MUCK: A toolkit for extracting and visualizing semantic dimensions of large text collections

Rebecca Weiss
Stanford University
Stanford, CA, 94305
rjweiss@stanford.edu

Abstract

Users with large text collections are often faced with one of two problems; either they wish to retrieve a semantically-relevant subset of data from the collection for further scrutiny (needle-in-a-haystack) or they wish to glean a high-level understanding of how a subset compares to the parent corpus in the context of aforementioned semantic dimensions (forest-for-the-trees). In this paper, I describe MUCK, an open-source toolkit that addresses both of these problems through a distributed text processing engine with an interactive visualization interface.

1 Introduction

As gathering large text collections grows increasingly feasible for non-technical users, individuals such as journalists, marketing/communications analysts, and social scientists are accumulating vast quantities of documents in order to address key strategy or research questions. But these groups often lack the technical skills to work with large text collections, in that the conventional approaches they employ (content analysis and individual document scrutiny) are not suitable for the scale of the data they have gathered. Thus, users require tools with the capability to filter out irrelevant documents while drilling-down to the documents that they are most interested in investigating with closer scrutiny. Furthermore, they require the capability to then evaluate their subset in context, as the contrast in attributes between their subset and the full corpora can often address many relevant questions.

This paper introduces a work-in-progress: the development of a toolkit that aids non-technical users of large text collections by combining semantic search and semantic visualization methods. The purpose of this toolkit is two-fold: first, to ease the technical burden of working with large-scale text collections by leveraging semantic information for the purposes of filtering a large collection of text down to the select sample documents that matter most to the user; second, to allow the user to visually explore semantic attributes of their subset in comparison to the rest of the text collection.

Thus, this toolkit comprises two components:

1. a distributed text processing engine that decreases the cost of annotating massive quantities of text data for natural language information
2. an interactive visualization interface that enables exploration of the collection along semantic dimensions, which then affords subsequent document selection and subset-to-corpora comparison

The text processing engine is extensible, enabling the future development of plug-ins to allow for tasks beyond the included natural language processing tasks, such that future users can embed any sentence- or document-level task to their processing pipeline. The visualization interface is built upon search engine technologies to decrease search result latency to user requests, enabling a high level of interactivity.

2 Related work

The common theme of existing semantic search and semantic visualization methods is to enable the user to gain greater, meaningful insight into the structure of their document collections through the use of transparent, trustworthy methods (Chuang et al., 2012; Ramage et al., 2009). The desired insight can change depending on the intended task.
For some applications, users are understood to have a need to find a smaller, relevant subset of articles (or even a single article) in a vast collection of documents, which we can refer to as a needle-in-a-haystack problem. For others, users simply require the ability to gain a broad but descriptive summary of a semantic concept that describes these text data, which we can refer to as a forest-for-the-trees problem.

For example, marketers and social scientists often study news data, as the news constitute a vitally important source of information that guide the agendas of marketing strategy and inform many theories underlying social behavior. However, their interests are answered at the level of sentences or documents that contain the concepts or entities that they care about. This need is often not met through simple text querying, which can return too many or too few relevant documents and sentences. This is an example of a needle-in-a-haystack problem, which has been previously addressed through the application of semantic search (Guha et al., 2003). Much of the literature on semantic search, in which semantic information such as named entity, semantic web data, or simple document categories are added to the individual-level results of a simple query in order to bolster the relevance of resulting query hits. This type of information has proven to be useful in filtering out irrelevant content for a wide array of information retrieval tasks (Blanco et al., 2011; Pound et al., 2010; Hearst, 1999b; Hearst, 1999a; Liu et al., 2009; Odijk et al., 2012).

Remaining in the same narrative, once a subset of relevant documents has been created, these users may wish to see how the semantic characteristics of their subset contrast to the parent collection from which it was drawn. A marketer may have a desire to see how the tone of coverage in news related to their client’s brand compares to the news coverage of other brands of a similar type. A social scientist may be interested to see if one news organization covers more politicians than other news organizations. This is an example of a forest-for-the-trees problem. This type of problem has been addressed through the application of semantic visualization, which can be useful for trend analysis and anomaly detection in text corpora (Fisher et al., 2008; Chase et al., 1998; Hearst and Karadi, 1997; Hearst, 1995; Ando et al., 2000).

The toolkit outlined in this paper leverages both of these techniques in order to facilitate the user’s ability to gain meaningful insight into various semantic attributes of their text collection while also retrieving semantically relevant documents.

3 Overview of System From User Perspective

The ordering of a user’s experience with this toolkit is as follows:

1. Users begin with a collection of unstructured text documents, which must be made available to the system (e.g., on a local or network drive or as a list of URLs for remote content)
2. Users specify the types of semantic detail relevant to their analysis (named entities, sentiment, etc.), and documents are then parsed, annotated, and indexed.
3. Users interact with the visualization in order to create the subset of documents or sentences they are interested in according to semantic dimensions of relevance
4. Once a view has been adequately configured using the visual feedback, users are able to retrieve the documents or sentences referenced in the visualization from the document store

Items 2 and 3 are further elaborated in the sections on the backend and frontend.

4 Backend

The distributed processing engine is driven by a task planner, which is a framework for chaining per-document tasks. As diagrammed in figure 1, the system creates and distributes text processing tasks needed to satisfy the user’s level of semantic interest according to the dependencies between the various integrated third-party text processing libraries. Additionally, this system does not possess dependencies on additional third-party large-scale processing frameworks or message queueing systems, which makes this toolkit useful for relatively large (i.e. millions of documents) collections as it does not require configuration of other technologies beyond maintaining a document store 2 and a search index.

2http://www.mongodb.com
**Task planner and resolver system** The semantic information extraction process occurs via defining a series of tasks for each document. This instantiates a virtual per-document queues of processing tasks. These queues are maintained by a task planner and resolver, which handles all of the distribution of processing tasks through the use of local or cloud resources. This processing model enables non-technical users to describe a computationally-intensive, per-document processing pipeline without having to perform any technical configuration beyond specifying the level of processing detail output desired.

**NLP task** Currently, this system only incorporates the full Stanford CoreNLP pipeline, which processes each document into its (likely) constituent sentences and tokens and annotates each sentence and token for named entities, parts-of-speech, dependency relations, and sentiment (Toutanova et al., 2003; Finkel et al., 2005; De Marneffe et al., 2006; Raghunathan et al., 2010; Lee et al., 2011; Lee et al., 2013; Recasens et al., 2013; Socher et al., 2013). This extraction process is extensible, meaning that future tasks can be defined and included in the processing queue in the order determined by the dependencies of the new processing technology. Additional tasks at the sentence- or document-level, such as simple text classification using the Stanford Classifier (Manning and Klein, 2003), are included in the development roadmap.

### 5 Frontend

A semantic dimension of interest is mapped to a dimension of the screen as a context pane, as diagrammed in figure 2. Corpora-level summaries for each dimension are provided within each context pane for each semantic category, whereas the subset that the user interactively builds is visualized in the focus pane of the screen. By brushing each of semantic dimensions, the user can drill-down to relevant data while also maintaining an understanding of the semantic contrast between their subset and the parent corpus.

This visualization design constitutes a *multiple-view* system (Wang Baldonado et al., 2000), where a single conceptual entity can be viewed from several perspectives. In this case, the semantic concepts extracted from the data can be portrayed in several ways. This system maps semantic dimensions to visualization components using the following interaction techniques:

**Navigational slaving** Users must first make an initial selection for data by querying for a specific item of interest; a general text query (ideal for phrase matching), a named entity, or even an entity that served in a specific dependency relation (such as the dependent of an *nsubj* relation). This selection propagates through the remaining components of the interface, such that the remaining semantic dimensions are manipulated in the context of the original query.

**Focus + Context** Users can increase their understanding of the subset by zooming into a relevant
selection in a semantic dimension (e.g. time).

**Brushing** Users can further restrict their subset by highlighting categories or ranges of interest in semantic dimensions (e.g. document sources, types of named entities). Brushing technique is determined by whether the semantic concept is categorical or continuous.

**Filtering** The brushing and context panes serve as filters, which restrict the visualized subset to only documents containing the intersection of all brushed characteristics.

This visualization design is enabled through the use of a distributed search engine\(^5\), which enables the previously defined interactivity through three behaviors:

**Filters** Search engines enable the restriction of query results according to whether a query matches the parameters of a filter, such as whether a field contains text of a specific pattern.

**Facets** Search engines also can return subsets of documents structured along a dimension of interest, such as by document source types (if such information was originally included in the index).

**Aggregations** Aggregations allow for bucketing of relevant data and metrics to be calculated per bucket. This allows the swift retrieval of documents in a variety of structures, providing the hierarchical representation required for visualizing a subset along multiple semantic dimensions defined above.

**Nesting** All of these capabilities can be stacked upon each other, allowing for the multiple view system described above.

The visualization components are highly interactive, since the application is built upon a two-way binding design paradigm\(^6\) between the DOM and the RESTful API of the index (Bostock et al., 2011).

### 6 Discussion and future work

This paper presents a work-in-progress on the development of a system that enables the extraction and visualization of large text collections along semantic dimensions. This system is open-source and extensible, so that additional per-document processing tasks for future semantic extraction procedures can be easily distributed. Additionally, this system does not possess requirements beyond maintaining a document store and a search index.

\(^5\)http://www.elasticsearch.com

\(^6\)http://www.angularjs.org
References


