TextTile: An Interactive Visualization Tool for Seamless Exploratory Analysis of Structured Data and Unstructured Text

Cristian Felix, Anshul Vikram Pandey, and Enrico Bertini, *Member, IEEE*

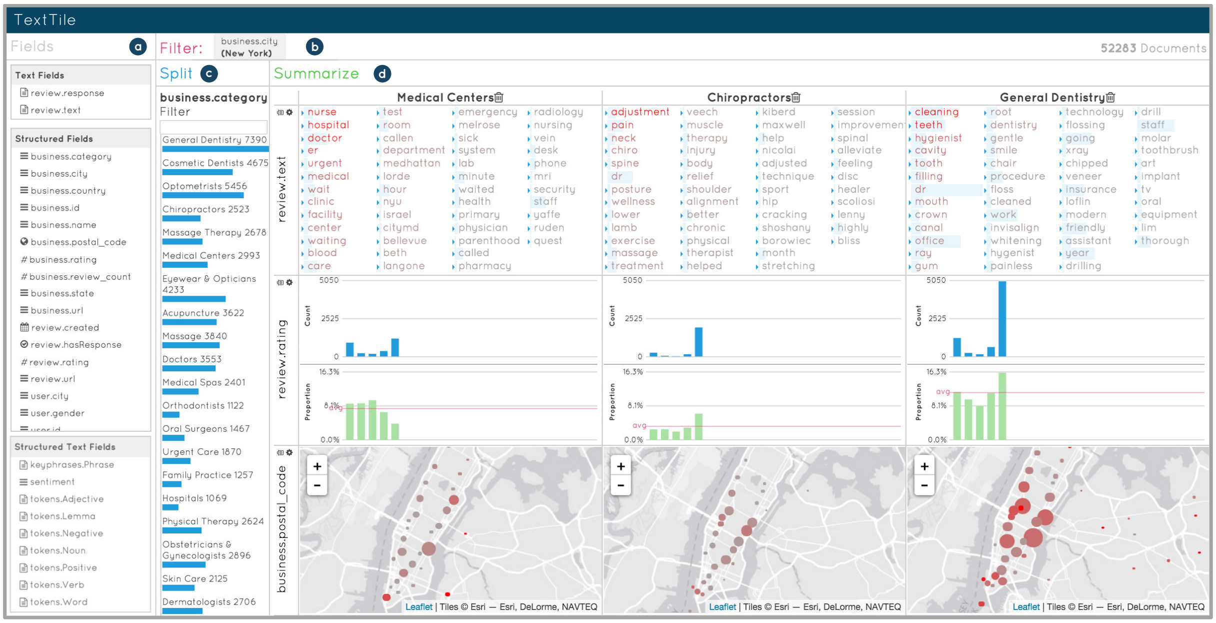


Fig. 1. *TextTile* interface showing data from the *Yelp-Heathcare* reviews dataset with: a) *fields* panel showing all the fields present in the data; b) *filter* panel with filter specification to select only reviews from New York; c) *split* panel with three segments generated using the *business category* field; d) *summarize* panel having three segments (*Medical Centers*, *Chiropractors*, and *General Dentistry*) with *keywords charts* to show relevant words, bar charts to show rating distribution and maps for location distribution by zip code.

Abstract— We describe *TextTile*, a data visualization tool for investigation of datasets and questions that require seamless and flexible analysis of structured data and unstructured text. *TextTile* is based on real-world data analysis problems gathered through our interaction with a number of domain experts and provides a general purpose solution to such problems. The system integrates a set of operations that can interchangeably be applied to the structured as well as to unstructured text part of the data to generate useful data summaries. Such summaries are then organized in visual *tiles* in a grid layout to allow their analysis and comparison. We validate *TextTile* with task analysis, use cases and a user study showing the system can be easily learned and proficiently used to carry out nontrivial tasks.

Index Terms—Exploratory Text Analysis, Knowledge Discovery, Text Visualization

1 Introduction

In this article, we propose *TextTile*, an interactive data analysis tool that enables open-ended investigation of datasets in which structured data and unstructured text co-exist and need to be analyzed in concert.

This condition happens in many practical applications, such as customer relationship analysis, investigative journalism, and survey research, in which such data configuration is very common (e.g., structured user responses plus open-ended text, or customer reviews plus user profiles). In such situations, according to what kind of question is currently pursued, the analyst may need to specify conditions on text first, to see how they are reflected in the structured part of the data, as well as specify conditions on the structured part first, to see how they are reflected in the textual part of the data.

For instance, in the analysis of a survey dataset about a given product of interest, one may be interested in the following two questions: "*In which region do the participants mention the word 'unsatisfied' more often?*"; and "*What do the participants complain about when they give a low rating to the products?*"

Despite being highly related, these two questions require completely different strategies. In the first one, the user needs to first search for entries containing the word 'unsatisfied' and then look at how they distribute over a structured field (the region field). In the second, one needs to first single out entries with a low rating and then find keywords to describe what people complain about in these entries.

Pursuing such a variety of questions, that seamlessly involve structured data and unstructured text in different orders, is often necessary within the context of a single analysis session and currently limited by two main factors: (1) the lack of a systematic way to think about how this integrated set of operations should be organized and linked together; (2) the lack of applications that allow users to utilize this set of operations in an integrated and flexible visualization environment.

In this paper, we propose such a principled method through a data

- *All authors are with New York University. E-mail: cristian.felix@nyu.edu, anshul.pandey@nyu.edu, enrico.bertini@nyu.edu*

Manuscript received 31 Mar. 2016; accepted 1 Aug. 2016. Date of publication 15 Aug. 2016; date of current version 23 Oct. 2016.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TVCG.2016.2598447

visualization system that integrates three main components: (a) a data model that organizes input data in a standard, unified format; (b) a set of operations that transform data into multiple summaries of text and structured data; and (c) graphical representations and interaction methods to compare these summaries organized in a grid.

The work we propose stems from our experience building a set of visual data analysis and presentation tools with several groups of people (see Section 2) including: data journalists, business owners, and development organizations. Through our collaboration, we first realized the need for this kind of analysis and opportunity to group them together into a small set of *abstract tasks* and *operations*.

The paper includes three main contributions. First, the identification of a variety of tasks that often need to be pursued together in the same analysis session and that are hard to pursue in existing visual data analysis tools without major coding or effort. Second, the development of an integrated set of operations that permits to seamlessly pursue such tasks. Third, the development and validation of a data visualization system (*TextTile*). The work is validated through two main steps: use cases to show the capabilities of the tool; and a user study to show that users can effectively analyze data with *TextTile*.

In the following section, we first provide an account of the multiple practical data analysis problems we have encountered and how they led to development of this work. In particular, we focus on how starting from very different domains and problems we can identify a common set of abstract tasks. In Section 3 we describe how our work is related to existing research. In Section 4 we describe the data model and operations; followed by a description of *TextTile* and its visualization and interaction strategies in Section 5. Finally, we present two use cases and a user study in Section 6.

2 Motivating Real-World Scenarios

As briefly mentioned in the introduction, this work stems from our experience developing visual analytics applications for specific groups of users, data, and problems. Our experience includes collaborations with: a business owner to understand how people review restaurants; two groups of investigative journalists to understand respectively how people review doctors and university professors; a private company to explore their internal set of reviews; a development agency to figure out main trends in the response obtained from a worldwide consultation on development priorities; and a group of university evaluators to make sense of how students judge courses and professors.

Out of this list of collaborations, we single out two prototypical ones with sufficiently different application domains and goals. The first one is *ProPublica* with whom we built *RevEx* [13] - a visual analytics system to browse more than a million medical reviews from *Yelp*, the internet reviews aggregator [7]. The second one is *United Nations Office for the Coordination of Humanitarian Affairs (UNOCHA)* with whom we built *WHS Explorer* [5] - a visualization tool to analyze and present the results of a worldwide consultation conducted for the *World Humanitarian Summit (WHS)* initiative [6].

Analyzing a million medical reviews from Yelp. *ProPublica* is an independent newsroom specialized in creating stories stemming from their investigations, with a strong focus on medical practices and issues such as pharmaceutical donations to doctors [2] and quality of surgeries [4]. The main goal of our collaboration is to help them single out interesting stories and trends about how people review doctors from over a million reviews gathered from *Yelp*. Such goal requires analysis of large sets of text with associated metadata and cannot be easily solved with current technologies without coding or sets of complex operations. Below we provide a representative set of questions we gathered through this collaboration.

- $Q1_{med}$: "*What do people mostly talk about when they give high or low rating scores to a health care institute or practitioner?*";
- $Q2_{med}$: "*How does patient opinion compare and differ across medical specialties? Are there issues that are specific to some medical specialties?*";
- $Q3_{med}$: "*How do reviews containing the keyword 'HIPAA' (Health Insurance Portability and Accountability Act of 1996 that protects privacy of patients in the United States) distribute across US states, ratings, and medical specialties?*";
- $Q4_{med}$: "*How do patient ratings compare when they mention 'first visit' versus 'second visit'?*"

Analyzing Consultation Data for UN World Humanitarian Summit. In collaboration with the *United Nations Office for the Coordination of Humanitarian Affairs (UNOCHA)* we developed *WHS Explorer*. The data used in *WHSExplorer* contains text segments extracted manually by UN analysts from document corpus collected from the consultations. Each text segment also contains other structured information, such as *author*, *origin*, *region*, etc. Our collaborators are primarily interested in understanding how different text segments compare according to a set of predefined emerging issues, geography and stakeholder groups. Additionally, they were interested in exploring discrepancies between data collected from "official" channels (reports, co-chairs' summaries) and "social" (on-line discussions, stakeholder analysis, public submissions) data sources. Similarly to the previous example, we present sample questions from this work:

- $Q1_{cons}$: "*How do the priorities differ from one region of the consultation to another?*";
- $Q2_{cons}$: "*How do issues and sub-issues identified in the documents compare by document topic and regional context?*";
- $Q3_{cons}$: "*How do the text segments containing the word 'poverty' distribute across stakeholders, regions and topics?*";
- $Q4_{cons}$: "*How regional distributions compare when text segments contain the phrase 'human rights violation' versus 'poverty'?*"

2.1 Task Abstraction and Operations

The information we collected during our collaborations allowed us to realize that there is a common and interrelated set of tasks users typically want to carry out with these data sets.

In the following, we describe the abstract tasks we derived analyzing the needs of our collaborators and the questions they typically ask. We use the word *document* to refer to the textual part of the data and the word *structured field* to refer to the structured part. More precise definitions will be given in Section 4.1.

Task 1: Evaluate Keyword Summaries. The main goal of this analysis is to find the keywords that characterize a user-defined subset of documents. This task can be split into two sub-tasks according to whether the subset is defined through a keyword search, to find keywords that co-occur with the searched term, or through a filtering predicate over structured fields, to find keywords that characterize the selection. Task $Q1_{med}$ for example requires a keyword summary evaluation while filtering the data by *rating*.

Task 2: Evaluate Structured Data Distributions. The main goal of this analysis is to find how the documents within a user-defined subset distribute over a structured field of interest. This task, similarly to the previous one, can also be split into two sub-tasks according to whether the subset is defined through a keyword search (to see how the searched documents distribute over a structured field) or through a filtering predicate over another structured field. Task $Q3_{med}$ require such analysis, by evaluating a state distribution while using a keyword search to select only documents containing the word 'HIPAA'.

Task 3: Compare Keyword Summaries. The main goal of this analysis is to compare keyword summaries that characterize two or more user-defined subset of documents. Similarly to Task 1, this task can be split into two sub-tasks according to whether the subsets are defined through a keyword search or through a filtering predicate over structured fields. This approach is needed in Task $Q1_{cons}$, since it requires comparing the priorities in the text, across different regions.

Task 4: Compare Structured Data Distributions. The main goal of this analysis is to compare the distribution of two or more user-defined subset over a structured field of interest. Similarly to Task 2, this task can be split into two sub-tasks according to whether the subset is defined through a keyword search or through a filtering predicate over another structured field. Task $Q4_{cons}$, for example, requires the comparison of region distributions across documents containing the words 'human rights violation' versus 'poverty'.

Task 5: Relate summaries and distributions to the original documents. The main goal of this final task is to ensure that at any stage of all the previous tasks it is possible for the user to retrieve and observe the documents that lie behind the statistical aggregates. In $Q2_{med}$ for example, a keyword summary would provide an important initial guidance to the user, but in order to better understand the user opinion may be necessary to look into the real text for context information.

From the analysis of these abstract tasks, we generated a data processing pipeline characterized by three main operations, namely, *filter*, *split* and *summarize*, which, as we describe in more details in Section 4.3, can be combined to answer the large variety of questions outlined above. *Filter* selects subsets of documents according to user-defined filtering predicates. *Split* generates data subsets to be compared according to a user-defined parameter. *Summarize* takes the results of these two operations and generates visual summaries that permit the user to evaluate and understand the results. These operations can take place in the structured, as well as in the unstructured part of the data and can be chained to create more complex operations.

3 Related Work

Although this work shares intersections with various research areas, such as interactive querying and graphical specifications systems [37, 39, 42, 32], in this section we focus on works related to graphical methods to combine and analyze structured data with text. Although some commercial systems exist, such as Tableau and JMP, that allow interactive exploration of datasets, users often have to either code or provide detailed instructions on how to encode the data and how to pre-process it to derive structured information out of text.

Many visualization methods and systems exist that integrate unstructured text and structured data in *specific* ways. In such systems, unstructured data (or text) is analyzed through one of the two strategies, i.e., by deriving statistical information from the text, such as, frequency, sentiment, polarity, etc., or by creating textual representations, such as frequent words, topics, entities etc. We group the systems that integrate unstructured text and structured data into three broad categories of systems. For the purpose of simplicity, we do not distinguish between these strategies while discussing the literature related to unstructured data analytics and visualization.

The first category of systems integrates text with *temporal* data. One such system is TIARA [41] that uses a ThemeRiver[16] like visualization to show the evolution of topics and related words over time while allowing the users to map structured data to colors. Several other works have also visualized the evolution of topics, themes, entities, events, and frequencies of certain keywords over time [12, 31, 40, 24].

The second category of systems uses a combination of *geographical* data and text. From monitoring news related to specific issues like climate change [33], to exploring habitat preferences of various species of birds [14], these systems visualize processed text on geographical maps. These systems are also common in the social surveillance, event analysis, and disaster management where the goal is to associate citizen preferences and needs with geo-coordinates [27, 20, 28].

The third and final category of systems integrates text with *other structured fields*. Such systems allow users to select *metadata* or text and return a set of visualizations that provide a summary of the selected fields. For instance, ConVis[19] visualizes sentiment of forum threads by topics, OpinionSeer [43] visualizes customer feedback on hotel reviews over time and geography etc. A more advanced version of these systems allow the users to create segments based on various fields and compare them through a visual channel. One such example is Opinion Observer [22] that allows users to compare sentiment extracted from product reviews across *aspects* extracted from the text as well as the structured fields. ViTA-SSD [38] is another system that allows users to select and visualize the distribution of various structured fields, as well as create and compare keyword clusters using word clouds. Lastly, Parallel Tag Clouds [11] supports the comparison of subsets of the data using word clouds without visualizing the distribution of other structured fields.

Despite all the development in the analysis of text-metadata analysis, to our knowledge there is no system that allows for the exploration of text and metadata simultaneously, while supporting the three operations described in this paper, i.e., *filter, split* and *summarize*.

With regard to interaction, most modern systems employ keyword and/or faceted search strategy [17] to retrieve documents of interest, where the interface provides interactive components and visual representations to allow the user to specify values to include or exclude from a search. While it is still possible to focus on specific subsets of documents using this approach, the main goal of our work is to enable the detection and analysis of quantitative and qualitative trends in the dataset. In *TextTile* we integrate elements of these classic search interfaces allowing the users to query the dataset according to the desired keywords and various data fields (facets).

4 Operations and Data Transformation

The system is based on the pipeline presented in Figure 2 which transforms, through a series of *user-driven interactive operations*, the *input data* into useful *visual representations*.

4.1 Data Model

We define the input data as a data table `D` in which every row represents one *text document* and every column represents information associated to this document. The table is characterized by three sets of fields we define as follows:

Structured fields contain structured information associated to the documents. Following widely accepted conventions in previous visualization research [29], we categorize *structured fields* into: *categorical*, *ordinal* and *quantitative* fields. We also distinguish between two field *semantics*: *geographical*, to represent a geographical region and *temporal* to represent dates and time. **Unstructured text fields** contain the actual text that represents the document. These fields do not contain any structural or meta information: they only contain sequences of characters (letters, spaces, punctuation, etc.). **Structured text fields** contain structured information *derived* from the *unstructured text fields* through natural language processing (NLP) methods (e.g., sentiment scores, cluster membership, topic).[1] These fields can be of any of the same types used to describe structured fields.

4.2 Overview

The pipeline consists of three main steps through which the input data table is transformed into multiple smaller tables that populate the visualizations generated by the system.

- **Filter.** The filter step allows the user to filter the data according to user-specified conditions over one or more of the data fields outlined above. In particular, the collection can be filtered according to logical operations applied to structured fields as well as keyword search mechanisms over the unstructured text field.
- **Split.** The split step allows the user to split (and later *compare*) the data into multiple subsets according to the (unique) values found in one user-selected field or according to keywords provided by the user (one or more keywords for each segment). We will refer to these subsets as *data segments* (or just *segments*) in the rest of the paper.
- **Summarize.** The summarize step allows the user to decide which field to use to summarize the data segments generated by the *filter* and *split* operations. Once the original table has been filtered and split into multiple tables, it is necessary to decide how to summarize the data found in each segment. *Summarize* uses the user-selected field to aggregate the data found in the segment, e.g., counting the number of documents in each category of a selected categorical field.

[1] The system is agnostic to what specific NLP methods are used to derive structured text fields. We only assume that it is possible to derive one value for each document.

The pipeline acts as a table transformation sequence that takes the whole data table `D` as an input and transforms it into multiple data tables that summarize the data according to the filtering, splitting and summarization conditions selected by the user. More precisely, the *filter* step reduces the data, the *split* step splits the data into multiple *segments* to compare and the *summarize* step aggregates them into meaningful aggregates (see details in Section 4.3). *Split* and *summarize* create a matrix of data summaries $N \times M$, where N is the number of data segments generated by *split* and M is the number of distinct fields used to summarize each segment.

The operations accept both structured and unstructured data fields as parameters selected by the user. The *filter* step can use logical predicates on structured fields and keyword search to filter the data. The *split* step can use the values found in a structured field or keywords inputted by the user to create multiple data *segments* to compare. The *summarize* step can create summaries over *structured fields* as well as over *unstructured text fields* (see details in Section 4.3.3).

To exemplify how interesting questions can be pursued by tying these operations together, here we provide one example based on reviews of medical providers obtained from *Yelp*, the popular Internet reviews aggregator. Let us assume the user is interested in the question: "*what do people talk about when they give negative reviews and how does this change across medical specialties?*" *TextTile* helps answering this question by (1) focusing only on reviews with a rating between 1 and 2 [**filter**]; (2) splitting and comparing the reviews by medical speciality (e.g., dentist, pediatrician, chiropractor) [**split**]; and (3) visualizing a summary of relevant keywords for each segment [**summarize**]. More examples will be provided in the use cases section (Section 6.1). In the following subsections we will first describe the operations in more details, then we will describe how the data produced by the data transformation pipeline is visualized through a number of visualization and interaction strategies.

4.3 Operations

In this section, we use *SQL* notations to describe how the data at each stage is processed. It is important to note that such notation is not necessarily an accurate representation of the internal workings and implementation of the system; rather it is intended as an explicative tool of the system's behavior. We will use the convention of first describing how these processing steps apply to *structured fields* and then how they apply to *unstructured text fields*. We will not distinguish between *structured fields* found in the original dataset and those derived from text (i.e., *structured text fields*). Therefore, in this section we will refer to both cases when mentioning any structured field type.

4.3.1 Filter

The main goal of this operation is to filter the dataset according to user-specified filtering rules. The *filter* operation is described by the following statement expressed in an *SQL* query:

```
SELECT * FROM D WHERE {filters}
```

where `D` is the original data table and `{filters}` is a collection of logical statements/specifications concatenated by the `AND` operator. The filters have the effect of removing the items that do not satisfy their specifications. For a categorical field `a`, a filter is a logical condition such as `a = K` or `NOT a = K`, where `K` is in the domain of field `a`. For an ordinal or quantitative field `a`, the same logical statements are allowed with the addition of filters involving inequality operators e.g., `a > K` or `a < K`.

For fields of type text, the same statement holds. The only difference consists in the use of a keyword search filter: `search(t, key)`, in which `t` represents an *unstructured text field* in the data table and `key` is a keyword to search for. The filter returns `TRUE` if the keyword `key` is present and `FALSE` otherwise.

It is worth noticing that filters of many types can be concatenated in the same statement; thus allowing the specification to involve, if needed, multiple structured and unstructured fields.

4.3.2 Split

The main goal of this operation is to support the comparison task defined in Section 2.1 by generating a new segment for each distinct value of a selected field of interest. For instance, splitting by the field *state* contained in the medical reviews dataset described in Section 2, creates as many separate data segments as the number of US states contained in the field. Unlike *filter*, which leads to elimination of data objects that do not satisfy the user-defined filtering criteria, *split* does not remove any data items. It exclusively divides the data into units the user is interested in comparing. For categorical and ordinal fields, the *split* operation takes as an input the data table `D′` generated by *filter* and one user-selected field `a` in `D`, and generates data tables for each unique value in `a`. For a given unique value `K`, the generation of its segment is described by the following statement:

```
SELECT * FROM D′ WHERE a = K
```

For quantitative fields, the generation of each segment is driven by a binning function `bin(a, s)`, that, given a field `a`, returns a *bin label* with the field value and the size `s`. This function is used to generate a set of *bin labels* and for a given *bin label* `B` a segment is thus created with the same *SQL* statement described above using the following clause: `WHERE bin(a, s) = B`

For *unstructured text fields* the splitting is driven by the user, who specifies the desired segments by providing keywords; one for each segment. That is, while for *structured fields* the generation of the segments is generated implicitly from the field according to its content and type, for *unstructured text fields* the segments are generated exclusively from keywords provided by the user.

Each segment associated to an *unstructured text field* `t` is thus generated using the same search function described above through the following clause: `WHERE search(t, key) = TRUE`. It is worth noticing that while the *filter* operation allows to combine statements across multiple fields (of mixed type if desired), the *split* operation focuses exclusively on one data field at a time.

4.3.3 Summarize

The main purpose of this operation is to summarize the data contained in each generated segment according to one or more selected fields to see the data distribution. For instance, using our medical data example, one may want to *split* the data by *medical specialty* and observe how *rating distribution* differs across each segment (e.g., does *general dentistry* receive a larger proportion of 1 stars compared to *cosmetic dentists?*). While other options exist for a summarize operation, e.g., computing one single aggregate statistic for each segment, we opt for a solution that provides more information without overwhelming the user.

Summarize is applied to all the tables generated from *split* to create new summary tables that capture aggregate information from their values, generating a grid of $N \times M$ summaries where N is the number of segments generated in the *split* step and M is the number of summaries fields specified by the user. For a given segment table `D′′` and user-selected field `a`, the summary operation is described by the following query expressed in *SQL* language:

```
SELECT a, COUNT(a) FROM D′′ GROUP BY a
```

which results in a table containing information about the distribution of the field `a` over the table `D′′`. For ordinal and categorical fields, the query result returns the count for each unique value, while for quantitative field the aggregation is driven by the same binning function described in the *split* operation which returns a binned ordinal value for each quantitative value found in the field.

For summaries of *unstructured text fields*, we opt for a model based on keywords, that is, for a given collection of documents we define a summary as a list of words with associated statistics (e.g., word frequency). While this is not the only possible model, we believe this is an intuitive and reasonably transparent way of integrating text summaries into our model. In order to build such a summary, we need a procedure that extracts keywords and associated statistics from

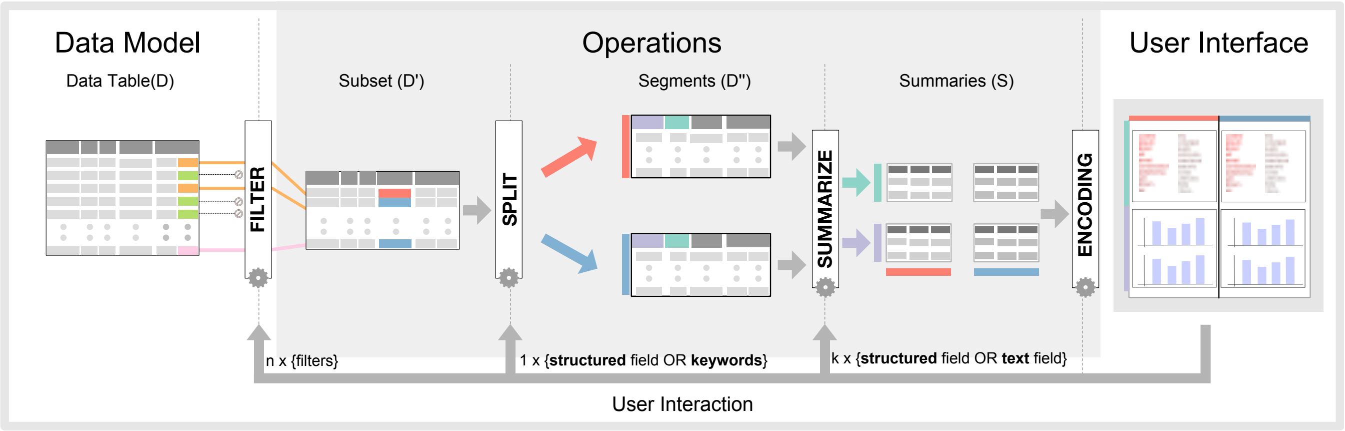


Fig. 2. Overview of the model with the three main operations (filter, split, and summarize). Based on user interaction, the data flows from left-to-right, from raw data to visual summaries.

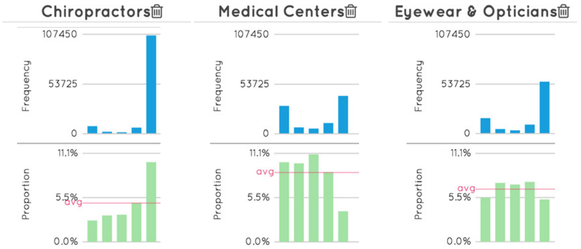


Fig. 3. Difference in the distribution of ratings by category. While all categories have mostly 5-star reviews by frequency (blue bar chart), the proportion chart (green bar chart) shows positive reviews trend for *chiropractors* and negative for *medical centers*.

documents belonging to a given segment. To ensure decoupling between the query mechanism and the specifics of how keywords are extracted, we introduce an auxiliary table `keywords` which keeps track of which keywords are associated to each document in the dataset. The table has two columns: `doc_id` containing document identifiers, and `keyword`, containing keywords extracted from the text, and records information on which keywords are associated to each document.

Such a table is populated by an internal routine that scans the document collection and derives the association between documents and keywords, thus creating a loose association between the querying mechanism and the specifics of how keyword extraction and association are performed.

Relying on a separate table has the advantage of making the term extraction and association process flexible and decoupled from the main logic. It also allows to treat separately important processing steps such as stop-words removal and stemming [25], and the addition of extended fields obtained from part-of-speech extraction such as adjectives, nouns and verbs [25]. For a text field the summary operation is thus described by the following query expressed in *SQL* language:

```
SELECT keyword, COUNT(keyword)
FROM D’’ JOIN keywords GROUP BY keyword
```

which counts how many documents in the segment contain each keyword contained in the list of extracted keywords.

So far, in the descriptions and queries presented above, we limited ourselves to summaries based on raw counts of data items in the data tables. When data is filtered and/or split, we typically want to see what effect these operations have on the overall distribution of a given field of interest. Figure 3 shows an example of such distribution, where the blue bars represent the count of data items across medical specialties (the panels) and rating (the bars). As one can see, while it is possible to realize that there is an impact of medical specialty on the rating distributions, it is hard to identify how much each category has changed.

To overcome this problem, we include the calculation of *proportions* together with *raw counts*. The proportion is calculated as follows. Let us call C_i the *raw count* across the whole dataset of the i-th category of a given field, and $C_{i,j}$ the count of the same category in the j-th segment. The proportion $P_{i,j}$ is then calculated as $C_{i,j}/C_i$. As one can see in Figure 3, where the green bars below represent proportions, it is much easier to understand that *chiropractors* have a much more favorable distribution, compared to the baseline, than *medical centers* and *eyewear & opticians*.

Scoring words by relevance. In order to make summaries of *unstructured text fields* more useful and usable, we have to introduce a scoring function able to score the words according to a computed *relevance* value. Such relevance value has the main objective of ranking words according to how *specific* they are for a given data segment rather than how frequent. While a given set of words may be frequent across all segments, they are usually relevant only for a few.

There are many computational methods to compute a relevance score for a given data segment [10]. Here we focus on the well-known and popular *term-frequency over document frequency (TF/IDF)* model which computes relevance as the ratio between how frequent a word is in a segment over how frequent it is over the entire collection.

5 System Description

We now describe various components of *TextTile* in detail. As we have seen above, the user can make the following choices: (1) select criteria to filter the data; (2) select which field or keywords to use to split the data into segments; and (3) select which fields to use to summarize the data contained in each segment. In what follows, we describe the data processing and the model implementation, the data visualization and interaction strategies, and the user interface.

5.1 Data Processing and Model Implementation

As outlined in section 4.1, the system expects a table as an input containing a collection of *structured fields* and one or more *unstructured text fields*, with metadata providing information about their type.

Once the table is loaded, the system needs to pre-process the information contained in *unstructured text fields* to generate the associated *keyword tables* and *structured text fields* described above.

The *keyword table* contains information about which keywords are associated to each document, plus additional meta-data such as word frequency. In our current implementation the keyword table is generated while importing the data into the system, applying in sequence: word *tokenization*, *stemming* and *stop words removal*.

For the *structured text fields* we currently rely on external preprocessing and the fields are loaded together with *structured fields*. As of now, we have experimented with derived information such as *sentiment analysis* [21], *document classification* and *topic extraction* [25]. As part of our future work we want to devise methods to directly integrate these processing steps within *TextTile*.

The current implementation is based on *Elasticsearch* [3] for data processing and querying, *Lucene* [1] to index the documents, and the *Stanford CoreNLP* library [26] to compute structured text fields. We have also developed an initial *data adapter* that aims at making the connection to future database systems, such as *MongoDB* or *MySQL*, as transparent and seamless as possible.

5.2 Visualization and Interaction Strategies

We now discuss general design choices we made on *TextTile* to allow the user to create data queries and effective visual representations.

Interactive, visual and reversible specifications. While the operations outlined above could be implemented in a system based on command-line-driven approach, *TextTile* follows the principles of *direct manipulation* and *visual feedback* [35] in which the user has good *visibility* of what operations are available, good *mental model* of what actions are needed to generate a given specification, visual feedback to observe the effect of the actions and clear ways to *reverse* or *edit* these actions, if needed. *TextTile* implements these principles using a drag-and-drop mechanism (inspired by Polaris [39] and Tableau) to specify which fields to use for which operation and allowing all operations to have immediate visual feedback and be reversible.

Intuitive and consistent layout and comparison mechanisms. The data summaries generated by the data pipeline can be imagined as a matrix in which the rows represent multiple fields selected for the *summarize* operation (e.g., three rows – one for a text field, one for a geography field and one for time), and the columns represent multiple *segments* (or *tiles*) based on the values of the field on which the *split* operation is performed (e.g., one segment for category/value of a given categorical field). As shown in Figure 1(d), this metaphor is indeed used in *TextTile*. Such layout allows the user to compare across *segments* (horizontally between values) and across *summaries* (vertically between variables).

To ensure consistency, primarily for comparison, we employ the following strategies: (a) all the summaries (for structured fields) in a given row share a common y-axis; (b) any configuration applied to one *segment/tile*, e.g. sorting by count or proportion, is applied to the other *segments* in the same row; (c) hovering over an element of a summary highlights comparable elements across all *segments*; and lastly, (d) given the high cardinality of words in text, in the *keyword chart* we add an additional visual cue, i.e., a small triangle next to the word to communicate exclusivity, allowing better comparison.

Familiar, effective and data-specific graphical representations. When choosing what graphical representation is appropriate for a given data summary, we aim to maximize *simplicity*, *familiarity*, *intuitiveness* and *effectiveness*. To satisfy these goals, *TextTile* (a) uses visualizations based on charts with proven efficacy and widespread adoption, and (b) produces visual representations that adapt to the specific field type selected by the user. More precisely, *TextTile* uses different representations for the 6 available field types as follows: bar charts for categorical, ordinal and quantitative fields (sorted bars for ordinal and binned for quantitative), line charts for temporal, maps for geographical, and keyword charts for keywords extracted from text (an intuitive and effective representation we describe below). While it is always possible to implement more sophisticated visualizations techniques for such field types, we strive to strike a balance between the aforementioned goals. We believe simplicity and familiarity should always come first and that it is important to keep into account the cost of introducing novel and non-standard visual representations into a system.

Transparent aggregates. As we explained in Section 4.3.3, the *summarize* operation aggregates information using two main aggregate functions: one with the *raw count*, and the other with the *proportion* between count *within* the segment and *across* the whole dataset. This solution enhances the transparency as the two values can be visualized at the same time. In turn, this means that we have to devise graphical solutions to show these two values at the same time in all the visual representations we provide. While it is possible to provide interactive options to switch between one value to another, we deem important, and hence implement within *TextTile* the ability to visualize both values at the same time. To solve this problem, the system uses the following solutions for each of the chart types we described above. Bar charts and line charts are always split into two portions: *raw count* on top and *proportion* at the bottom (as depicted in Figure 3). For maps, we use a bubble with size mapped to *raw count* and color saturation mapped to *proportion*. For keyword summaries, we use the graphical solution presented below.

Effective representations for keyword summaries. When an *unstructured text field* is used to create a summary, it is important to find an effective visual representation able to present the following three pieces of information: the *keywords*, their *frequency* and their *relevance* (as explained in Section 4.3.3). While a standard *word cloud* representation would seem a natural choice for this task, *word clouds* have numerous shortcomings that make them ineffective for analytical purposes [30, 8, 15, 23, 34, 18]. Some of these shortcomings include the lack of natural ordering, the ineffective use of font size to communicate quantitative information, the variation in size due to word length rather than value. For this reason, we introduce a simple but effective representation we call *keyword chart*, which can be observed in Figure 4. In this chart, the words are arranged in columns and ordered top to bottom and left to right. The words can be ordered according to the *raw count*, *relevance* or *key*, with relevance being the default option. Behind each word, a bar represents the raw count of the word. Color is used to identify relevance. This representation is intuitive and overcomes the issues mentioned above.

5.3 *TextTile* Interface

The *TextTile* interface is divided in four main panels, as shown in Figure 1, (a) fields, (b) filter, (c) split, and (d) summarize panels, and provides the user with interaction methods to: first, specify *filter*, *split* and *summarize* parameters, and to second, observe the results through carefully crafted graphical representations that follow the principles provided in Section 5.2.

The **fields** panel contains the fields available in the dataset divided in three sections, according to field type: *structured fields*, *unstructured text fields* and *structured text fields*. The **filter** panel is used to add multiple filters by dragging fields from the **fields** panel onto it. When a new filed is dragged in the **filter** panel, *TextTile* reacts by showing a *filter selection* view that allows the user to select the desired values to use for filtering . For instance, if the selected field is *categorical*, the *filter selection* view shows the distribution of the various categorical values in the field, while allowing the user to select/deselect specific values. Similarly, if the attribute is *text*, it allows the users to specify one or more keywords to include in the filters.

The **split** panel displays data distribution across segments created by the *split* operation. To create segments, the user can drag-and-drop a field from the **fields** panel onto the **split** panel, where each segment corresponding value of the selected field is represented by a horizontal bar. The length of the bar represents the number of data points contained in the segment. The segments can also be merged on-demand by dragging and dropping a segment on top of another. The user can then assign a new name to the merged segment.

The **summarize** panel allows the user to generate visual summaries of selected fields and compare them across segments. To generate visual summaries the user drags the field from the **fields** panel onto the **summarize** panel, the system creates a new row for each field the user drags, the visualization adapts to the field type while following the conventions described in Section 5.2, and provides an appropriate visualization to summarize the data. If no segments are selected for comparison, *TextTile* will generate an overall summary of the selected field(s); thus allowing the user to start exploring the dataset without preemptively applying any data reduction or splitting operation.

To compare the segments, the user can drag-and-drop segments onto the **summarize** panel . This creates a new column that contains the summaries of the segment, allowing a side-by-side comparison.

Finally, it is important to always allow the user to retrieve specific raw data instances related to some pattern of interest found in the visualization. While interacting with *TextTile*, the users can select elements of the visualization, such as the words in the the *keyword chart* or bars in the bar chart, and see the details, i.e., the underlying data table entries, on-demand.

6 Validation

We use a two-way approach to validate *TextTile*: (1) a pair of use cases to highlight the *capabilities* of the system; and (2) a user study to evaluate the *usability* of the system, as detailed below.

6.1 Use Cases

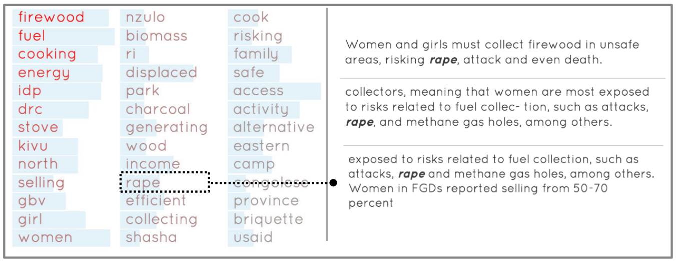
As the first validation step, we present two use cases to highlight the capabilities of the system in data analysis tasks.

6.1.1 Understanding World Humanitarian Needs

The World Humanitarian Summit (WHS) is an initiative of the United Nations, and is managed by United Nations Office for the Coordination of Humanitarian Affairs. The goal of the project is to work with humanitarian stakeholders, and seek better ways to help millions of people around the world who are in need of humanitarian support.

As part of this initiative, a dataset was curated, that contains parts of documents generated through consultations (with humanitarian stakeholders) around the world. The dataset also contains documents and recommendations generated by academicians as well as the concerned departments of United Nations. Overall, the dataset comprises of 478 documents that are segmented in 16760 text snippets, where each text snippet contains metadata information such as *topic*, *subtopic*, *national context*, etc.

We start our analysis with the question "*which humanitarian aspects arise in each national context?*". We first *split* the data by the *national context* field, and *summarize* the *text* field. This results in a set of *keyword charts* containing the most frequent words for each country. By looking at these summaries, we notice that words related to *disasters* and *refugees* are very frequent. However, *texts* originating from Congo show high usage of words like *fuel, cooking, firewood and women* (Figure 4). Upon investigating the underlying text in the *details view*, we find that although some people in 'Congo' receive food donations, women often have to go to the forests to collect firewood for cooking, making them easy victims of violence and sexual harassment. This finding highlights the fact that humanitarian organizations often donate food items, while overlooking the requirement to setup a safe mechanism to cook them.



Women and girls must collect firewood in unsafe areas, risking ***rape***, attack and even death.

collectors, meaning that women are most exposed to risks related to fuel collec- tion, such as attacks, ***rape***, and methane gas holes, among others.

exposed to risks related to fuel collection, such as attacks, ***rape*** and methane gas holes, among others. Women in FGDs reported selling from 50-70 percent

Fig. 4. *Keyword chart* showing words relevant to 'Congo', and text segments containing the word 'rape';

As there is no structured field that can tell if a document is about *disaster* or *refugees*, we *split* the data by these two keywords to create two segments, containing documents with the word *disaster* and *refugee*, respectively. We generate *keyword charts* for each of these segments to see other co-occurring keywords. The most discriminative words in the *disaster* segment include *risk, management, plan, capacity*, pointing towards disaster management strategies to subsidize risk, whereas, those in the *refugee* segment show related countries like *Jordan, Syria, Lebanon*, and also the affected population, like *women and children*. It also points to some issues existent in refugee camps like *violence, harassment* and the need for *education* and *health* systems.

Next, we decide to see how authors from different countries talk about these two issues (i.e., disaster and refugees). We drag the *national context* field to the *summarize* panel, which is already split into two segments over the previous steps. Using the *map*, we find that texts talking about *disaster* are more frequent in the pacific region, whereas the refugees-related issues are discussed more in Africa and the middle-east Asian region (Figure 5). Papua New Guinea is the only country with large number of documents related to both issues, as it is in a disaster-prone area, and also has a refugee camp.

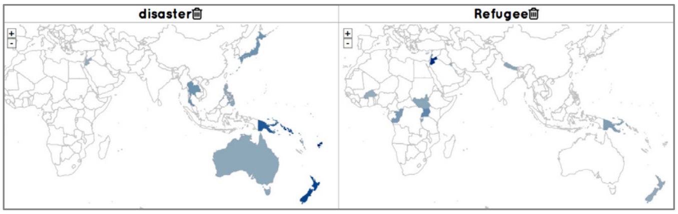


Fig. 5. Distribution of documents by countries across *disaster* and *refugees* segments

Understanding how relevant these two issues are to various stakeholders is important to create plans for tackling humanitarian crisis. For this reason, we add *stakeholders* to the summarize panel, creating bar charts that show the distribution of frequency and proportion of documents for each *stakeholder* category. We find that *Member States* discuss *disaster* more than any other topic, with the word *disaster* being present in 24% of all text segments they wrote, whereas, just 1.1% containing the word *refugees*. *Media*, however, write more about *refugees* (21%) than *disaster* (15%), even though there are more documents about *disaster* than *refugees* (Figure 6(b)) in overall. *Affected Communities* mention *disaster* more than *refugees*, however, the proportion to the total number of documents mentioning *refugees* is almost twice the proportion of those that mention *disaster* (Figure 6(a)). This mismatch between what *Member States*, *Media* and *Affected Communities* talk about shows a possible inconsistency between the views of the government, media and the affected people.

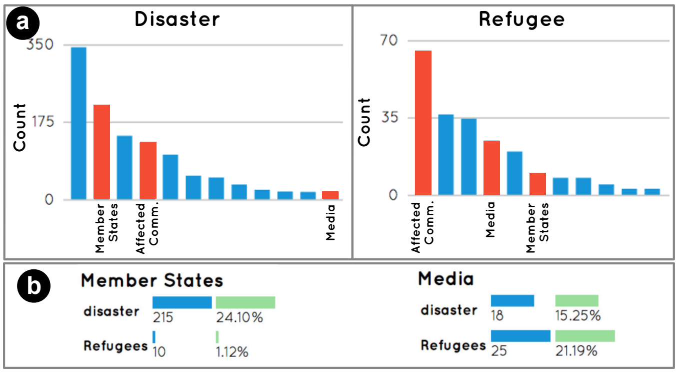


Fig. 6. (a) Distribution of *stakeholders* in *disaster* and *refugee* related documents. (b) Tooltips for *Media* amd *Member States* stakeholder, showing count and proportion for *disaster* and *refugee* segments

In this use case, we demonstrate how a simple analysis using *TextTile* can lead to some interesting insights for WHS, such as, making sure that people who receive food also have means to cook; need to synchronize what government and people care about; and how refugee camps suffer from critical issues such as education, health and safety.

6.1.2 Detecting Malpractices at Healthcare Centers

The main goal of our collaboration with *ProPublica* is to help them identify specific healthcare centers that are involved in medical malpractices, using customer reviews on *Yelp*.

To that purpose, we start our analysis with a high level exploration of how reviewers rate heathcare centers/businesses, i.e., how the *rating* distributes in the data. We *split* the data by *review rating* and find that the distribution is highly skewed towards positive reviews. To know what people say, we create visual summaries of the reviews with highest (5/5) and lowest (1/5) ratings by dragging the two ratings from *split* to *summarize* panel. By investigating the most distinct words in the two segments, we find, in the low rating segment, keywords that refer to malpractices, such as *unethical*, *scam*, *lied* etc.

We now want to expand this dictionary with other terms that may also relate to medical malpractices. Hence, we clear all the configurations and add the three terms – *unethical*, *scam*, *lied* – to perform the *filter* operation. This returns only reviews that contain at least one of the these keywords. Next, we *summarize* the reviews by dragging the text field to the *summarize* panel, creating the *keyword chart*. We identify additional terms, such as *fraud*, *dishonest*, *liar*, *illegal*, etc., that appear with at least one of the three terms. We update the *filter* operation by including the additional terms.

We now have a list of reviews that mention at least one of the terms related to malpractices. As our dataset contains reviews coming from all over the United States, our next step is to identify cities with a large proportion of malpractices-related reviews. We generate a new summary using the *city* field to look for the *cities* with highest proportion of reviews containing these terms. We find that New York City, when compared to the average line, has the highest proportion of such reviews (Figure 7(a)). To investigate this proportion, we update the 'filter' operation to get reviews for businesses from New York City.

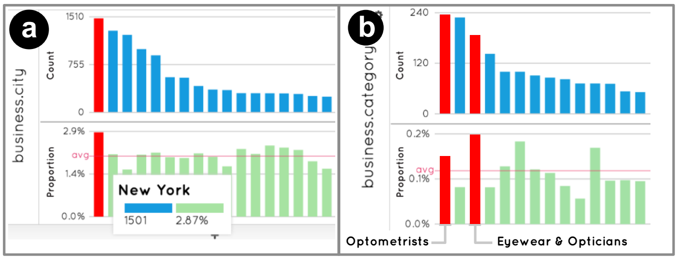
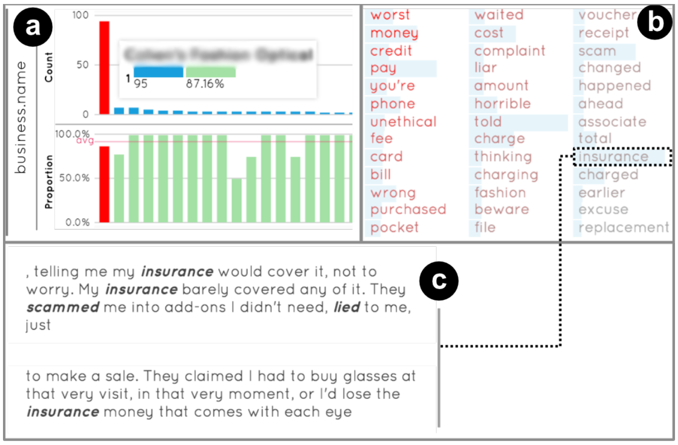


Fig. 7. (a) Distribution of reviews mentioning malpratices words (e.g fraud, dishonest, liar) by cities. (b) Distribution of reviews mentioning malpratices words by business categories located in New York

In the third step, we move further to identify the *business categories* within New York that would be of interest. Hence, we add the *business categories* field to the *summary* panel. Figure 7(b) shows the proportion of reviews by business categories with only reviews related to malpractices for businesses based in New York City. It can be seen that the second and third highest mentions were for categories *Optometrists* and *Eyewear and Opticians*, and that their proportion of malpractices reviews are fairly above the average. We update the *filter* operation to choose only these two categories.

Next, we *split* the remaining reviews based on *rating* and *summarize* them by *business name*. This time we project only the 1 rating segment. The resulting view shows the names of businesses along with the distribution of the number of reviews that fit the *filter* criteria. We identify *Optical Inc.*(real name removed for anonymity) as having the largest number of reviews mentioning one of the listed terms, with over 10 times more reviews than the business that ranks second (Figure 8(a)). Finally, we summarize the reviews further by *text* (Figure 8(b)). In the *keyword chart*, we find that *insurance* is the most commonly co-occurring term. By inspecting the reviews associated with *insurance*, we find that reviewers have been regularly complaining about insurance billing fraud. Most of them complained that they were told that their treatment was covered under the health insurance, but were later billed. This finding raises suspicion about the ethical functioning of *Optical Inc.*, demanding a careful scrutiny of their business.



, telling me my ***insurance*** would cover it, not to worry. My ***insurance*** barely covered any of it. They ***scammed*** me into add-ons I didn't need, ***lied*** to me, just

to make a sale. They claimed I had to buy glasses at that very visit, in that very moment, or I'd lose the ***insurance*** money that comes with each eye

Fig. 8. Diplaying summaries of reviews with 1 star, mentioning malpratices words from business located in New york (a) Business distribution, (b) discriminat keywords, (c) sample reviews.

Tasks	Guidance	Correct State	Correct Answer	Time [95% CI]	Easiness [95% CI]
Task 1	High	100%	100%	1:35 [1:20, 1:50]	4.92 [4.76, 5.07]
Task 2	High	100%	100%	1:51 [1:36, 2:05]	4.42 [4.14, 4.70]
Task 3	High	100%	**67%**	1:42 [1:22, 2:02]	4.50 [4.13, 4.87]
Task 4	High	100%	83%	2:25 [1:58, 2:53]	3.58 [3.22, 3.95]
Task 5	Low	100%	100%	4:54 [3:02, 6:45]	**2.75 [2.28, 3.22]**
Task 6	Low	100%	83%	1:30 [1:11, 1:49]	4.08 [3.72, 4.45]
Task 7	Low	100%	92%	2:20 [1:45, 2:55]	4.25 [3.73, 4.77]
Task 8	Low	92%	75%	3:57 [2:54, 5:01]	3.67 [3.19, 4.15]
Task 9	None	100%	100%	2:22 [1:43, 3:01]	3.83 [3.28, 4.39]
Task 10	None	100%	100%	**5:42 [4:22, 7:01]**	3.25 [2.78, 3.72]
Task 11	None	83%	83%	3:04 [1:48, 4:20]	3.25 [2.55, 3.95]
Task 12	None	100%	83%	1:47 [1:38, 1:57]	4.17 [3.78, 4.56]

Table 1. Aggregated participants' performance for each task along with the level of guidance provided to successfully complete the task.

In this use case, we demonstrate how *TextTile* provides a complete solution to conduct investigative analysis on a large corpus of reviews, starting from a simple heuristic, followed by a series of more complex queries and analyses, leading to interesting findings in healthcare malpractices.

6.2 Usability Study

To evaluate the usability of *TextTile*, we conducted a lab study with 12 participants (10 males, 2 females). The study comprised of four sessions in the given order: a 20 minutes introduction and demonstration of *TextTile*, a 20 minutes training session with three tasks with different levels of guidance, an optional 10 minutes review session, and a 45 minutes final study. All the sessions were conducted on a 27" high-definition monitor with *TextTile* pre-loaded on Google Chrome browser. Participants used a keyboard and a mouse for interaction.

In the demo session, a walk-through of *TextTile* was provided using the *WHS* dataset, followed by a training session using *Yelp-Restaurants* dataset. The three tasks in the training session had to be successfully completed (with correct answers) to complete the training. An optional 10 minutes review session was conducted to revisit certain features of *TextTile*, which only 3 out of 12 participants used.

Upon successful completion of the training session, the final study was conducted using *Yelp-Health* dataset. Both the datasets, *Yelp-Health* and *Yelp-Restaurants*, have the same schema. A total of 12 tasks were given to the participants, one at a time. The 12 tasks include combinations of filter, split and summarize operations, to be performed on structured and text fields. In the first 4 tasks, we provide a *high* level of guidance to the users by mentioning the data fields to choose, and operations to perform, in sequence, to successfully complete the task. The next 4 tasks contain a *low* level of guidance, with only important data fields mentioned in the task. The last 4 tasks contain *no* guidance, requiring the participants to translate the question into fields and set of operations to use. The participants were allowed to ask questions and seek clarity about the tasks as well as the data fields. Additional help and clarification was provided if a participant was stuck, while keeping a note of the reason they got stuck. Each task started from a vanilla state of *TextTile*, i.e., without any pre-saved configurations.

For each task, we recorded the findings in a textfield, the final state of the *TextTile* at the time of submitting the response, a 5-point easiness score, and the time to complete the task. We consider the final state correct if it is possible to correctly respond to the question based on the information presented on the screen. Similarly, we consider the collected response (in textfield) correct if it matches with the ground truth, which is established by the instructors through prior exploration.

At the end of final study, participants were provided a post-study questionnaire, which was not timed. The first part of the questionnaire had demographic questions (age, gender and education), followed by system usability questions [9]. Next, we presented an ease of use questionnaire to record participants' agreement/disagreement with a presented statement on a 5-point Likert scale (1 = strongly disagree, 5 = strongly agree) about the ease of use of operations, visualizations and the analytical flow supported by *TextTile*. Finally, an open-ended questionnaire was provided to learn about users' prior experience with similar tools and their overall preference. Finally, all participants received $20 for participation.

6.2.1 Results

We now present the analysis of participants' performance on the tasks and response about the overall system usability and preference.

Analysis by Task. Participants took an average time of 27:31 [23:56, 31:06] (format: minutes:seconds) to complete the 12 tasks in the final study. Table 1 shows the aggregated performance metrics – success in reaching the correct *state*, providing the correct *answer*, and *time* spent on the task. Each task in the table is marked by the level of guidance provided to the participants to successfuly complete the task.

We found that participants consistently found it hard to successfully complete the task that required using *maps* and took more time in completing the task, as shown in the Table 1[Task 5], even though they answered the task with 100% success rate. Task 5 required the participants to *summarize* the postal codes, that generated a geographical map of New York city, and identify the postal code with the highest number of records, which was encoded as the size of the bubble. The same was corroborated in the qualitative feedback, when participants reported the lowest average easiness score of 2.91 [2.35, 4.48] in interpreting the information contained in *maps*. 4 out of 12 participants mentioned that using *maps* was the hardest while responding to which feature/aspect of *TextTile* they found the hardest to understand or use. Upon further scrutiny, we found that the participants found it difficult to pan every time they zoom into the map, which increased the complexity of the task. Allowing the participants to switch between the map and bar charts could have mitigated this problem, which we identify as an improvement to be made on the tool.

Another cumbersome task was Task 10, with highest average completion time of 5 minutes 42 seconds. This task required participants to identify at least one topic (supported by 3 keywords found in the *keyword chart*) as a reason for why reviewers give high (5/5) rating, and one for why they give low (1/5) rating to a medicare center. As no guidance was provided, the difficulty of the task increased. This task also sought additional cognitive processing by requiring participants to interpret common topics based on keywords . As expected, participants found the task comparably difficult to do (avg. easiness score: 3.25 [2.78, 3.72]), and required the maximum amount of time (5 : 42 [4:22, 7:01]) to complete the task. As participants were able to achieve the correct *state* and *answer* with 100% success rate, we believe that *TextTile* can be efficiently used for cognitively challenging analyses.

An exception here is the performance of participants on Task 3, where they were asked to provide top 4 relevant words related to the business category 'General Dentistry'. Being one of the highly guided tasks, all participants were able to complete the task quickly and considered the task to be easy with easiness score of 4.50 [4.13, 4.87]. However, participants recorded lowest success rate of 67% in anwering the question. We found that most participants provided the top 4 most 'frequent', instead of 'relevant', words, whereas others provided only 3 most relevant words, pointing to carelessness as a primary cause of error. Such errors could have been avoided by emphasizing the key information in the task, such as the keyword 'relevant' and the number 4.

Analysis by Preference and Usability. We aggregated the participants' response on the system usability questionnaire (see [9] for scoring method) to obtain the overall system usability score. *TextTile* received the usability score of 81.67 out of 100, equivalent to grade *A* in usability, based on the standard SUS percentile score that considers average usability score of systems to be 68.

Analyzing the open-ended response about the features of the system that were easiest/hardest to understand and use, we found that *maps* were one of the hardest components to use, as also discussed in **analysis by task** section. For the easiest features, we received a set of mixed responses. While some participants considered specific operations, such as *filter*, *split* or *summarize* to be the easiest, others mentioned specific interaction and visualization strategies to be the most beneficial, such as the *drag-and-drop* interaction or *column-based* visual summaries for comparison. The overall preference score, based on participants' response was 4 [3.58, 4.42].

In the last section of the post-study questionnaire, 6 out of 12 participants mentioned that they have built or used custom solutions, built over months of development effort, to solve similar analytical tasks in the past. 1 participant mentioned that he is not aware of any system that can help him analyze both, structured data and unstructured text, together. 10 out of 12 participants said that they would prefer using *TextTile* over other systems, and mentioned the 'need to code complex flows in other systems' as one of the primary reasons behind this choice. One of the two participants who said that they would prefer other systems over *TextTile* mentioned the limitation of *split* operation – that it allows segmentation on only one attribute at a time – as the primary reason for preferring other custom tools. We plan to implement support for multiple attributes in the *split* operation, supporting a broader range of analyses that require multi-level segmentation.

7 Conclusions, Limitations and Further Work

In this paper, we presented *TextTile*, a data visualization tool for seamless exploration of structured data and unstructured text. We have shown how the system can be used to answer a varied set of important questions analysts have and struggle to answer with existing methods. Through use cases and a user study we have also shown the capabilities of the tool and its effectiveness. The work we have presented has the following limitations we intend to overcome in the future works.

First, *TextTile* needs to be deployed in real-world settings and evaluated according to how domain experts would integrate it and use it in their day-to-day activities. Following longitudinal methodologies such as those suggested by Shneiderman and Plaisant in their MILC process [36], we would like to perform longitudinal analysis of *TextTile* when used for a prolonged time in working environments. Currently, a development organization, two newsrooms, and two software firms are using *TextTile* to analyze their data, as we wait for their feedback and recommendations.

Second, *TextTile* includes only basic, though powerful, natural language processing method to create keyword summaries out of text. These can be largely expanded and improved using existing or new methods derived from natural language processing (NLP) research [25]. One particularly important improvement is the implementation of methods that reduce keyword semantic redundancy (i.e., different words that carry the same meaning) and rank the words according to how semantically relevant they are (that is, they carry useful information). We also plan to experiment with bi-grams and phrases and mechanisms to let the user decide what type of keywords he or she wants to see (e.g., using part-of-speech or entity extraction methods [25]).

Third, in the current version the system allows splitting only by one single field at a time, but as we have seen in our validation, splitting by more attributes may be desirable in some circumstances. This design choice was made to reduce complexity and maximize learnability, but we now realize it may indeed make sense to make it available. As part of our future work, we will investigate this issue in more detail and experiment with solutions that allow splitting with multiple attributes with minimal impact on the ease of use.

Fourth, as outlined in Section 5.2, we developed a visual representation alternative to word clouds due to their ineffectiveness for data analysis tasks. Our solution needs to be expanded and thoroughly validated through controlled experiment, which we intend to perform in the near future.

Fifth, while we believe that the questions we gathered and the task abstraction we derived are a representative sample of the type of questions analysts may ask, this subject still needing to be explored in depth in order to produce a more complete and accurate description of tasks that require relating text and structured data.

8 Acknowledgments

We thank Stephen Few and Hendrik Strobelt for their invaluable feedback on early versions of this manuscript. This work is supported in part by the Knight Foundation's Prototype Fund and CAPES Foundation, Ministry of Education of Brazil - process: BEX 13235/13-3.

References

[1] Apache Lucene. https://lucene.apache.org. Accessed 2016-6-27.

[2] Dollars for Doctors. http://www.propublica.org/series/dollars-for-docs. Accessed 2016-3-30.

[3] Elasticsearch. https://www.elastic.co. Accessed 2016-6-26.

[4] Surgeon Scorecard. https://projects.propublica.org/surgeons/. Accessed 2016-3-30.

[5] WHS Explorer. http://nyuvis.github.io/whs/. Accessed 2016-3-30.

[6] World Humanitarian Summit (Office for the Coordination of Humanitarian Affairs). https://www.worldhumanitariansummit.org. Accessed 2016-3-30.

[7] Yelp. http://www.yelp.com. Accessed 2016-3-30.

[8] S. Bateman, C. Gutwin, and M. Nacenta. Seeing things in the clouds: the effect of visual features on tag cloud selections. In *Proceedings of the Conference on Hypertext and Hypermedia (HT)*, pages 193–202. ACM, 2008.

[9] J. Brooke et al. SUS-A quick and dirty usability scale. In P. W. Jordan, B. Thomas, I. L. McClelland, and B. Weerdmeester, editors, *Usability evaluation in industry*, chapter 21, pages 189–194. CRC Press, 1996.

[10] R. L. Cilibrasi and P. M. Vitanyi. The Google Similarity Distance. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 19(3):370–383, 2007.

[11] C. Collins, F. B. Viegas, and M. Wattenberg. Parallel tag clouds to explore and analyze faceted text corpora. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 91–98. IEEE, 2009.

[12] W. Dou, X. Wang, D. Skau, W. Ribarsky, and M. X. Zhou. Leadline: Interactive visual analysis of text data through event identification and exploration. In *Proceedings of IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 93–102. IEEE, 2012.

[13] C. Felix, A. V. Pandey, E. Bertini, C. Ornstein, and S. Klein. RevEx: Visual Investigative Journalism with A Million Healthcare Reviews. In *Proceedings of Computation+Journalism Symposium (CJ)*, 2015.

[14] N. Ferreira, L. Lins, D. Fink, S. Kelling, C. Wood, J. Freire, and C. Silva. BirdVis: Visualizing and understanding bird populations. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2374–2383, 2011.

[15] M. J. Halvey and M. T. Keane. An assessment of tag presentation techniques. In *Proceedings of the International Conference on World Wide Web (WWW)*, pages 1313–1314, 2007.

[16] S. Havre, B. Hetzler, and L. Nowell. ThemeRiver: Visualizing theme changes over time. In *Proceedings of IEEE Symposium on Information Visualization (InfoVis)*, pages 115–123. IEEE, 2000.

[17] M. Hearst. *Search User Interfaces*. Cambridge University Press, 2009.

[18] M. A. Hearst. Whats Up with Tag Clouds? *Visual Business Intelligence Newsletter*, 2008.

[19] E. Hoque and G. Carenini. ConVis: A Visual Text Analytic System for Exploring Blog Conversations. *Computer Graphics Forum*, 33(3):221–230, 2014.

[20] S. Kumar, G. Barbier, M. A. Abbasi, and H. Liu. TweetTracker: An Analysis Tool for Humanitarian and Disaster Relief. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*, 2011.

[21] B. Liu. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167, 2012.

[22] B. Liu, M. Hu, and J. Cheng. Opinion observer: analyzing and comparing opinions on the web. In *Proceedings of the International Conference on World Wide Web (WWW)*, pages 342–351, 2005.

[23] S. Lohmann, J. Ziegler, and L. Tetzlaff. Comparison of tag cloud layouts: Task-related performance and visual exploration. In *Proceeding IFIP International Conference on Human-Computer Interaction (INTERACT)*, pages 392–404. 2009.

[24] D. Luo, J. Yang, M. Krstajic, W. Ribarsky, and D. Keim. EventRiver: Visually Exploring Text Collections with Temporal References. *IEEE Transactions on Visualization and Computer Graphics*, 18(1):93–105, 2012.

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

[26] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP Natural Language Processing Toolkit. *Annual Meeting of the Association for Computational Linguistics: System Demonstrations (ACL)*, page 55, 2014.

[27] A. Marcus, M. S. Bernstein, O. Badar, D. R. Karger, S. Madden, and R. C. Miller. Twitinfo: aggregating and visualizing microblogs for event exploration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 227–236. ACM, 2011.

[28] F. Morstatter, S. Kumar, H. Liu, and R. Maciejewski. Understanding Twitter Data with TweetXplorer. In *Proceedings of the SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1482–1485. ACM, 2013.

[29] T. Munzner. *Visualization Analysis and Design*. CRC Press, 2014.

[30] A. W. Rivadeneira, D. M. Gruen, M. J. Muller, and D. R. Millen. Getting our head in the clouds: toward evaluation studies of tagclouds. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 995–998. ACM, 2007.

[31] C. Rohrdantz, M. C. Hao, U. Dayal, L.-E. Haug, and D. A. Keim. Feature-based visual sentiment analysis of text document streams. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(2):26, 2012.

[32] A. Satyanarayan and J. Heer. Lyra: An interactive visualization design environment. In *Computer Graphics Forum*, volume 33, pages 351–360. EG/IEEE, 2014.

[33] A. Scharl, A. Hubmann-Haidvogel, A. Weichselbraun, H.-P. Lang, and M. Sabou. Media Watch on Climate Change–Visual Analytics for Aggregating and Managing Environmental Knowledge from Online Sources. In *Proceedings of the Hawaii International Conference on System Sciences (HICSS)*, pages 955–964, 2013.

[34] J. Schrammel, S. Deutsch, and M. Tscheligi. Visual search strategies of tag clouds-results from an eyetracking study. In *Proceeding of IFIP International Conference on Human-Computer Interaction (INTERACT)*, pages 819–831. 2009.

[35] B. Shneiderman. Direct manipulation: A step beyond programming languages. In P. W. Jordan, B. Thomas, I. L. McClelland, and B. Weerdmeester, editors, *Sparks of Innovation in Human-Computer Interaction*, chapter 1. Intellect Ltd, 1993.

[36] B. Shneiderman and C. Plaisant. Strategies for evaluating information visualization tools: multi-dimensional in-depth long-term case studies. In *Proceedings of the AVI workshop on BEyond time and errors: novel evaluation methods for information visualization (BELIV)*, pages 1–7. ACM, 2006.

[37] A. Slingsby, J. Dykes, and J. Wood. Configuring hierarchical layouts to address research questions. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):977–984, 2009.

[38] A. J. Soto, R. Kiros, V. Kešelj, and E. Milios. Exploratory Visual Analysis and Interactive Pattern Extraction from Semi-Structured Data. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 5(3):16, 2015.

[39] C. Stolte, D. Tang, and P. Hanrahan. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):52–65, 2002.

[40] G. Sun, Y. Wu, S. Liu, T.-Q. Peng, J. J. Zhu, and R. Liang. EvoRiver: Visual analysis of topic coopetition on social media. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1753–1762, 2014.

[41] 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 *Proceedings of the SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 153–162. ACM, 2010.

[42] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):649–658, 2016.

[43] Y. Wu, F. Wei, S. Liu, N. Au, W. Cui, H. Zhou, and H. Qu. OpinionSeer: interactive visualization of hotel customer feedback. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1109–1118, 2010.