Exploratory Analysis of
Highly Heterogeneous Document Collections

Arun S. Maiya, John P. Thompson, Francisco Loaiza-Lemos, and Robert M. Rolfe
Institute for Defense Analyses — Alexandria, VA, USA
{amaiya, jpthomps, floaiza, rolfe}@ida.org

ABSTRACT

We present an effective multifaceted system for exploratory analysis of highly heterogeneous document collections. Our system is based on intelligently tagging individual documents in a purely automated fashion and exploiting these tags in a powerful faceted browsing framework. Tagging strategies employed include both unsupervised and supervised approaches based on machine learning and natural language processing. As one of our key tagging strategies, we introduce the KERA algorithm (**K**eyword **E**xtraction for **R**eports and **A**rticles). KERA extracts topic-representative terms from individual documents in a purely unsupervised fashion and is revealed to be significantly more effective than state-of-the-art methods. Finally, we evaluate our system in its ability to help users locate documents pertaining to military critical technologies buried deep in a large heterogeneous sea of information.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Text Analysis*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Natural Language*

General Terms

Algorithms; Experimentation; Human Factors

Keywords

faceted navigation, faceted browsing, tag clouds, machine learning, keyphrase extraction, topic modeling

1. INTRODUCTION AND MOTIVATION

Given a large and diverse collection of unstructured text documents, how does one (1) characterize the subject areas present and (2) use these discovered subject areas to efficiently navigate the collection to locate critical information? Many previous works have investigated such questions within specific domains such as microblog posts (*e.g.,* characterizing tweets [10]), but comparatively less attention has been paid to investigating more general and diverse contexts. Unfortunately, in practice, approaches that may work well for domains consisting exclusively of a single document type (*e.g.,* tweets, emails, or scientific articles) do not always translate easily or directly to other more *heterogeneous* and "messy" document collections. In this work, we present a tag-based system in which tags (*i.e.,* terms or character strings automatically assigned to individual documents) are exploited to efficiently characterize and explore document collections. Document collections of interest in our work exhibit a high degree of diversity (*e.g.,* arbitrary files residing on a high-capacity laptop drive or file server). The U.S. federal government, for instance, is often presented with the challenge of what essentially is exploratory analyses of highly heterogeneous document collections. These collections requiring analyses are nowhere near as homogeneous as tweets, news stories, patents, or scientific abstracts — the typical objects of study in text analytics research (*e.g.,* [2, 8, 21, 22, 26]). To better illustrate this point, we briefly describe three motivating examples.

1. **Digital Investigations:** In 2012, it was reported that General David Petraeus, CIA Director, had been having an affair with his biographer, Paula Broadwell [18]. In the ensuing investigation, the FBI discovered that a laptop owned by Paula Broadwell may have contained sensitive classified information, constituting a security violation [18]. Reviewing computers for sensitive or critical information is a task that arises in many scenarios. Examples include digital forensics and the identification of critical information (*e.g.,* trade secrets) exposed through cyber intrusions. Performing such reviews and making such discoveries can be extremely difficult and burdensome, as analysts are faced with the challenge of locating critical information buried deep in a large heterogeneous sea of files.

2. **Intelligence Analysis:** Intelligence Analysis generally involves acquiring knowledge of subjects, entities, or situations of interest and characterizing and understanding possible future scenarios [13]. Much of this situational awareness is achieved through analyses of unstructured text collections comprised of diverse sets of document types and file formats. The size and breadth of information embedded in such collections can be overwhelming to intelligence analysts.

3. **Appraisal of Electronic Records:** The National Archives and Records Administration (NARA) is charged with determining the value of federal records for archival purposes [12]. This process, known as *document appraisal*, makes documents either permanent or temporary and involves time-consuming reviews of massive collections of documents that are diverse in both their content and format [12].

All three of the above examples necessitate the need for an efficient and intelligent way to explore heterogeneous, large-scale document collections for critical information of interest. In the present work, we explore the task of characterizing, browsing, and searching large collections of unstructured text documents using *faceted navigation*. We present a system that effectively discovers and exploits the use of *information facets* to efficiently characterize and search large document collections.

1.1 Information Facets

Information facets (or simply *facets*) are classes of attributes describing objects in an information repository. They are used to facilitate searches, filtering, and navigation of the information by different dimensions of the data [20, 24]. Faceted classification systems were first conceived in the 1930s by S.R. Ranganathan, an Indian librarian considered to be the father of library science [20]. Today, faceted search is used extensively in information retrieval systems (*e.g.,* [24]).[1] Electronic commerce sites, for instance, employ facets to facilitate browsing products along various dimensions (*e.g.,* brand, price). The vast majority of faceted search systems populate the facet attributes from *pre-existing* fields in a data repository. One example is Twitter's use of *hashtags*, which are user-generated topic tags assigned to tweets [10]. Another such example is an *author* or *title* field in a publication database. Unfortunately, for most large document collections, these manually-generated tags typically do not exist. For these cases, the attributes used to populate facets must be mined or *discovered*.

What facets should be used for unstructured text and how might we populate them in an automated fashion (*i.e.,* discover them)? In the context of a physical library system, Ranganathan proposed classifying information according to five, manually-populated facet categories he referred to as Personality, Matter, Energy, Space, and Time (PMEST) [20]. Motivated by the PMEST model and today's surge in electronic records, we propose organizing unstructured digital text collections by the following general set of discoverable facet categories.[2]

Topic Facets. Topic Facets relate to the overall subject or "aboutness" of an electronic document.[3] In the present work, our focus is on the automated discovery of *topic tags*. *Topic tags* are key terms that capture or represent the overall topic of the document. Such tags can be used to characterize and navigate document collections and refine search results. The problem of discovering topics (and topic tags) from text collections has been extensively studied, of course. In Section 3, we discuss multiple approaches to building *Topic Facets*.

Mention Facets. Documents often make mention of entities, relations, or events that are of importance despite being unrelated to the overall topic of the document. For instance, Personally Identifiable Information (PII) such as social security numbers are often important to detect, yet are not necessarily representative of the subject of a given document. The same is true of username mentions in tweets. Mention Facets are populated by extracting such entities (or relations) from text and can be used to help discover and locate information of interest within a document collection.

Format Facets. The Format Facet allows navigation of document collections by file type and format. For homogeneous text collections like tweets, the Format Facet is rather uninteresting, as all tweets have the same format (*e.g.,* plain text and up to 140 characters) . In our work, however, the document sets from which topic tags and entity mentions are extracted are highly diverse in both type and format. Collections can include technical reports, news articles, Powerpoint briefs, Excel files, Web pages, programming language source code, emails, and many other files. In practice, the availability of a Format Facet is especially important for helping to hone in on particular information elements of interest. For instance, different file types often cover different sets of topic areas and entities. Moreover, the file type and format affect the way in which topics, entities, and terms of interest are extracted (*e.g.,* a scientific report in PDF format *vs.* a .dat file containing Web search history).

Location Facets. The meaning and implementation of the Location Facet is subject to interpretation and choice depending on the application. For documents residing on a file server or workstation, the Location Facet may comprise file paths or folders of documents. For tweets, on the other hand, it would make most sense to use the geolocation information of the tweet for this facet.

Time Facets. The way in which the Time Facet is implemented will also depend on the application. For tweets, it might be the time the tweet was posted, whereas an extracted publication date would be of most interest for scientific articles. For arbitrary files residing on a file server or laptop drive, it is sometimes useful to utilize the *created* or *last-modified* date for the Time Facet (which must be extracted from the file metadata).

Author Facets. As with the Location and Time facets, use of the Author facet will vary by application domain. In the case of news articles, the author might be extracted from the document content. For tweets, it will simply be the Twitter user account producing the tweet. For documents produced by Microsoft Office applications (*e.g.,* Microsoft Word, Excel, and Powerpoint), the author might be taken as the *Last-Author* field extracted from document metadata.

Throughout the remainder of this paper, we describe a concrete implementation of our proposed faceted classification system. First, we summarize our contributions.

---

[1]Faceted classification systems, in the context of computer systems, are also referred to as *faceted search*, *faceted navigation*, or *faceted browsing*.

[2]By "discoverable", we mean that the attributes for these facets are populated in an automated fashion.

[3]Sentiment analysis might also be included here.

1.2 Contributions

In this work, we explore the general problem of characterizing, navigating, and searching large collections of diverse documents. Our contributions are as follows:

- Based on our proposed faceted classification system, we present a fully implemented application for exploring highly heterogeneous document collections that span a wide array of file types, formats, and subject areas. The tool is designed to facilitate the identification of documents pertaining to military critical technologies, but can readily be used as a general-purpose tool for exploratory analyses of arbitrary text collections. Our system is based on *intelligently tagging* individual documents in a largely automated fashion.

- We propose the KERA (**K**eyword **E**xtraction for **R**eports and **A**rticles) algorithm, as one means by which informative *Topic Facets* can be constructed in an unsupervised and automated fashion.

- For scenarios where KERA may be inappropriate, we present supplemental strategies to characterize and locate information of interest based on supervised machine learning (*i.e.,* LinearSVM), unsupervised machine learning (*i.e.,* latent Dirichlet allocation or LDA), and natural language processing (*e.g.,* Named Entity Recognition or NER).

- We evaluate the application using two separate case studies at sites of deployment.

We begin a discussion of our work by providing an overview of our implemented system.

2. APPLICATION OVERVIEW

Sponsors of our research were in need of a tool to analyze arbitrary document collections in order to help analysts identify documents pertaining to military critical technologies. The term "military critical" here is defined by senior officials and subject matter experts. (We will sometimes simply use the term "critical" when referring to such technologies.) However, over time, they became interested in the general problem of characterizing, browsing, and searching through arbitrary and heterogeneous document collections (where the definition of critical will vary by application and even user). The document collections of interest here contain diverse sets of file types and formats that typically exist on file server and workstation drives. Examples include PDF articles and reports, Microsoft Office documents, plain text log files containing Web browser history, HTML documents, programming language source code files, and more. Our application is the end result of these objectives and interests.

Figure 1 shows a screenshot of one of the main interfaces. On the surface, it appears to be a standard search engine interface where users can type *ad hoc* search queries and view search results. However, the standard search functionality is enhanced (on the left in Figure 1) with numerous information facets based on the faceted classification system described in Section 1.1. The facets are populated by intelligently tagging each document in the collection along various dimensions. Documents can be viewed either in their original form or using a "Quick View" feature in which case the plain text is shown with highlighted terms (*e.g.,* discovered topic-representative keywords, search terms entered by user). Most (but not all) of the facets take the form of tag clouds. A *tag cloud* is a visualization of a set of words where the relative sizes of the words are determined by either features of the word or features of the entity represented by the word. In our work, the sizes of tags indicate the number of documents assigned the tag. Tag clouds are used both as a visualization and as an interface for faceted browsing of document collections, as the tags in the cloud can be used to filter and refine search results. Each facet can be expanded or collapsed by clicking the + and − symbols, as shown in Figure 1. The four facets viewable in Figure 1 (*e.g., Top Discovered Keywords*, *Topic Clusters*) all fall under the category of *Topic Facet* from our aforementioned faceted classification system. For the remaining facets, there is a one-to-one correspondence with the facet types described in Section 1.1. All facets (both those appearing in Figure 1 and not) are described at length later.

Tag Clouds as Lenses. Figure 2 shows a sample tag cloud displaying topic-representative keywords discovered using KERA, our unsupervised algorithm for keyterm extraction. This tag cloud facet can be viewed as a "lens" into document collections. The remaining facets may be viewed as controls used to point, zoom, and focus this "lens" to areas of high interest in the corpus.[4] For instance, when filtering the search results by folder using a *Location Facet*, the tag cloud shown in Figure 2 will dynamically re-generate to display the top discovered keywords of only the refined search results (*i.e.,* documents residing in the folder selected). In this way, users can quickly "triage" noisy document collections for information of interest (in some cases *even before opening and reading documents*). Although tag clouds have come under criticism in the past, our tag-based system is demonstrated to be surprisingly effective in locating critical information of interest buried deep within document collections. The key to achieving this success is constructing informative clouds free from noise. Throughout later sections, we describe how precisely we accomplish this. But first, we briefly describe the implementation of our system.

Implementation Details. The underlying engine driving our application is the Solr search server,[5] which natively supports text extraction, full text search, and faceted navigation. Documents in the Solr index are tagged using a series of supervised and unsupervised data mining algorithms, and it is these tags that power faceted browsing and tag clouds. All data mining algorithms are developed using the Python language and libraries including scikit-learn,[6] NLTK,[7] and Gensim topic modeling toolkit.[8] Moreover, all algorithms are implemented to process documents in a stream using both online and parallel processing. The graphical user interface is implemented using Flask[9] and AJAX-Solr.[10] Finally, communication between Python scripts and Solr is

---

[4]In actuality, each facet can play either the role of a "lens" or a "lens control."

[5]http://lucene.apache.org/solr/

[6]http://scikit-learn.org/stable/

[7]http://code.google.com/p/nltk/

[8]http://radimrehurek.com/gensim/

[9]http://flask.pocoo.org/

[10]http://github.com/evolvingweb/ajax-solr/wiki

Figure 1: A screenshot of our system. On the left, a rich set of information facets are provided for exploratory analysis. Only a subset (*i.e.,* the *Topic Facets*) is viewable in this screenshot. The application also provides standard search engine functionality powered by Solr, as shown.

handled using the pysolr library.[11] For the rest of this paper, we describe the concrete facets employed in our system and the analytics algorithms used to populate them.

Top Discovered Keywords –

association measures | collocation extraction | data set | extraction methods | feature selection | file format | file size | font size | harmonic mean | hidden topics | impression formation | information retrieval | international conference | keyphrase extraction | keyword extraction | lda | machine learning | mutual information | noun phrase | semantic web | tag clouds | topic detection | topic models | training data | web search

Figure 2: A tag cloud generated by the KERA algorithm for 64 documents used as references in this paper. (Only a subset were cited due to space constraints.) KERA is discussed in Section 3.1.

3. TOPIC FACETS

Topic Facets are intended to help discover and characterize the subject areas present in a document collection. Moreover, they allow users to better navigate the collection to find information of interest. The first *Topic Facet* we discuss is based on unsupervised keyterm extraction.

3.1 Automated Keyword Extraction

Our first approach to populating a *Topic Facet* is based on extracting topic-representative terms (*i.e.,* keywords) from documents (shown as *Top Discovered Keywords* in Figures 1 and 2). Here, we present the KERA algorithm (**K**eyword

Extraction for **R**eports and **A**rticles). KERA is an *unsupervised* algorithm to extract keywords from *individual* unstructured text documents (*i.e.,* it does not require an entire corpus like TF-IDF and other strategies). At its core, KERA is a *descriptive* model for keyword assignment. It is based on several key observations of human-assigned keywords[12] (especially those in scientific and technical publications):

- Many keywords assigned are multi-word terms (as opposed to unigrams).
- Most keywords are noun phrases.
- When a single-word term is used as a keyword, it often appears in the title (or abstract) or is a *proper* noun. Proper nouns often indicate an algorithm, system, or program being described or employed.
- If a keyword is composed of three or more terms, it can often (but not always) be split in two and maintain a high level of expressivity of the underlying topic described. For instance, "stream data mining" splits into "stream data" and "data mining," and "social network analysis" splits into "social network" and "network analysis."

Many of these observations have also been noted in other works (*e.g.,* [21]). The KERA algorithm is shown in Algorithm 1. We now describe its three main components: collocation extraction, part-of-speech filtering, and ranking.

Collocation Extraction. We first employ the use of *collocation extraction* to identify candidate key terms. A collocation is "an expression consisting of two or more words that corresponds to some conventional way of saying things." [16] We posit that it is these sets of words that are most likely to contain topic-representative phrases. Although it is possible to extract collocations of three or more terms, we find

---
Algorithm 1 KERA algorithm

Require: D, an unstructured text document
Require: K, the number of keywords to extract
1: # generate candidate keywords
2: $terms1 = extractCollocations(D)$
3: $terms2 = extractNounPhrases(D)$
4: $terms3 = extractProperNounUnigrams(D)$
5: $candidates = (terms1 \cap terms2) \cup terms3$
6: # rank candidates
7: **for all** $c \in candidates$ **do**
8:   **if** c is unigram **then**
9:     α = normalized frequency of term c in D
10: **else**
11:    α = normalized collocation score
12: **end if**
13: $\beta = 1 - \frac{\text{index of first occurrence of } c \text{ in } D}{\text{num. of words in D}}$
14: rank score of term $c = \frac{2 \cdot \alpha \cdot \beta}{\alpha + \beta}$
15: **end for**
16: # optionally prune based on domain-specific criteria
17: # $candidates = prune(candidates)$
18: return top K candidates based on rank score
---

that such phrases do not lend themselves to aggregation (*e.g.,* for use in tag clouds). At the same time, we find that one word terms are not expressive enough for users to discern the topics of documents. Thus, we extract only collocated *bigrams* (*i.e.,* two-word expressions). Although our system supports multiple collocation extraction strategies including the log-likelihood ratio test [5] and Pointwise Mutual Information (PMI) [16], we currently use the log-likelihood ratio exclusively, as we find it performs best with respect to *Topic Facets*. Using the log-likelihood ratio test, the collocation score for a bigram of words w_1 and w_2 is $2 \sum_{ij} n_{ij} \log \frac{n_{ij}}{m_{ij}}$, where n_{ij} are the observed frequencies of the bigram from the contingency table for w_1 and w_2 and m_{ij} are the expected frequencies assuming that the bigram is independent [5, 16].

Part-of-Speech Filtering. As mentioned, the most expressive keywords are typically noun phrases. Thus, we filter the set of collocations by removing terms that do not match the pattern (ADJECTIVE)*(NOUN)+. If the extracted phrases are greater than two terms, we begin truncating from the left until we are left with a bigram. Such filtering also helps to remove bogus words sometimes introduced by the text extraction process for non-plain-text document formats. To this filtered set, we *add* extracted unigrams that are proper nouns, as we find such terms can be critical to the topic of documents. This is especially true of government, scientific, and technical publications, as proper nouns often refer to a system, algorithm, program, or initiative being described.

Ranking Keywords. Finally, we rank the extracted terms, as shown in Algorithm 1 and return the top K candidates.[13] Our ranking methodology takes into account both the position of terms within a document and the collocation score (or term frequency). The final score is taken as the harmonic mean of these metrics. Prior to returning the final set, one might optionally prune the candidates based on domain-

---
[13]Currently, we set $K = 5$ or $K = 10$ for KERA.

specific criteria. For instance, in our case, the set of proper noun unigrams may be pruned to only contain those unigrams that are upper-case, since it is those terms that often signify important technical systems and programs.[14]

Comparison to Other Approaches. Development of KERA was motivated by the fact that existing algorithms did not meet one or more of our needs. For instance, a number of the existing approaches are either supervised, require an entire corpus, or both. Such characteristics are unacceptable, as supervised approaches are labor-intensive and corpus-based methods (such as those like TF-IDF that use inverse document frequency) may undervalue terms associated with prevalent topics. TextRank [17] and RAKE [21] are two methods that are both unsupervised and operate on *individual* documents. Unfortunately, although both methods can perform reasonably well when supplied only paper abstracts, they sometimes perform less well on longer, messier, and more realistic document structures.[15] For instance, Table 1 shows the keywords extracted for this very paper. Note that the keywords extracted by KERA are qualitatively superior to TextRank [17] and RAKE [21].

Method	Top 5 Extracted Keywords for This Paper
KERA	"document collections", "KERA", "tag clouds", "machine learning", "LDA"
RAKE	"plain text log files containing web browser history", "files spanned numerous file formats including MS office", "dat file containing web search history", "provides standard search engine functionality powered", "faceted browsing framework yields significant advantages"
TextRank	"document collections", "system", "document", "tagging", "faceted"

Table 1: Top 5 extracted keywords for this paper ordered by rank assigned by each method. KERA-generated terms are most expressive and well-suited to tag cloud aggregation. TextRank is also too slow for our needs.

3.2 Topic Modeling and Clustering

A second *Topic Facet* we employ is based on the concept of topic clusters. Topic modeling and clustering algorithms segment documents into groups, where the intent is for each group to consist of documents pertaining to a particular topic or theme. Whereas many clustering algorithms produce "hard" clusters or disjoint sets of documents, topic models produce "soft" or overlapping clusters. Topic models and clustering strategies may also tag clusters with topic-representative words. In topic models like latent Dirichlet allocation or LDA [1], topics are modeled as word probability distributions, and these tags are simply the most probable words in a distribution. Our application supports *multiple* approaches to topic clustering including LDA [1], Hierarchical Dirichlet Process (HDP) [23], Latent Semantic Indexing (LSI) [15], and K-Means [7]. All approaches are provided by

---
[14]Other possible variations include discarding candidates when proper noun unigrams also appear as part of extracted bigrams, removal of unigrams that do not first appear until later in the document, significance testing to filter the set of collocations, and setting α always as normalized frequency.
[15]We base these statements on the TextRank implementation available at `http://search.cpan.org/~kubina/Text-Categorize-Textrank-0.51/` and the RAKE implementation at `http://github.com/aneesha/RAKE`. Full experimental results are omitted due to space limitations.

the machine learning libraries mentioned in Section 2. For the current deployment, we employ LDA exclusively. Documents are assigned to a topic only if the topic proportion assigned by LDA is greater than 0.3, and documents are tagged using the top 10 LDA-derived topic tags. LDA requires the number of topics, K, as input, and we currently set this heuristically based on the size of the document collection. However, in the future, we plan to migrate to HDP, which is a non-parametric approach to topic modeling [23]. The facet populated by LDA is labeled "Topic Clusters" and appears as a menu showing the list of discovered topics. These topic clusters are labeled by LDA-derived tags and ordered by the topic ranking methodology described in [26].

3.3 Document Classifier Facets

All *Topic Facets* discussed thus far (including topic models) are focused on identifying trends and hotspots within the topic collection. That is, they are not well-suited to finding "needles in haystacks." A document pertaining to a lone topic of high interest to a particular user may not be identifiable in the presence of large topic clusters displayed in a tag cloud or other interface. To address this, we supplement the facets populated by KERA and LDA with additional tag cloud facets populated with *supervised* document classification. We have previously reported our work on supervised machine learning for critical technologies in [14] . Thus, we only include brief and sparse descriptions here. For more information on the development of document classifiers in this domain, please see [14].

3.3.1 Military Critical Technology Finder

The facet labeled **Military Critical Technology Finder** in Figure 1 is populated using a set of binary supervised machine learning classifiers. Each binary classifier is trained to identify documents pertaining to a particular critical technology, and each tag in the cloud represents the positive class of a classifier. For any individual document, if no binary classifier categorizes the document as positive, then the document is assigned the tag "other", which also appears in the cloud. We use LinearSVM as our main learning algorithm for all classifiers. Constructing training sets for these classifiers poses a number of challenges. For instance, when training these binary classifiers for arbitrary file collections (*e.g.,* a workstation hard drive), the negative class becomes *highly* heterogeneous. If this heterogeneity is not represented or otherwise addressed in the training set, performance can degrade. In addition, documents pertaining to critical technologies can sometimes compromise a very small minority of all possible files encountered. This is known as the class imbalance problem and can also cause performance to suffer due to bias. To address these and other problems, we employ heavy use of *active learning* in a two step sampling procedure [14]. We first employ active learning strategies (*e.g.,* minimum marginal hyperplane) to sample only the most informative of negative examples for the initial training set (which helps address heterogeneity). We, then, balance the training set by further sub-sampling this initial training set to produce the final training set.

3.3.2 Report Type Filter

Using a very similar methodology to the one described in the previous section, we develop an additional classifier to categorize documents based on report type. That is, doc-uments are categorized into one of four categories: Technical Information (*e.g.,* a research paper), Test Information (*e.g.,* a test plan for a system), Programmatic Information (*e.g.,* details of a program for development of a system), and Other (*i.e.,* everything else).

4. MENTION FACETS

Users sometimes may be interested in locating documents not by topic but by mentions of particular entities, terms, or expressions of interest (*e.g.,* IP addresses). To address this, we employ the use of a *Mention Facet*, which allows users to upload a plain text file containing expressions of interest. These expressions can currently take the form of simple lists of terms, gazetteers (*i.e.,* entity dictionaries), or regular expressions for patterns of interest (*e.g.,* a social security number). The results are displayed as either a tag cloud or menu, where the items are either explicit terms with matches in the document collection or high-level categories described by expressions (*e.g.,* tagging documents containing social security numbers with "PII"). Recall that our current research sponsor is specifically interested identifying military critical information. Thus, for the first deployment of our application, we populate the *Mention Facet* in the following manner. We take the training sets used for our binary classifiers described in Section 3.3.1 and extract the top 25 most discriminative terms based on *information gain* [15]. The *entropy* H of a set of labeled documents D measures impurity as follows: $\text{H}(D) = -p^+ \log_2(p^+) - p^- \log_2(p^-)$, where p^+ and p^- are the proportions of positive and negative documents in D, respectively.[16] The information gain IG of a word w in training set D, then, is the expected entropy reduction due to segmenting on w: $\text{IG}(w, D) = \text{H}(D) - \frac{|D^w|}{|D|}\text{H}(D^w) - \frac{|\overline{D^w}|}{|D|}\text{H}(\overline{D^w})$, where D^w is the set of documents in D containing word w. Thus, words with the highest information gain in a training set are expected to be the most discriminative. (Although the *Mention Facet* can be used for many purposes, populating the facet in this fashion, in a sense, transforms it into yet another kind of *Topic Facet*.) We have also used KERA to populate the *Mention Facet* directly from only the positive training documents. Finally, we supplement this list of discriminative terms with a set of markings for sensitive documents (*e.g.,* "For Official Use Only", "FOUO").

5. FORMAT, LOCATION, TIME, AND AUTHOR FACETS

Our final set of facets are populated through direct extraction from document metadata. The *Format Facet* is populated by tagging documents based on file type (*e.g.,* pdf, doc, ppt, txt) and is labeled "Top File Types." The *Location Facet* (labeled "Top Folders") is populated by tagging each document with the directories in its file path. The *Time Facet* (labeled "Date" in our application) is populated by extracting the *Last-Modified* time from documents. Finally, the *Author Facet* is populated using the *Last-Author* or *Author* name (when available). The *Location Facet* is displayed as a menu listing the most populous folders, and the *Time Facet* is displayed as a calendar widget. All other facets are displayed as tag clouds. (Note that none of these facets are viewable in Figure 1.)

---

[16]Note that $\log_2(0)$ is taken to be 0.

6. CASE STUDIES

We conduct a series of case studies at the deployment sites using a prototype of our application undergoing field testing. Motivated by the recent position paper "Machine Learning That Matters" by Wagstaff [25], we focus on *external* validation of our application by assessing time saved and insights gained in collaboration with domain experts. Although our system can be used for many purposes, we focus our evaluation on the current application of interest to our sponsors — locating information pertaining to military critical technologies within heterogeneous document collections. To locate such information, analysts at the sponsoring agency currently use simple keyword searches exclusively. Thus, we compare our new approaches to this existing approach. Since our system employs the use of multiple approaches to locate and discover information, we also draw comparisons *among* our new approaches. For reasons of sensitivity, we cannot reveal the deployment sites, the sponsoring agency, or technical subjects of interest to the agency. Thus, we redact information as necessary.

6.1 Case Study 1: Search

Search here involves the task of finding information pertaining to a particular military critical technology within a document collection. We consider a particular technology of high interest to our sponsors and assess how well the *supervised* approaches in our application are able to locate this critical information. We refer to this technology simply as *Technology-X*. A case was provided to us containing 30,128 files acquired from workstation hard drives of roughly 11 users. The files spanned numerous file formats including Microsoft Office, HTML, PDF, and plain text. Analysts confirmed to us that the case was positive. That is, it was manually verified previously to contain information about *Technology-X* but not searched thoroughly. The files were spread across multiple media (*e.g.,* external USB hard drives, SATA drives, DVDs). We built machine learning classifiers and a custom mention search for *Technology-X*, as described in Sections 3.3 and 4. Upon loading and indexing the case into our application, we evaluated these approaches and compared results to those obtained from a manual review of the case by two analysts using their existing methodology (*i.e., ad hoc* keyword searches only). Results are shown in Figure 3 as a Venn diagram.
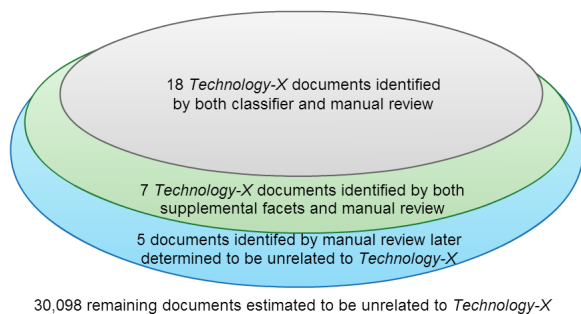

18 *Technology-X* documents identified by both classifier and manual review

7 *Technology-X* documents identified by both supplemental facets and manual review

5 documents identifed by manual review later determined to be unrelated to *Technology-X*

30,098 remaining documents estimated to be unrelated to *Technology-X*

Figure 3: Venn diagram of search results. The top (innermost) oval shows documents identified by classifier. The middle oval shows documents identified by other facets. The bottom (outermost) oval shows documents identified via a manual review by two analysts.

As shown, both the classifier and the two analysts identified 18 documents as pertaining to *Technology-X*. The analysts also identified twelve additional files. Upon review by subject matter expert (SME), only seven of the twelve files were related to *Technology-X*, whereas the classifier achieved perfect precision. Of these seven false negatives, one was a figure with no accompanying text and a second was a 5 sentence email that was deemed critical by the SME. The most striking result, however, is the time savings achieved. The two analysts took roughly 7 hours (or 14 person-hours) to locate *Technology-X* documents. By contrast, the classifier identified 18 of the 25 *in mere seconds*. The remaining files (*i.e.,* all the seven false negatives) were located in *less than 30 minutes* using the *Mention Search*, *Report Type Filter*, and *Top Folders* facets in our application. We attribute most false negatives committed by the classifier to the fact that, due to political complications, the positive examples available to us were limited (only 51 examples were used). Given this and the breadth and depth of military critical technology information, *unsupervised* topic discovery is of high importance to this domain. We discuss this next.

6.2 Case Study 2: Discovery

Discovery involves *browsing* document collections and allows users to locate information for which they did not even know to look. A framework to facilitate discovery can clearly facilitate a *search* for something specific, as well. Due to logistical and policy-related issues, we were not able to evaluate discovery on the case described in Section 6.1. Instead, we were provided a new case to evaluate, which contained 39,515 files. Unlike the case from Section 6.1, we did *not* have any approximation of ground truth, as the case had not been formally reviewed. Here, we assess the knowledge discovered and summarize lessons learned from execution of our application on this case.

Identified Critical Topics. Table 2 shows the two topics pertaining to military critical technologies discovered by our application (referred to as *Technology-Y* and *Technology-Z*. Of course, numerous non-critical topics within the document collection were also discovered (some of which were of a personal, non-work-related nature). As for the critical topics, there were 89 documents found pertaining to *Technology-Z* and 232 documents pertaining to *Technology-Y* (including duplicate files). Through a subsequent exhaustive manual review of the case, we estimate that no additional information on military critical technologies of interest was present on this case. Using our facet-based system, most documents for these two critical topics were identified in less than an hour (and in some cases only minutes). By contrast, domain experts informed us that cases of this size typically require hours or days of analysis to produce similar results, which is consistent with our experience during the manual review. Several facets were identified as highly effective in identifying these topics when used in combination with each other. We discuss these next.

Topic	Docs	Facets Employed
Technology-Y	232	**Method A:** Topic Clusters (LDA)→KERA **Method B:** Report Type Filter →KERA
Technology-Z	89	**Method A:** Topic Clusters (LDA)→KERA **Method B:** Report Type Filter →KERA **Method C:** Top Folders →KERA

Table 2: Critical topics found and effective usage patterns.

Effective Usage Patterns Discovered. The third column of Table 2 displays the facet combinations that were found to be most effective in identifying critical documents. The combination labeled as **Method A** indicates that the "Topic Cluster" facet populated by LDA was first explored and used to filter the search results. The "Top Discovered Keywords" facet (populated by KERA), then, was used to identify document sets related to the critical technology. The combination labeled as **Method B** indicates that the "Report Type Filter" was used to locate documents pertaining to *Technical Information* followed again by the "Top Discovered Keywords" facet. Finally, for **Method C**, the "Top Folders" facet (*i.e.,* our *Location Facet*) was used to filter the search results, with the KERA-populated tag cloud being used to quickly assess documents within folders. Notice that *multiple* facet combinations often exist to locate the same set of critical documents. This highlights a significant advantage to our multi-faceted system: topics are more likely to be discovered by users when more paths lead towards them. As shown in the table, **Method A** and **Method B** were employed heavily to find both *Technology-Y* and *Technology-Z*. **Method C** was only used for *Technology-Z*, not *Technology-Y*. Since the files for *Technology-Y* were scattered across many directories, the "Top Folders" facet was not as useful. As can be seen, the "Top Discovered Keywords" facet populated by KERA played an indispensable role in all three methods, as it allowed for quick exploration and assessment of the document collection.[17] It was particularly useful as a complement to the "Topic Cluster" facet, as we now explain.

Using KERA as a Cluster Labeling Strategy. One of the issues with topic models like LDA is that the terms (or tags) they assign to topics are often not very expressive of the topic. In other words, in practice, it is quite difficult for humans to go directly from LDA-derived tags to a thematic label for the cluster *without* reading documents in the cluster. This has been recognized in other works on deployed applications based on topic models (*e.g.,* see [26]), and we found this to be the case in our evaluation, as well. However, from the effective usage patterns observed previously, we observed that the "Top Discovered Keywords" facet populated by KERA is a highly effective way to quickly determine the overall subject matter of a topic cluster (or even a folder). We cannot illustrate this on topics related to military critical technologies due to their sensitive nature. Figure 4, however, shows tags produced for a non-critical document cluster (extracted from documents residing on the first author's laptop). Although the tag cloud generated by KERA is not quite a thematic label for the cluster, it is significantly more expressive than the tags assigned by a typical LDA implementation. This, then, illustrates yet another way in which tag cloud facets are useful "lenses" into document collections, as described in Section 2.

LDA Performance on Critical Technologies. Table 3 shows the precision and recall with respect to the LDA clustering. A true positive is defined as placement into an appropriately labeled cluster having a majority of the doc-

association rules | automatic keyword | candidate keywords | computational linguistics | controlled vocabulary | data mining | digital computer | frequent terms | information retrieval | keyphrase extraction | keyword extraction | machine learning | natural language | noun phrase | opinion mining | semantic desktop | semantic web | sentiment analysis | single document | submit question | term extraction | term frequency | text mining | topic extraction | training data

(a) KERA-generated Tag Cloud

based, system, semantic, information, paper extraction, question, words, set, results

(b) Top LDA-Derived Tags

Figure 4: Tags produced for an LDA-generated cluster of 99 documents about *text analytics*: (a) KERA-generated tag cloud (b) top-ranked tags derived from a typical LDA implementation. KERA-generated tags tend to be significantly more expressive of the underlying topic, thereby, facilitating exploratory analysis.

uments pertaining to the same militarily critical topic (as judged by a SME). Critical documents placed into clusters with largely non-critical and possibly unrelated documents are considered false negatives. Non-critical documents appearing in a cluster of largely critical documents are considered false positives. Note the low precision for *Technology-Z* resulting from 84 of the 89 critical documents being placed into a larger cluster of non-critical documents. Since these non-critical documents were indirectly related to the topic covered by *Technology-Z*, LDA was unable to distinguish them from the truly critical documents. All 89 of these critical documents, however, were ultimately located with help from other facets such as *Top Folders* and *Top Discovered Keywords*. These results (combined with an intolerance to false negatives by users in this domain) justify our decision to employ *multiple* facets — as opposed to relying only on topic models, which some other works have done (*e.g.,* [26]).

Critical Technology	Precision	Recall	F-Score
Technology-Y	0.89	0.98	0.93
Technology-Z	**0.56**	0.94	0.70

Table 3: LDA performance with respect to military critical technologies. Low precision (in bold) and general intolerance to false negatives in this domain motivate our use of multiple information facets.

Issues Requiring Future Investigation. We conclude our discussion of this case study by noting two issues observed during our evaluation. The first relates to setting the number of topics, K, in LDA. Most works, including ours, set this value in a largely *ad hoc* fashion. Although there are heuristics and rules-of-thumb that have been proposed (*e.g.,* [3]), most machine learning practitioners acknowledge that the choice of K is "more art than science."[18] Guessing the correct value of K is particularly difficult for hetero-

---

[17]In more recent tests, the *Mention Facet*, when used to locate documents with sensitive markings, also was found to be useful in combination with all methods listed in Table 2.

[18]http://cwiki.apache.org/MAHOUT/latent-dirichlet-allocation.html

geneous document collections, as K can be severely underestimated. Moreover, an incorrect setting of K can have detrimental effects on the results. We have personally found this to be true in our evaluations. One approach to addressing this is to employ the use of newer non-parametric topic models like HDP [23]. We plan to explore such methods in the future to address these issues. A second issue relates to KERA. Although bigrams are appropriate and well-suited for automated tag cloud generation, in some cases, they can produce sub-optimal results (*e.g.,* extracting "Dirichlet allocation" and not "latent Dirichlet allocation"). Some recent approaches to word segmentation based on probabilistic models can potentially be exploited for better keyterm extraction [6]. This, then, is another area for potential future exploration.

7. RELATED WORK

Given the diverse set of facets employed by our application, several different lines of related work exist. We briefly describe these areas here.

Characterizing Large Document Collections. There are several works describing text analytic systems designed to characterized large text corpora (*e.g.,* [4, 10, 26]). Most systems focus on a particular document type (*e.g.,* tweets, emails), whereas as our system is designed with heterogeneous document collections in mind.

Topic Modeling, Clustering, and Categorization. Many text analytic systems perform topic analysis through use of topic models (*e.g.,* LDA [1], HDP [23]) or clustering algorithms like K-Means [1,7], which are both unsupervised. Supervised text classification approaches are also sometimes employed [15]. Our objective in this work is to bring to bear *multiple* approaches for topic analysis. As we have shown, using multiple approaches in concert with each other through a faceted browsing framework yields significant advantages.

Keyphrase Extraction. Several works describe algorithms to extract keywords and keyphrases from documents. Some approaches are supervised or require an entire corpus as input (*e.g.,* [2,27]), which, as described previously, is not suitable for our purposes. TextRank [17] and RAKE [21] are two approaches that are purely unsupervised and operate on individual documents. However, as we have shown, they do not appear well-suited to automated tag cloud generation. A related area of research is collocation extraction (*e.g.,* [5,16,19]), which we exploit in the KERA algorithm.

Tag Cloud Research. Numerous works leverage tag clouds for both faceted navigation and corpus visualization (*e.g.,* [9, 11,28]). The overwhelming majority of this work focuses on manually-generated tags (*e.g.,* social-tagging systems) as opposed to automated generation of tags, which is one of the foci of our work.

8. CONCLUSION

In this paper, we have proposed a demonstrably effective system for exploratory analysis of arbitrary document collections. Our system, based on multiple *information facets*, is designed to address a major capability gap within the U.S. federal government: investigative analysis of highly heterogeneous document collections. We have presented a concrete implementation of this multi-faceted system that aids users in identifying information pertaining to military

critical technologies embedded within large and arbitrary document collections. A prototype of our application was successfully deployed in May 2013. In the future, we plan to extend the tool in numerous ways including sentence-based summaries of topics and visualizations of topic clusters.

9. REFERENCES

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(4-5):993–1022, Mar. 2003.

[2] K. K. Bun and M. Ishizuka. Topic extraction from news archive using TF*PDF algorithm. In *WISE 2002*.

[3] F. Can and E. A. Ozkarahan. Concepts and effectiveness of the cover-coefficient-based clustering methodology for text databases. *ACM Trans. Database Syst.*, 15(4):483–517, Dec. 1990.

[4] G. Cselle, K. Albrecht, and R. Wattenhofer. BuzzTrack: topic detection and tracking in email. In *IUI '07*.

[5] T. Dunning. Accurate methods for the statistics of surprise and coincidence. *Comput. Linguist.*, 19(1):61–74, Mar. 1993.

[6] S. Goldwater, T. L. Griffiths, and M. Johnson. A Bayesian framework for word segmentation: Exploring the effects of context. *Cognition*, 112(1):21–54, July 2009.

[7] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, corrected edition, July 2003.

[8] Q. He, B. Chen, J. Pei, B. Qiu, P. Mitra, and L. Giles. Detecting topic evolution in scientific literature: how can citations help? In *CIKM '09*.

[9] K. Knautz, S. Soubusta, and W. G. Stock. Tag Clusters as Information Retrieval Interfaces. In *HICSS '10*.

[10] S. Kumar, F. Morstatter, G. Marshall, H. Liu, and U. Nambiar. Navigating information facets on twitter (NIF-T). In *KDD '12*.

[11] B. Y. L. Kuo, T. Hentrich, B. M. Good, and M. D. Wilkinson. Tag clouds for summarizing web search results. In *WWW '07*.

[12] S. C. Lee, W. McFadden, and P. Bajcsy. Text, Image and Vector Graphics Based Appraisal of Contemporary Documents. In *ICMLA '08*.

[13] D. J. MacEachin. *The tradecraft of analysis: Challenge and change in the CIA (Working Group on Intelligence Reform papers)*. Consortium for the Study of Intelligence.

[14] A. S. Maiya, F. Loaiza-Lemos, and R. M. Rolfe. Supervised learning in the wild: Text classification for critical technologies. In *MILCOM '12*.

[15] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 1 edition, July 2008.

[16] C. D. Manning and H. Schütze. *Foundations of statistical natural language processing*. MIT Press, Cambridge, MA, USA, 1999.

[17] R. Mihalcea and P. Tarau. TextRank: Bringing Order into Texts. In *EMNLP '04*.

[18] M. Pearson. The Petraeus affair: A lot more than sex - CNN.com. Nov. 2012.

[19] P. Pecina. An extensive empirical study of collocation extraction methods. In *ACLstudent '05*.

[20] S. R. Ranganathan. *Elements of Library Classification*, volume 8. Asia Pub. House, 1962.

[21] S. Rose, D. Engel, N. Cramer, and W. Cowley. Automatic Keyword Extraction from Individual Documents. *Text Mining: Applications and Theory.*, pages 1–20, 2010.

[22] J. Tang, B. Wang, Y. Yang, P. Hu, Y. Zhao, X. Yan, B. Gao, M. Huang, P. Xu, W. Li, and A. K. Usadi. PatentMiner: topic-driven patent analysis and mining. In *KDD '12*.

[23] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet Processes. *J of the Am. Statistical Assoc.*, 101(476):1566–1581, Dec. 2006.

[24] D. Tunkelang. *Faceted Search (Synthesis Lectures on Information Concepts, Retrieval, and Services)*. Morgan and Claypool Publishers, June 2009.

[25] K. Wagstaff. Machine Learning that Matters. In *ICML '12*.

[26] F. Wei, S. Liu, Y. Song, S. Pan, M. X. Zhou, W. Qian, L. Shi, L. Tan, and Q. Zhang. TIARA: a visual exploratory text analytic system. In *KDD '10*.

[27] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, and C. G. Nevill-Manning. KEA: Practical Automatic Keyphrase Extraction, Feb. 1999.

[28] A. Zubiaga, A. P. Garcia-Plaza, V. Fresno, and R. Martinez. Content-Based Clustering for Tag Cloud Visualization. In *SASONAM '09*.