{S|S\}$. Decision rules can also be much more complicated than a single keyword, including any Boolean search string. For simplicity, we first consider

three types of decision rules.

1. **single keywords**: "foo" $\Longrightarrow \{R|S\}$
2. **AND statements**: "foo" & "bar" $\Longrightarrow \{R|S\}$
3. **NOT statements**: NOT "foobar" $\Longrightarrow \{R|S\}$

Single keywords are words (or word stems) prevalent in $\{R|S\}$ but not $\{S|S\}$. "AND statements" are combinations of single keywords prevalent in $\{R|S\}$ but not $\{S|S\}$. Note that the traditional n-grams are simply subsets of AND statements that identify the string of words together. "NOT statements" are single (or multiple if combined with AND statements) keywords that are prevalent in $\{S|S\}$ but not $\{R|S\}$.

If we had infinite time or infinitely fast and powerful computers, we would list every possible Boolean search string, quantify how well each does in the task of discriminating $\{R|S\}$ from $\{S|S\}$, and then choosing the best one. Thus, to generate a set of decision rules for $\{R|S\}$, we woud need a method to generate a list of possible keyword combinations on the left-hand side of the decision rule and a metric to rank the performance of the rules in characterizing the $\{R|S\}$.

Of course, enumerating all empirical keyword combinations is infeasible. Fortunately, we only need to identify keyword combinations that occur frequently either within $\{R|S\}$ or $\{S|S\}$, which we can do using ideas in the association rule learning literature. (The association rule learning algorithms were not originally intended for use with keywords and text mining. In that literature, our document sets are known as the "transactions" and the combinations of keywords play the role of "itemsets".) For example, these properties can be found in the "Apriori algorithm" (Agrawal and Srikant 1994) for frequent item set mining, although several others are available as well. This algorithm allows one to reasonably restrict the number of keyword combinations by (1) increasing the minimum support for keyword combinations within the search set, (2) mining for possible keyword combinations only within given clusters, and (3) setting limits on the number of words (AND statements) allowed in any keyword combination.

Once we have a set of frequently used keyword combinations, we need a metric to

rank them by how well they characterize $\{R|S\}$ relative to $\{S|S\}$. Here we assume the following data generating process. For simplicity, we first drop the right-hand side of the rule and let r denote any frequent keyword combination within our frequent item set. Then the vector y is drawn from the Beta-Binomial distribution with Beta parameters α_{SS} and α_{RS}, Binomial parameters θ^r and θ^{-r}, and counts $n_{r,RS}$, $n_{r,SS}$, $n_{-r,RS}$, and $n_{-r,SS}$.

To calculate the likelihood, define $n_{r,RS}$ as the number of documents in $\{R|S\}$ that satisfy rule r (i.e., documents which include the keyword combinations in r) and $n_{r,SS}$ as the number of documents in $\{S|S\}$ that satisfy rule r. Also let $n_r = n_{r,RS} + n_{r,SS}$ be the total number of documents in S that satisfy rule r. The notation $-r$ denotes the analogous quantities for documents that do not satisfy rule r. Finally, define N_{RS} and N_{SS} to be the total number of documents in each group such that $N_{RS} = n_{r,RS} + n_{-r,RS}$ and $N_{SS} = n_{r,SS} + n_{-r,SS}$. Then the likelihood for this model is:

$$L(y_1, ...y_n|\theta^r, \theta^{-r}, r) = \text{Bin}(n_{r,RS}, n_{r,SS}|n_r, \theta^r) \times \text{Bin}(n_{-r,RS}, n_{-r,SS}|n_{-r}, \theta^{-r})$$

where

$$\begin{aligned} \theta^r &\sim \text{Beta}(\alpha_{RS}, \alpha_{SS}) \\ \theta^{-r} &\sim \text{Beta}(\alpha_{RS}, \alpha_{SS}) \end{aligned}$$

In our examples, we set $\alpha_{RS} = \alpha_{SS} = 1$. We marginalize over θ^r and θ^{-r} to get:

$$\begin{aligned} p(y_1, ...y_n|\alpha_{RS}, \alpha_{SS}, r) \propto & \frac{\Gamma(n_{r,RS} + \alpha_{RS})\Gamma(n_{r,SS} + \alpha_{SS})}{\Gamma(n_{r,RS} + n_{r,SS} + \alpha_{RS} + \alpha_{SS})} \times \\ & \frac{\Gamma(N_{RS} - n_{r,RS} + \alpha_{RS})\Gamma(N_{SS} - n_{r,SS} + \alpha_{SS})}{\Gamma(N_{RS} - n_{r,RS} + N_{SS} - n_{r,SS} + \alpha_{RS} + \alpha_{SS})} \end{aligned}$$

We then calculate the likelihood for all keyword combinations generated from the Apriori algorithm and rank them in order from highest to lowest likelihood. The likelihood receives a higher value when a keyword combination is much more prevalent in one group versus another. However, the likelihood does not distinguish whether a keyword combination that characterizes $\{R|S\}$ well versus one that characterizes $\{S|S\}$ well. To resolve this issue of nonidentification, we first orient all decision rules to characterize $\{R|S\}$ (the right-hand side is always $\{R|S\}$). This allows us to drop the right-hand side

of the decision rule and use "keyword combination" and "rule" interchangeably. Then for every rule that is more prevalent in $\{S|S\}$ than $\{R|S\}$ (as measured by the percentage of each group that contains the rule), we simply turn the rule into a NOT rule. For example, if the word "foobar" has a high likelihood and occurs in a greater percentage of $\{S|S\}$ documents than $\{R|S\}$ documents, then the rule becomes NOT "foobar" in characterizing $\{R|S\}$.

Finally, we use the same set of keyword combinations to characterize each cluster by repeating the same process within each cluster, switching off the definition of the $\{R|S\}$ set.[4]

5.4 Usage

The keywords from this algorithm can be of interest themselves or they can serve as interpretable aids for retrieving or identifying the target set. In either case, human input is central to the process.

One place for human inputs is choosing the number of clusters produced by the clustering algorithm. This number can be "estimated", but different numbers of clusters can define different uses and so it is often best to make the choice ex ante. In practice, it is also easy to run the algorithm multiple times with different numbers of clusters to explore the search set and extract keywords. For example, if the application has relatively unambiguous language, one can sometimes only two clusters to roughly divide the search set into an approximation of the target set and everything else. In most cases, however, a larger number of clusters leads to more interesting discoveries. If the search set contains many distinct topics, increasing the number of clusters may allow the algorithm to pick up some of these topics and the keywords associated with them. Furthermore, since documents in different clusters are likely to be about slightly different topics, different keywords are likely to be prominent in different clusters, so choosing a larger number of clusters allows the user to quickly view more relevant keywords. On the other hand, having too many

[4]More specifically, in the keyword combination mining stage, we run the Apriori algorithm on the entire search set to find all unique of keyword combinations, which we then rank for each cluster. The only difference across clusters is in the rank ordering of the keyword combinations and whether a keyword combination becomes a NOT rule.

clusters may result in some clusters without coherent topics or some topics split across multiple clusters.

Once the algorithm produces a set of ranked keywords for each cluster, the user can examine each directly. For example, if the goal is to produce relevant keywords on the reference topic, monitor conversational drift, or further understand of the search set, then the user would look through the suggested keywords in each cluster, choosing those of interest and adding them to the definition of the reference set, likely by expanding the Boolean search with OR operators.

In producing keywords, our algorithm mines from the search set to find keywords that may retrieve the target set hidden within it. One may also mine the reference set directly to find keywords of interest. While we encourage the use of the algorithm for extensions that involve the reference set, we suggest initial focus on mining the search set. For one, the reference set has already been defined by the user with the goal being to find documents within the search set. For another, keywords mined directly from the reference set would perform well only insofar as the documents and words used in the reference set are representative of the target set. In cases such as conversational drift or intentional language changes to evade authorities, the reference set is unlikely to include the new words of interest. In fact, a key benefit of our algorithm is that the reference set need not be a random or representative sample of the target set. In our extensive testing and examples, we generally find that the algorithm picks up most of the important keywords used in the reference set, but it also finds many keywords and keyword combinations that either do not exist in the reference set or keywords that have changed in their importance to the topic relative to the reference set.

6 Illustrations

We now offer four illustrations of the use of algorithm. For simplicity and ease of replicability, we use only six computationally fast classifiers (naïve Bayes, k-nearest neighbor ($k = 5$), logistic regression, support vector machine, decision tree, and random forests), and 100 randomly selected training sets (run as parallel processes). The training sets each

have 1,000 documents (except in the smaller corpus in Section 6.1, where we use 200).

In each case, the algorithm succeeds in finding keywords that would be highly unlikely to be identified by a human being doing this task entirely by hand, as well as some they might have identified but only if they had spent a great deal more time thinking about it. In many cases, the keywords our algorithm discovers appear to be clearly related to the reference topic and may seem quite "obvious" after seeing them. However, relatively few of these words are so obvious that they are chosen by the user without prompting, so the benchmark by which we judge the usefulness of a keyword is not whether a user can imagine that he or she would have come up with the keyword ex ante after seeing it ex post, but rather whether or not a user would have identified the keyword beforehand without the help of our algorithm.

Hindsight bias is of course an extremely well known psychological phenomenon (Roese and Vohs 2012), and so we must be wary of it in evaluating these illustrations. Section 4 demonstrates that humans working without computer assistance generate few keywords in a highly unreliable fashion. The psychological literature is also clear that even when subjects know about hindsight bias, once they know about an outcome (such as the keywords our algorithm has selected), they are unable to "unlearn" this fact and estimate what keywords they would have selected prior to knowing this information. The best estimate of quantities like those come from experiments such as those in Section 4.

6.1 Obamacare and the Supreme Court Decision on the Affordable Care Act

In March of 2010, President Obama signed the *Patient Protection and Affordable Care Act* into law. By the time the Affordable Care Act was brought before the supreme court, in June 2012, the nickname "Obamacare" was commonplace among both in both Democratic and Republican conversations. Because of the many ways of referring to the healthcare act, defining a search on social media that would enable one to follow coverage of the supreme court decision online, and especially to distinguish it from posts about one's personal health or healthcare became a difficult task. Our approach offers a simple solution.

To see this, we create search and reference sets of social media posts of size 367 and

689 respectively from blogspot.com in June of 2012. (Normally, these sets would be much larger, but we offer this simple example here, that nevertheless contains all the essential features, for easy study.) We define the reference set by posts with the word "Obamacare", and so we know with high probability that these posts reference the Obama healthcare act. For the search dataset, we select social media posts that use the keyword "healthcare" (and by definition not the word "Obamacare"). In this first example, we demonstrate the most basic implementation of our algorithm using only two clusters, again for simplicity. We also restricted the keyword combinations to be only those of length 1 (single keywords) and which appear in at least 10 documents in the entire search set.

Table 3 shows the top 25 keywords (from a much longer list) from each of the two clusters found by our algorithm. We find that these lists are helpful distinguishing documents within the search set, some obvious and some that would have been very difficult to identify by hand, without our algorithm. For example, Cluster 1 appears to be composed mainly of documents pertaining to Obamacare. We see many of the keywords relating to the politics of Obamacare (*obama*, *mandate*, *republican*, *senate*, *tax*, *penalty*) and to the Supreme Court rulings on Obamacare (*supreme*, *court*, *constitutional*, *uphold*, *roberts*, *repeal*, *justice*, *decision*, *unconstitutional*). However, we also see a somewhat surprising reference to Congress and the commerce clause (*congress*, *clause*, *commerce*). The reference here refers to the debate over whether or not Congress is allowed to regulate healthcare and impose the individual mandate from Obamacare under the commerce clause in the Consitution. While this was a significant argument that the Supreme Court rejected at the time, it is unlikely that the average user think of searching for the commerce clause when thinking about Obamacare. Cluster 2, on the other hand, appears to consist of documents on a variety of topics related to healthcare that do not pertain to Obamacare. The top keywords from Cluster 2 are less coherent from a topic standpoint because of the diversity of ways people can write about healthcare. Nevertheless, we can see some words related to the healthcare field (*develop*, *medicine*, *intern*, *experiment*, *study*, *hospital*, *clinic*, *medical*, *profession*) as opposed to the political issue of Obamacare.

Cluster 1 ($n = 84$)	Cluster 2 ($n = 283$)
supreme	inform
court	manage
consitutional	develop
obama	medicine
mandate	intern
law	help
uphold	learn
president	train
republican	experiment
congress	study
roberts	resource
senate	city
repeal	industries
insurance	operate
tax	hospital
rule	build
justice	time
afford	food
decision	clinic
penalty	receive
unconstitutional	medical
hill	profession
act	research
clause	during
commerce	data

Table 3: Keywords about Obamacare versus healthcare (but not Obamacare). Keyword stems in black and if stem is not a word, the most logical word used within the search set based on context is used (filled in with red).

6.2 Identifying Child Pornographers Online

Child pornography is illegal in the United States. However, the Internet has made the distribution of such illegal material easier to spread quickly on the open web, on forums, and through peer-to-peer sharing services. Producers and consumers of this material hide in plain sight, trying to stay in contact with each other while still evading the authorities (e.g., `j.mp/NYTSchildP`). The key to fighting the spread of such material online (and preventing the subsequent damages to those pictured) is identifying commonly used keywords that child pornographers use to distribute the images. In this example, we worked with the Department of Homeland Security (DHS) and used our algorithm to help find

keywords that may have been difficult to detect otherwise. A thesaurus or other method used to mine a reference set would be of little use when content creators change their language so frequently and intentionally.

We first identified a reference set of 2,039 documents consisting of forums and blogs containing at least one of four words the DHS knew to uniquely define child pornography. We then used a search set of 9,340 documents defined by the word "pedo", which is sometimes associated with child pornography and at other times used colloquially. The documents were all collected between August and October 2013. We ran the algorithm once with eight clusters. Results from four of these clusters appear in Table 4.

Cluster 1 ($n = 492$)	Cluster 2 ($n = 688$)	Cluster 7 ($n = 1,599$)	Cluster 8 ($n = 3,091$)
lolita	fuck	snimila	que
bbs	lolita	fotomodel	por
fuck	hot	postala	con
hot	love	modny	los
preteen	pussy	yorku	para
nude	teen	pokopali	una
teen	nude	pogreba	las
porn	girl	etala	como
girl	loli	sobu	pero
pussy	porn	petra	del
model	preteen	glorijuvideo	todo
loli	pic	dobio	este
love	model	ubijenog	eso
free	gallery	ubili	cuando
sex	free	milijune	ser
pic	cock	jetiji	porque
young	sex	sovjetske	esta
russian	bbs	ekspedicije	cosa
gallery	russian	tajnolasersko	mucho
cock	ass	procvatu	nada
ass	sexy	sanade	otro
tit	young	nagradu	tiene
sexy	nice	znanstvenici	uno
underage	cum	nedavno	bien
dick	great	glavnu	muy

Table 4: Keywords associated with child pornography versus just "pedo." Keyword stems in black and if stem is not a word, the most logical word used within the search set based on context is used (filled in with red). Only words that appear in at least 1 percent of documents within the cluster are shown. Clusters 3-6 not shown.

The first two clusters produces keywords that are substantively related to child pornography, most of which are sexual in nature. One surprising keyword that appears is *bbs*, which refers to the bulletin board systems that child pornographers often use to exchange links and advice over where to find child pornography. The word *loli* also appears, which seems to be an evolution from *lolita*, a traditional search term for child pornographers. Clusters 3-6 (not shown) do not appear to have any coherent topics based on the top keywords. Interestingly, in cluster 7, the top words all appear to be from Slavic languages, mostly Serbo-Croatian. In cluster 8, the top keywords are all Spanish words. Our algorithm is able to determine the different languages that exist within the search set using the search term "pedo" and the keywords produced allow the user to discern these languages, and rule out those lists, quickly.

6.3 Tweets About the Boston Bombings

April 15, 2013, two bombs exploded near the finish line of the Boston Marathon, killing two people and injuring dozens. In the police chase and city shutdown that followed in the week after the bombings, Twitter was flooded with conversation about the bombings and giving updates of events around the city. Of course, the tweets used different words to describe the bombings, as the conversation about the bombings shifted from the event itself to the investigation and car chase that followed.

In this example, we set our goal to find keywords that could collect the full set of Twitter posts about the bombings from posts related to other events or topics related to Boston. To do so, we define a reference set of 7,643 documents using the hashtag *#BostonBombings* and a search set of 9,655 documents with the word *Boston* but without the hashtag *#BostonBombings*. We gather Twitter data containing these words during the 2013 year and use our algorithm with two clusters to find relevant keywords that separated the bombing incident from other events in Boston.

Table 5 shows the top 25 keyword combinations from our algorithm in each of two clusters. In this example, we included AND statements (noted as "&"). So for example, the rule *suspect & marathon* in Cluster 1 suggests that searching tweets that include both words (not necessarily in order or adjacent). Unsurprisingly, the word *suspect*, either by

Cluster 1 ($n = 3,578$)	Cluster 2 ($n = 6,077$)
suspect	game
police	red
people	sox
fbi	sox & red
say	celtics
suspect & marathon	bruins
report	fan
say	new
terror	tonight
tsarnaev & suspect	los
police & suspect	back
investigate	que
news	come
arrest	win
fbi & suspect	por
attack	york
tsarnaev	play
why	del
kill	chicago
muslim	love
obama	new & york
cnn	see
#bostonmarathon	#mlb
dzhokhar	team
terrorist	series
⋮	⋮
#prayforboston	#jobs
#tcot	college
#benghazi	kevin

Table 5: Keywords about Boston Bombings versus Boston in general. Keyword stems in black and if stem is not a word, the most logical word used within the search set based on context is used (filled in with red). The top 25 keyword combinations and other interesting keywords outside of the top 25 that appear in at least 10 documents within the cluster are shown.

itself or in combination with other words, ranks highly in tweets about the bombings, since social media discussion focused heavily on finding and apprehending the two bombing suspects in the few days after the bombings. Most of the other keywords in Cluster 1 are also substantively related to the bombings, with keywords describing the attack or the suspects (*attack*, *kill*, *tsarnaev*, *dzhokhar*), the relation to terrorist activity (*muslim*, *terrorist*), law enforcement activity (*police*, *fbi*, *investigate*, *arrest*), and media coverage of the event (*report*, *news*, *cnn*).

We also find some surprising keywords in our lists that users may not have been able to identify without our algorithm. For example, the 35th keyword in Cluster 1 (*#tcot*) refers to a popular hashtag on Twitter that stands for "top conservatives on Twitter." This keyword would probably not have been found by users doing keywords by hand but it captures a significant political trend of conservatives responding to the attacks. We also find two hashtags that were used consistently in reference to the bombing attacks: *#bostonmarathon* at number 23 and *#prayforboston* at number 29. At number 99, we find the hashtag *#benghazi*, another keyword that would likely not have been found by hand. This hashtag captures a sentiment among certain individuals that linked both the Boston bombings and the 2012 Benghazi attack on the US diplomatic mission in Libya as terrorist attacks that the Obama administration failed to stop.

In Cluster 2, we find keywords that characterize Twitter discussion related to Boston that is not about the bombings.[5] Interestingly, the overwhelming discussion on Twitter of Boston outside of the bombings is about Boston sports (*game*, *red & sox*, *celtics*, *bruins*, *fan*, *win*, *#mlb*, *series*). We also find a few surprises that users may not have thought about. At number 44, we see the hashtag *#jobs*, which is related to Twitter discussion about job openings available in Boston. At number 77, we find references toward Boston as a university town, including Boston College (*college*). Further down the list at 133, we find the keyword *kevin*, which often refers to Kevin Garnett, the former popular Celtics basketball player. In this illustration of the Boston bombings, we have shown how the algorithm helps us find keywords that are both obvious and non-obvious and are related

[5] We also find a few common Spanish words (*los*, *que*, *por*, *del*), which reflect the presence of Spanish posts in our search set.

to our topic of interest. We have also demonstrated the power of keywords and how our algorithm can help track interesting discussion on Twitter about various other topics.

6.4 Identifying Blog Posts Related to Bo Xilai

In March of 2012, the Chinese government removed Bo Xilai from his post as party chief in Chongqing. Bo had been one of the most popular politicians in China, known for promoting a return to Mao Zedong thought. His policies in Chongqing, known as the "Chongqing model", including fighting corruption and encouraging a return of "red" culture were controversial within the Chinese Communist Party, but were hugely popular within parts of the Chinese population.

Bo, however, is thought to have engaged in many illegal activities himself, including wiretapping of other Communist Party officials as well as having a part in the murder of a British man named Neil Heywood in November, 2011. Wang Lijun, Bo's police chief, became aware of these activities and Bo suddenly demoted Wang in January, 2012. Immediately after his demotion, Wang Lijun fled to the U.S. consulate in Chengdu. Wang reportedly provided the U.S. consulate evidence against Bo, outing the Chinese politician. Not long after the Wang Lijun incident, Bo was removed from his post as party chief and stripped of his membership in the Chinese Communist Party.

Although the Bo Xilai scandal includes many more details than can be delineated here, the main takeaway is that the scandal itself rattled the Chinese Communist Party soon before the November 2012 leadership transition. Coverage of the event was highly controlled within the Chinese news media and objectionable material about the event was quickly removed from the Chinese Internet. "Bo Xilai" (薄熙 来) was heavily censored. However, many netizens continued to talk about the scandal online, without using Bo Xilai's name. For researchers studying the online discussion of the Bo Xilai incident, and public policy officials and others attempting to follow this event in real time, findings alternative keywords used to describe the event is imperative to identifying documents related to the discussion.

To apply our algorithm, we first identify reference and search datasets specific to the scandal. We begin by obtaining all social media posts, before the Chinese government

can censor them (using methods already available; see King, Pan and Roberts 2013), but we still need to find the relevant posts. We define the reference set of 913 social media posts that contain the word "Bo Xilai" (薄熙来) at the time of the scandal, and the search set of 938 posts which use the word "Chongqing" (重庆), the city of which Bo Xilai was the party chief (and by definition, do not use the word "Bo Xilai").

Cluster 1 ($n = 271$)	Cluster 2 ($n = 938$)
王立军 (Wang Lijun)	上海 (Shanghai)
政治 (government)	深圳 (Shenzhen)
事件 (incident)	杭州 (Hangzhou)
市委 (municipal committee)	长沙 (Changsha)
人民 (the people)	武汉 (Wuhan)
打黑 (strike corruption)	中心 (center)
利益 (to benefit)	从事 (undertake)
犯罪 (commit a crime)	文学 (culture)
民主 (democracy)	南京 (Nanjing)
权力 (power)	湖北(Hubei)
文革 (Cultural Revolution)	苏州 (Suzhou)
领导 (leader)	无锡 (Wuxi)
改革 (reform)	西安 (Xian)
群众 (the masses)	专用 (special)
中央中共 (Central Communist Party)	商品 (product)
社会主义 (socialism)	品牌 (brand name)
唱红 (sing red songs)	山东 (Shandong)
黑社会 (black society)	郑州 (Zhengzhou)
干部 (cadre)	广东 (Guangdong)
市政府 (city government)	妈妈 (mother)
路线 (party line)	陕西 (Shanxi)

Table 6: Keywords about Bo Xilai versus Chongqing (but not Bo Xilai). Only words that appear in at least 1 percent of documents within the cluster are shown.

The keywords that our method identifies are presented in Table 6. The words in Cluster 1 are most related to the scandal, while the words in Cluster 2 are less related to the scandal. We immediately see that the most likely word to be related to the scandal is Wang Lijun (王立军), the police chief who reported Bo and "incident"(事件). Many people called the Bo Xilai scandal the "Wang Lijun incident" (王立军事件) or the "Chongqing incident" (重庆事件) instead of using the name Bo Xilai. We also see words that could refer to Bo without his name. "Leader" (领导) in conjunction with Chongqing would accurately describe Bo at the time, without using his name. In addition, there are words specifically related to Bo's policies which many think were related to arrest, in particular his emphasis on singing "red songs" (唱红) and revival of Maoist policies, including reminiscing about the Cultural Revolution (文革) and appealing to the masses (群众).

Cluster 2 suggests words that are related to the city of Chongqing, but not the Bo Xilai

scandal. These words include names of provinces that might be listed with Chongqing (山东, 陕 西), other cities like Chongqing (深圳), and words related to cities like city centers (中心), or shopping (商品). Certainly none of these would help identify text about Bo Xilai or the Bo Xilai scandal.

7 Algorithm Validations

In this section, we include a variety of tests of the method. We begin with a placebo test and show that when there is no pattern in the data, the resulting keywords do not pick up any meaningful keywords or topics. We then test the ability of the algorithm to produce keywords that retrieve the target set with some degree of accuracy. And finally, we address a methodological issue regarding intentionally mislabeled training data.

This methodology is partly supervised and partly unsupervised, with the human-in-the-loop; for methods like these, the best validation of the algorithm is in the context of a real example where if it suggests keywords useful to the user — helping them find words they deem useful faster or that they wouldn't have otherwise thought of — then the method works, and not otherwise.

7.1 Placebo Test

To test against false positives, we now verify that when there is no pattern within the data, the keyword algorithm does not pick up meaningful keywords. We randomly sample 9,988 documents from a selection of all English language social media blogs and reviews from January 2013 to March 2014. Within this set, we then randomly sample 3,000 of the documents to be the reference set with the remaining 6,988 defined as the search set. There should be no significant substantive difference between the two sets on any dimension, and the resulting clusters and keywords from the algorithm should not reflect any meaningful topics.

Table 7 shows the results of our algorithm on the random reference and search sets. As expected, there appears to be no coherent pattern to the keywords and no obvious topic in either of the two clusters. When the reference set is incoherent or when the target set

Cluster 1 ($n = 3,188$)	Cluster 2 ($n = 3,800$)
evaluate	year
accuracy	one
merge	first
stall	two
mini	time
mall	because
password	said
suspense	last
rick	since
taste	also
leak	week
specialist	come
size	think
length	people
verification	govern
pocket	concern
shoe	report
birth	start
android	before
gender	nation
instruct	back
stolen	past
joseph	continue
chat	would
hotel	gold

Table 7: Keywords from random reference and search sets. Keyword stems in black and if stem is not a word, the most logical word used within the search set based on context is used (filled in with red). Only keyword combinations that appear in at least 1 percent of documents within the cluster are shown.

does not exist, the keywords within a cluster should appear to be random as well, and that does appear to be the case.

7.2 Target Set Retrieval

I this section, we validate the ability of our algorithm to produce keywords that retrieve the target set, using several examples where the target set or its approximation is known in advance. The more specific goal is to retrieve as much of the target set as possible, using only keywords, while minimizing the retrieval of documents outside the target set.

To measure the ability to retrieve the target set, we use the two metrics of **recall**

and **precision**, which are widely used in the pattern recognition and information retrieval literature. Recall is the *percentage of the entire target set that is retrieved*, while precision is the *percentage of the retrieved set that belongs to the target set*:

$$\text{Recall} = \frac{\text{\# of target set documents retrieved by keyword(s)}}{\text{\# of documents in target set}} \times 100$$

$$\text{Precision} = \frac{\text{\# of target set documents retrieved by keyword(s)}}{\text{\# of documents retrieved by keyword(s)}} \times 100$$

In retrieving the target set via keywords, we would ideally want to maximize both recall and precision. However, recall and precision are inversely related, so one must usually trade off between the two goals depending on the specific application and goal at hand. In our tests, we find that our algorithm produces keywords that are substantively related to the reference topic in the cluster(s) with higher average probabilities of being classified in the reference set. Each individual keyword has limited recall given that there are rarely words that are common across a large percentage of the target set. However, recall monotonically increases as the number of keywords included with OR statements increases, so combining keywords together in a boolean search can greatly improve recall. Furthermore, as another test of our algorithm, we find that the top keywords in clusters that appear substantively related to the reference topic have much greater precision than keywords in seemingly non-related clusters.

7.2.1 A Composite Search Set

To establish a baseline level of performance, we first create a composite set with 3,000 posts from Twitter for each of five distinct topics from 2013 or early 2014, each one using very specific keywords or hashtags.

1. **SF BatKid**: Tweets about the trending topic of Miles Scott, a five-year-old cancer survivor in San Francisco who became Batkid for a day (see `j.mp/BATkid`), using hashtag *#SFBatKid*

2. **Nelson Mandela's passing**: Tweets about the death of Nelson Mandela, affectionately known as "Madiba," using hashtag *#Madiba*

3. **Hate speech toward Mexicans**: Tweets containing at least one of the terms, *stupid spic*, *stupid spick*, *wetback*, *fuckin spic*, or *fuckin spick*. This example was motivated by government agencies that might want to flag hate speech online.

4. **Geek chic**: Tweets about the fashion trend using the phrase *geek chic*.

5. **Boston Marathon bombing**: Tweets about the Boston Marathon bombing using the hashtag *#Bostonbombings* (same data used

We combined the 3,000 posts from each topic to create a search set of 15,000. We then created a reference set of 4,102 separate tweets with hate speech using the same search terms as in topic 3. Since we know which of the 15,000 posts corresponds to the hate speech topic, we can then run our algorithm to find keywords using our hate speech reference set and measure the keywords on the recall and precision metrics.

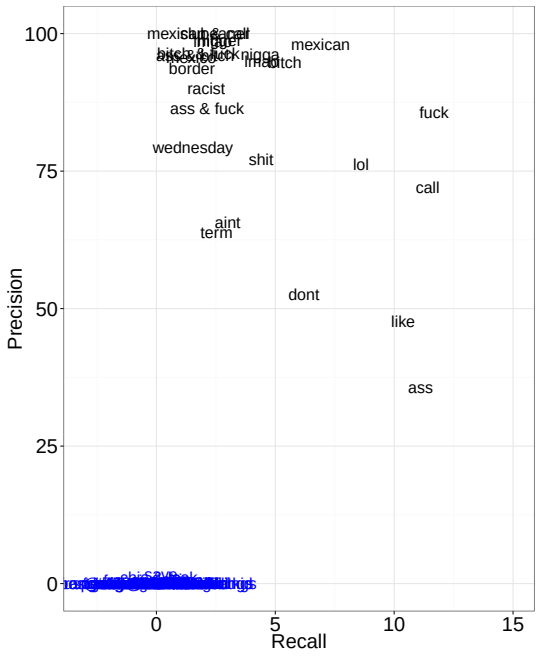
Figure 2a displays the top 25 keywords in each of the two clusters in one run of the algorithm. In the corresponding Figure 2b, we plot each keyword with recall horizontally and precision vertically, measuring how well it can retrieve the target set of hate speech posts. Colors denote Cluster 1 (black) and Cluster 2 (blue).

The results in Figure 2a clearly show that the keywords in Cluster 1 are very much related to hate speech, with various expletives, derogatory terms, and other terms about Mexicans. The keywords in Cluster 2, in contrast, are a combination of the most significant terms in the other four topics, with much lower recall and precision for the target set of interest. The recall and precision metrics also show that the terms in Cluster 1 are much higher in both recall and precision than Cluster 2. Any single keyword has relatively low recall given how varied a typical target document set is in terms of word usage, but the precision is almost always quite high. In this simplest of examples, it is clear that the algorithm performs as desired in producing meaningful keywords that help retrieve the target set.

In Figure 3, we simulate one possible usage of our algorithm by retrieving the target set using the set of keywords in Cluster 1 in Figure 2 and combining them with OR statements. We start by taking the first keyword in the list (*fuck*) and measuring recall and precision in retrieving the target set. The values are denoted by the point "1" in the

Cluster 1 ($n = 3,183$)	Cluster 2 ($n = 11,806$)
fuck	#sfbatkid
ass	chic
call	geek
mexican	#bostonbombings
lol	geek & chic
bitch	#madiba
nigga	suspect & #bostonbombings
lmao	suspect
beaner	city & #sfbatkid
shit	@sfwish & #sfbatkid
nigger	save & #sfbatkid
like	san & #sfbatkid
aint	@sfwish
dont	boston & #bostonbombings
lmfao	gotham & #sfbatkid
racist	save
mexican & call & fuck	batkid & #sfbatkid
bitch & fuck	make & #sfbatkid
mexico	francisco & san & #sfbatkid
ass & fuck	francisco & #sfbatkid
ass & bitch	day & #sfbatkid
slur	geek & look
border	chic & look
term & bitch	geek & chic & look
wednesday	francisco & san

(a) Top 25 Keywords



(b) Keywords with recall and precision of target set

Figure 2: Keywords from composite search set and reference set on hate speech. Keywords stems from Cluster 1 are in black and keywords stems from Cluster 2 are in blue. In Figure 2a, if keyword stem is not a word, the most logical word used within the search set based on context is used (filled in with red).

figure to represent the first keyword. Next, we take the first two keywords and combine them to make an OR statement (*fuck* OR *ass*). We then retrieve documents that contain at least one of the first two keywords and measure recall and precision. The values are denoted by the point "2" in the figure, which means that we used 2 keywords in an OR statement starting from the top. We continue this process of adding each keyword down the list into our search term until we include all of the top 25 keywords in our search (denoted by the point "25" in our figure), finding documents that contain at least one of the 25 keywords.

In Figure 3, the pattern becomes very apparent. Using more keywords in an OR combination monotonically increases recall as we retrieve more documents. Precision generally decreases, although the relationship is not monotonic. In our simple example, searching for documents in the search set with at least one out of the top 25 keywords retrieves approximately 60 percent of the target set, but out of those documents found, a little less than half are not from the target set.

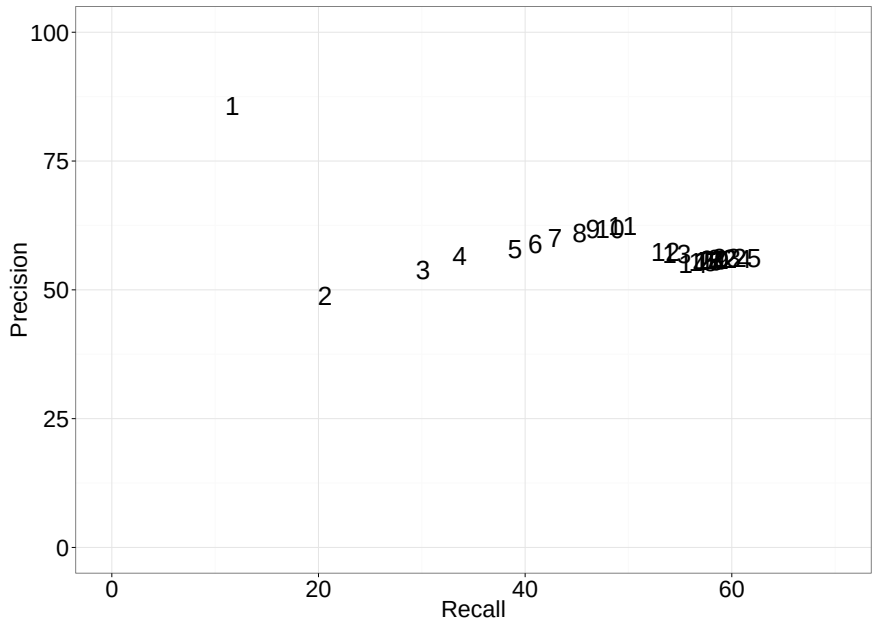


Figure 3: Recall and precision combining keywords using OR statements. The numbers in the figure represent the number of keywords were combined using OR statements while searching for target set documents. We start from the top of the list in Cluster 1 in Figure 2. For example, "5" means that we searched for documents containing any one of the first five keywords in the list.

7.2.2 A Hashtag-labeled Search Set

We now create a more heterogeneous validation set where, as usual, the division between topics is more ambiguous. To do this, we take real examples where authors of Twitter posts imperfectly code the topics themselves via hashtags. We choose three examples from popular hashtags during 2013–2014 (see first column of Table 8). For each, we pick a set of posts defined by the hashtag (second column). Within this set, we define a reference set using another term used in this set of posts (third column). The remainder of the set not chosen becomes the target set. We then combine the target set with added search set documents defined by another term (last column) to create the entire search set.

The hashtag thus "codes" the topic of both the reference and target sets. We use the hashtag for coding purposes only and remove it from the documents before the classification step. While any post that includes the hashtag is likely to be about the same topic, there may also be posts in the search set about the topic that do not include a hashtag.

In the first example, which was also used in Section 7.2.1, we examine tweets about

	Topic	Hashtag	Reference Set	Added to Search Set
1.	Nelson Mandela's passing	*#Madiba*	*Mandela* OR *Nelson Mandela*	*South Africa*
2.	SF BatKid	*#SFBatKid*	*Miles*	*San Francisco*
3.	Women of color within the feminist movement	*#SolidarityIsForWhiteWomen*	*white feminist* OR *white feminists*	*racism*

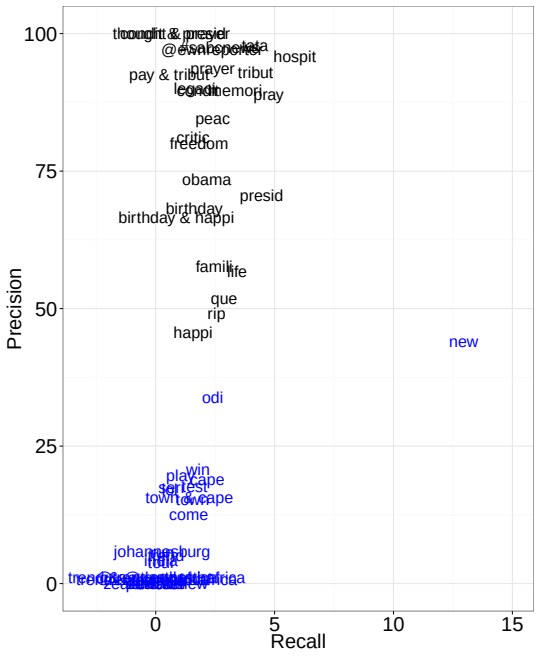
Table 8: Description of Hashtag Verification Document Sets. For each example, the reference set contains the terms indicated under the second and third columns. The target set contains the hashtag without the reference set terms. The additional search set does not contain the hashtag or the reference set terms. The entire search set used in the algorithm is the combined target set and additional search set.

President Nelson Mandela's passing in December 2013 with a reference set of 3,147 tweets and a search set of 13,797 tweets in 2013 or early 2014. We want keywords that distinguish tweets about Mandela from tweets about South Africa in general. The hashtag that we use to code tweets about Mandela, *#Madiba*, was Mandela's clan name as well as a name for him that was generally used affectionately. We ran our algorithm using two clusters and present recall and precision for the target set of tweets with *#Madiba* but without *Mandela* or *Nelson Mandela*.

Figure 4a shows the top 25 keywords from the two clusters. We find various words describing Mandela's passing (*hospitalized*, *critical*, *condition*) as well as tributes to him (*tribute*, *prayer*, *freedom*, *pray*, *thought & prayer*, *legacy*). We also find a non-obvious affectionate name for Mandela in *tata*, which means "Father" given that Mandela was often described as "the father of the nation." Interestingly, we also find references to *@ewnreporter* and *#sabcnews*, two entities on social media that covered Mandela's passing heavily. Users would have likely been unable to find these references without the use of our tool. Since our search set encompasses a large timespan, our target set includes tweets about Mandela's birthday as well as about his passing (*birthday*, *birthday & happy*). In Cluster 2, we find keywords that are related to South Africa more generally apart from Mandela. Capetown (*cape & town*) appears significantly as one of the most discussed cities in South Africa. The Twitter account *@trendssthafrica*, which has 80,000+ followers and tracks South Africa Twitter trends, appears as a significant keyword as well. In Figure 4b, we show the same pattern as before that the top keywords in Cluster 1 have

Cluster 1 ($n = 5,046$)	Cluster 2 ($n = 8,743$)
tribute	trend
prayer	cape
president	town
family	town & cape
birthday	come
hospitalized	@trendssthafrica
tata	trend & @trendssthafrica
freedom	pakistan
pray	new
birthday & happy	test
que	india
happy	tour
life	australia
critical	run
condition	win
peace	zealand
@ewnreport	zealand & new
rip	johannesburg
memory	odis
#sabcnews	lol
thought & prayer	trendssthafrica
obama	trend & trendssthafrica
legacy	series
pay & tribute	#cricket
condition & president	play

(a) Top 25 Keywords



(b) Keywords with recall and precision of target set

Figure 4: Keywords about Nelson Mandela versus South Africa. Keywords stems from Cluster 1 are in black and keywords stems from Cluster 2 are in blue. In Figure 4a, if keyword stem is not a word, the most logical word used within the search set based on context is used (filled in with red).

high precision in retrieving the target set compared to keywords in Cluster 2.

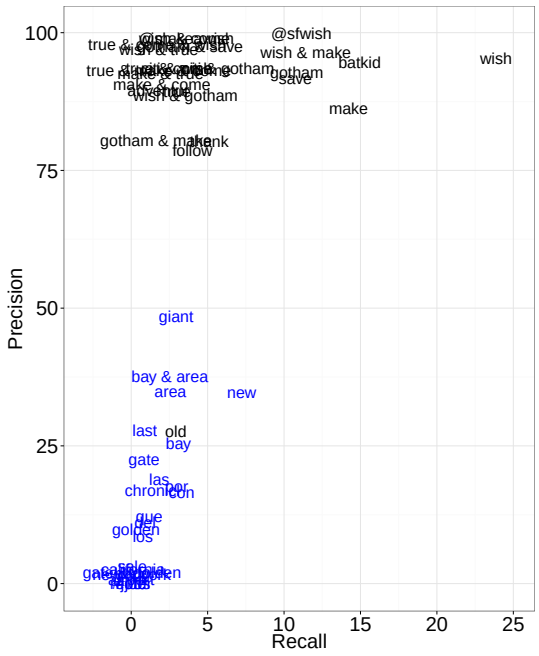
The second hashtag verification example, also used in Section 7.2.1, pertains to the San Francisco BatKid event. We use a reference set of 293 tweets and a search set of 14,901 tweets from 2013 and early 2014. We run our algorithm here with two clusters to try to distinguish keywords about BatKid in a search set that contains a target set labeled with *#SFBatKid* and also a general search set about San Francisco.

Figure 5a shows the top 25 keywords from our two clusters. In Cluster 1, we see relevant words describing the event (*save*, *city & gotham*, *adventure*), positive reaction to the event (*wish*, *wish & make*, *thank*, *true & make & come*), and references to the Make-A-Wish foundation that sponsored the event (*@sfwish*, *@makeawish*). In Cluster 2, we see generic words at the top concerning San Francisco (*chronicle*, *golden & gate*, *california*, *bay & area*, *giants*). We also see the same references to job searches (*#jobs*, *#job*) that we saw in the Boston example.

In Figure 5b, we show that in general, the top keywords in Cluster 1 outperform the

Cluster 1 ($n = 4,409$)	Cluster 2 ($n = 10,483$)
@sfwish	new
wish	los
gotham	que
wish & make	bay
save	chronicle
make & gotham	del
thank	gate & golden
city & gotham	golden
follow	california
batkid	gate
old & @sfwish	bridge
gotham & save & @sfwish	por
true	area
true & come	new & york
wish	last
adventure	york
gotham & make	bay & area
wish & come	airport
make & true	las
make & come	con
wish & true	#jobs
city & wish	#job
@makeawish	reuters
true & make & come	sale
true & come & wish	giants

(a) Top 25 Keywords



(b) Keywords with recall and precision of target set

Figure 5: Keywords about SF BatKid versus San Francisco. Keywords stems from Cluster 1 are in black and keywords stems from Cluster 2 are in blue. In Figure 5a, if keyword stem is not a word, the most logical word used within the search set based on context is used (filled in with red).

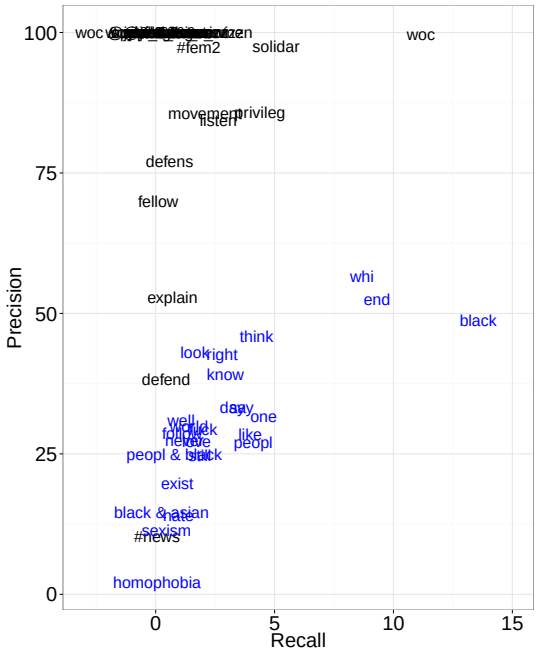
top keywords in Cluster 2 in both recall and precision of the target set. The group of blue points in the top left corner are the keywords from Cluster 2 that originate from the 94 posts referenced above. We show here that the algorithm successfully suggests keywords to the user to retrieve the target set and groups the keywords in an imperfect manner that nonetheless can be very helpful to the user.

The final hashtag verification example we present is from a controversial debate on women of color in the feminist movement. In July 2013, noted author, blogger, self-proclaimed feminist, and professor of gender studies Hugo Schwyzer announced he was "quitting the Internet" due to mental health and family concerns. In August 2013, he admitted on Twitter that he had been involved in the abuse of women of color and was complicit in racism. As a response, Mikki Kendall started the Twitter hashtag *#SolidarityIsForWhiteWomen* to spur debate over the role of women of color in the feminist movement. Women of color began using the hashtag to voice frustration over the condescension and racism that they faced from the feminist movement. We use a reference set

of 219 tweets and a search set of 15,668 tweets from 2013 and early 2014. We use our algorithm with two clusters here to try to retrieve the target set and keywords related to the debate amongst a set of tweets that mention the word "racism" in general.

Cluster 1 ($n = 4,820$)	Cluster 2 ($n = 10,828$)
woc	people
privilege	black
listen	think
@just_wizz_wizz	one
movement	like
woc & listen	love
#fem2	day
need & woc	know
support & woc	say
woc & get	still
defend	never
privilege & woc	people & black
yet & woc	end
defensive	follow
racist & woc	fuck
woc & @just_wizz_wizz	well
try & woc	exist
call & woc	why
woc & women	hate
explain	world
fellow	right
woc & movement	homophobia
#news	black & asian
solidarity	sexism
silence & woc	look

(a) Top 25 Keywords



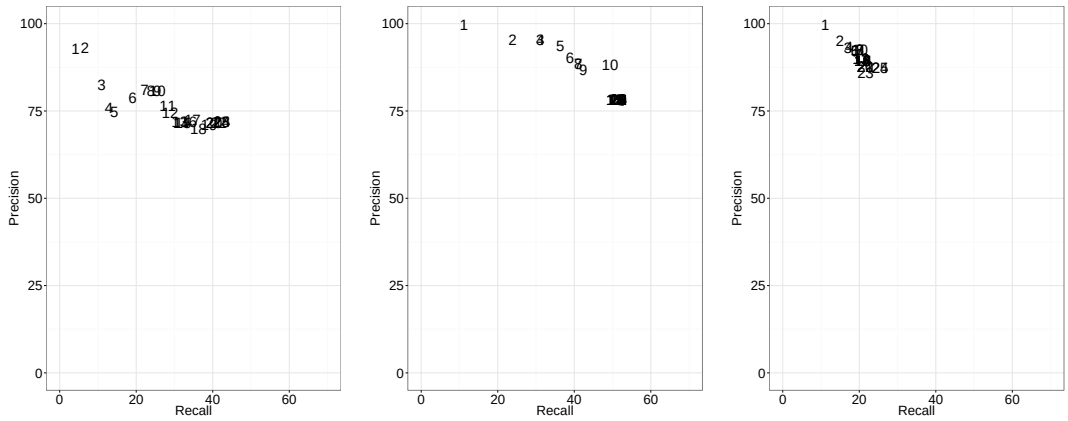
(b) Keywords with recall and precision of target set

Figure 6: Keywords about women of color in the feminist movement versus racism in general. Keywords stems from Cluster 1 are in black and keywords stems from Cluster 2 are in blue. In Figure 6a, if keyword stem is not a word, the most logical word used within the search set based on context is used (filled in with red).

Figure 6a shows the top 25 keywords from our two clusters. In Cluster 1, we see keywords relating to the movement such as the term *woc*, which stands for "women of color." We see other interesting and non-obvious keywords such as *privilege*, *listen*, *explain*, *solidarity*, and *silence & woc*. We also see reference to a Twitter account (*@just_wizz_wizz*) that was actively involved in tweeting about the issue as well as a hashtag reference to the feminism conference Fem 2.0 (*#fem2*). In Cluster 2 we generally find vague and generic terms that are often used in conjunction with discussions of race. In Figure 6b, we show again that the keywords we find in Cluster 1 perform well in precision when retrieving the target set.

We also simulate here one potential usage of combining keywords in OR statements to improve recall of the target set. We take the top 25 keywords from Cluster 1 of each

example and we combine them starting from the top sequentially.



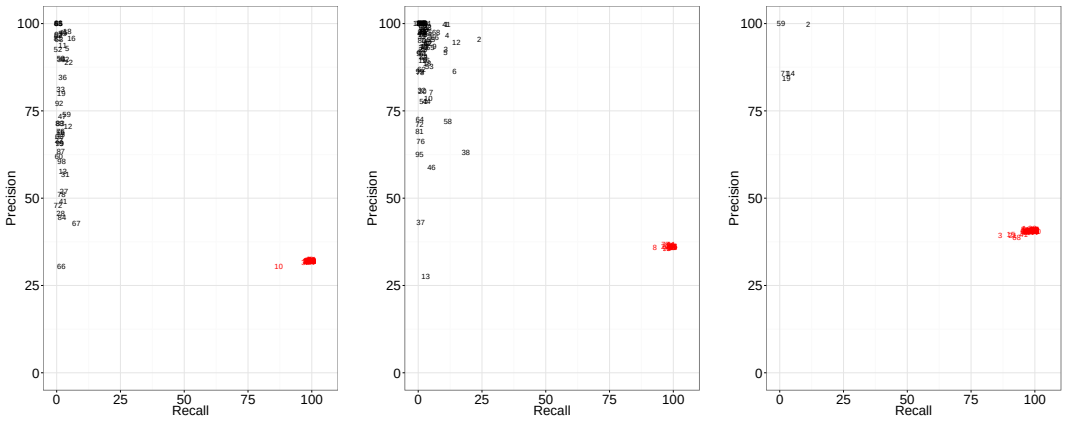
(a) Mandela (b) SF BatKid (c) Solidarity

Figure 7: Recall and precision combining keywords using OR statements for hashtag verification examples. The numbers in the figure represent the number of keywords were combined using OR statements while searching for target set documents. We start from the top of the list in Cluster 1 for each example. For example, "5" means that we searched for documents containing any one of the first five keywords in the list.

Figure 7 shows the results of combining keywords in OR statements for each example. The numbers on the plot represent the number of keywords that were combined in the OR statement, starting from the top of the list. As expected, recall increases monotonically in the number of keywords used to search for the target set while precision generally decreases. Also, we can see that the extent to which recall improves differs across examples. In the SF BatKid example, finding documents with at least one of the top 25 keywords leads to a recall of nearly 60 percent of the target set, while using the top 25 keywords in the Solidarity example only leads to a recall of just over 20 percent of the target set. This most likely reflects the heterogeneity of word usage across examples, with a larger set of words being using to describe women of color in the feminist movement than the SF BatKid event.

Finally, we simulate another usage of the algorithm by incorporating NOT rules in retrieving the target set. NOT rules include keywords that describe documents outside a cluster better than within a cluster. Since the likelihood does not differentiate between keywords that describe the cluster well versus keywords that describe the outside the cluster well, we consider all keywords which occur in a higher proportion outside of a

cluster versus in a cluster as a NOT rule for the cluster. For example, suppose the word "foobar" was ranked highly by the likelihood for Cluster 1. If "foobar" was ranked highly because it was highly prevalent outside Cluster 1 and not prevalent inside Cluster 1, then the rule becomes "(NOT) foobar" for Cluster 1. In the case of only two examples, the top descriptive keywords for Cluster 2 become NOT rules for Cluster 1 and vice versa. Previously, we discarded all NOT rules for simplicity, but we now show how NOT rules can also help improve recall in target set retrieval.



(a) Mandela (b) SF BatKid (c) Solidarity

Figure 8: Recall and precision with NOT statements for hashtag verification examples. The numbers in the figure represent the ranking from the top 100 rules in Cluster 1 of each example. NOT rules are colored in red.

In Figure 8, we take the top 100 rules for Cluster 1 including any NOT rules that may rank highly. The numbers on the plot represent the ranking of each keyword rule (with NOT rules in red). Immediately, we notice that all the NOT rules rank very highly on recall for the target set. Consider the top rule for the Mandela example. Previously, for Cluster 1, *tribute* was the top ranking keyword. However, when we include NOT rules, *(NOT) trend* becomes the top ranking rule, with very high recall. If the user would like to retrieve the target set, the best first move would be to discard all posts with the keyword *trend* in it. This performs better on recall than simply picking posts with the word *tribute* since the former leaves almost the entire target set in tact while the latter only picks out a portion of the target set. However, as expected, the former also performs poorly on precision because keeping all posts without the word *tribute* also includes many posts that

are not in the target set. From these examples, we can see that including NOT rules can help the user search for the target set by helping to discard posts that are not of interest to the user.

7.2.3 Robustness to Mislabeled Training Data

One concern about the algorithm is that the classification step we use may be incorrect due to what the machine learning literature terms as "mislabeled training data" or "class noise." Recall that the first stage of the algorithm defines a reference and search set, draws a random training set, and then uses multiple classifiers to classify the search set documents into search or reference, leveraging the mistakes of the classifier in order to find the target set. If we label the reference set documents as 1s and the search set documents as 0s then the target set documents are those that are really 1s but were mislabeled as 0s. Since our training data likely contains at least some of these target set documents, our classifiers are trained on mislabeled data, which may produce inaccurate or biased predictions for the search set, which in turn may affect the rest of the algorithm and the resulting keywords.

There are typically two imperfect ways to handle mislabeled data in classification. The first is to use or develop classifiers that either incorporate the mislabelling process directly or are "robust" to a certain proportion of mislabelling in the data (Masnadi-Shirazi, Mahadevan and Vasconcelos 2010, Bootkrajang and Kabán 2012). However, these models usually are complex and often hard to estimate. The field of "robust statistics" has developed estimators that are not unduly affected by deviations from model assumptions such as outliers (see Huber (1981) for example), which is closely related to the problem of mislabeled data. However, the methods developed are largely for estimating regression parameters rather than classification in the machine learning context. The second way of handling mislabeled data, which is more popular in the machine learning literature, is to first preprocess the training data by filtering out suspected mislabeled data before running the classifiers. Various methods of filtering have been suggested using different classifiers (Sánchez et al. 2003, Zhu, Wu and Chen 2003, Venkataraman et al. 2004), an ensemble of classifiers (Brodley and Friedl 1999, Verbaeten and Van Assche 2003), or unlabeled data (Guan et al. 2011). However, the filtering option is also not a perfect solution since often

the filtering methods themselves are based on the mislabeled data and several mislabeled observations might "mask" each other and remain undetected by the filtering process. Our suggestion for handling mislabeled data is that either of the two ways may be used with our algorithm. Users may choose to implement classifiers that are robust to mislabeled data in the first step and/or also first filter the training data for mislabeled data before training the classifiers.

However, we also show that even without implementing any particular methods to handle mislabeled data, our algorithm can be fairly robust and produce relevant keywords. We believe that the use of multiple classifiers may often reduce the error caused by mislabeled data since different classifiers are differently susceptible and affected by the errors. Furthermore, we note that our algorithm does not strictly require correct probabilities or classifications from the classifiers. Since we cluster multiple classifier predictions, simply retaining the rank ordering of the probabilities of each search document may be enough for our clustering algorithm to approximate the target set and generate useful keywords. Finally, depending on the composition of the search set, the algorithm may only require a small number of target set documents to be clustered together to produce meaningful results.

To show the robustness of our algorithm to mislabeled training data, we reuse the Mandela Twitter example from above with the same reference set. We hold the size of the search set (a subset of which is the target set) constant at 5,000 posts. We then vary the proportion of the search set that is the target set, running the algorithm on a search set with 1%, 10%, 25%, 50%, and 75% of its posts as the target set. Specifically, we first run the algorithm with 2 clusters with 4,950 randomly selected non-Mandela related search posts and 50 (1%) randomly selected Mandela-related target set posts tagged with the *#Madiba* hashtag. We then randomly add 450 *#Madiba* posts and remove 450 non-Mandela related posts so that our new search set consists of 10% in the target set and so forth. If mislabeled training data highly affects our results, then the problem should increase as the target set proportion increases.

Table 9 shows the results of our algorithm on the search sets with different target set

Target set	1%	10%	25%	50%
Keywords	president	president	hospital	birthday
	family	tribute	tribute	family
	die	zuma	president	birthday & happy
	zuma	birthday	birthday	president
	birthday	hospital	tata	happy
	obama	pray	prayer	condition
	pray	obama	pray	tribute
	freedom	freedom	freedom	que
	long	family	critical	tata
	follow	die	inspire	critical
	memories	inspire	birthday & happy	bless
	wish	prayer	remain	por
	man	never	condition	inspire
	celebrate	birthday & happy	peaceful	follow
	remember	@ewnreporter	#sabcnews	thought
	continue	critical	jacob	los
	leader	happy	happy	condition
	life	@encanews	family	pour
	inspire	peaceful	bless	#obama
	heart	tata	thought	condition & president
	media	#sabcnews	jacob & zuma	condition & critical
	arrive	legacy	man	obama
	state	bless	rest	soon & get
	very	bieber	que	well & soon
	never	lie	@chriseldalewis	rest
Recall	84	75.4	66.48	55.76
Precision	2.79	23.34	46.45	68.13
Cluster n	1,506	1,615	1,789	2,046

Table 9: Keywords associated with Mandela with varying target set proportion of search set. Keywords are from Cluster 1 of each run. Recall, precision, and size of the cluster are shown. Search set size is fixed at 5,000. Keyword stems in black and if stem is not a word, the most logical word used within the search set based on context is used (filled in with red).

proportions. We show Cluster 1 only for each run of the algorithm. Since our target set gets larger for each run, the keyword results are not expected to be the same across the different runs. However, what is important is that the topic of keywords across all runs seem remarkably stable. With either small or large target sets, the algorithm is still able to pick up keywords relating to Mandela and his passing. Furthermore, the cluster is able to consistently recall a majority of the target set posts and even when precision of the cluster is low, the algorithm still manages to rank Mandela-related keywords highly. Although these results are not comprehensive proof of robustness to mislabeling, they nonetheless suggest that the algorithm is still helpful for users regardless of the level of mislabeling.

8 Prior Literature

Although our algorithm is focused on extracting keywords from unstructured data, it is related to methods for the analysis of structured data, such as "query expansion" methods, which include algorithms that add or reweight keywords within search queries in order to retrieve a more representative set of documents (Xu and Croft (1996), Rocchio (1971), or Carpineto and Romano (2012) for a review). Our approach and current query expansion algorithms differ in two main ways. First, most query expansion methods retrieve new keywords by stemming the original keyword, applying synonyms to the original keyword, 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 corpea, that do not include the original keyword.

While some query expansion methods use large external corpea, such as Wikipedia, to enhance keyword retrieval (Weerkamp, Balog and de Rijke 2012), our method allows the user to define the external corpus by a general keyword without any structured or predefined data. We thus rely on the user's expertise to define the search set 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). 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 corpea of documents; see Carpineto et al. (2009) for a review.) It then retrieves keywords that are likely to appear in these document groups, but unlikely to appear in the rest of the search dataset.

9 Concluding Remarks

The computer-assisted, iterative algorithm we propose here learns from 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 by hand. The algorithm discovers keywords, and associated document sets, by mining unstructured text, defined by the user, without requiring structured data. The way this enables the statistical problem to be posed also opens up a new range of applications for further analyses.

References

Agrawal, Rakesh and Ramakrishnan Srikant. 1994. "Fast Algorithms for Mining Association Rules." *Proceedings of the 20th VLDB Conference* pp. 487–499.

Apte, Chidanand, Fred Damerau and Sholom M. Weiss. 1993. "Automated Learning of Decision Rules for Text Categorization." *ACM Transactions on Information Systems* 12(3):233–251.

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 p. 195–220.

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

Bootkrajang, Jakramate and Ata Kabán. 2012. Label-Noise Robust Logistic Regression and Its Applications. In *Machine Learning and Knowledge Discovery in Databases*. Springer pp. 143–158.

Brodley, Carla E. and Mark A. Friedl. 1999. "Identifying Mislabeled Training Data." *Journal of Artificial Intelligence Research* 11:131–167.

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.

Cohen, William W. 1996. Learning Rules that Classify E-Mail. In *AAAI Spring Symposium on Machine Learning in Information Access*.

Cohen, William W. and Yoram Singer. 1999. "Context-Sensitive Learning Methods for Text Categorization." *ACM Transactions on Information Systems* 17(2):141–173.

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.

Guan, Donghai, Weiwei Yuan, Young-Koo Lee and Sungyoung Lee. 2011. "Identifying Mislabeled Training Data with the Aid of Unlabeled Data." *Applied Intelligence* 35:345–358.

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, January):229–247. http://gking.harvard.edu/files/abs/words-abs.shtml.

Huber, Peter J. 1981. *Robust Statistics*. Wiley.

Kaufman, Leonard and Peter Rousseeuw. 1990. *Finding Groups in Data: An Introduction to Cluster Analysis*. New York: Wiley.

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.".

Masnadi-Shirazi, Hamed, Vijay Mahadevan and Nuno Vasconcelos. 2010. On the Design of Robust Classifiers for Computer Vision. In *IEEE International Conference on Computer Vision and Pattern Recognition*. pp. 779–786.

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.

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.

Roese, Neal J and Kathleen D Vohs. 2012. "Hindsight bias." *Perspectives on Psychological Science* 7(5):411–426.

Sánchez, José Salvador, Ricardo Barandela, AI Marqués, Roberto Alejo and Jorge Badenas. 2003. "Analysis of New Techniques to Obtain Quality Training Sets." *Pattern Recognition Letters* 24(7):1015–1022.

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.

Venkataraman, Sundara, Dimitris Metaxas, Dmitriy Fradkin, Casimir Kulikowski and Ilya Muchnik. 2004. Distinguishing Mislabeled Data from Correctly Labeled Data in Classifier Design. In *16TH IEEE International Conference on Tools with Artificial Intelligence*. pp. 668–672.

Verbaeten, Sofie and Anneleen Van Assche. 2003. Ensemble Methods for Noise Elimination in Classification Problems. In *Multiple Classifier Systems*. Springer pp. 317–325.

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

Zhu, Xingquan, Xindong Wu and Qijun Chen. 2003. Eliminating Class Noise in Large Datasets. In *Proceedings of the 20th International Conference on Machine Learning*. pp. 920–927.