Incorporating Proper Nouns into a Web-Based Document Visualization

Undergraduate Honours Thesis
Faculty of Science (Computing Science)
University of Ontario Institute of Technology

By: Brittany Kondo

Supervisor: Dr. Christopher Collins

April 8th 2012

Abstract

With internet use and online data sharing becoming more and more ubiquitous, the web becomes a desirable medium for deploying all different types of applications. In the field of information visualization, the need for web-based document content visualization tools directly coincides with the digitization of paper-based texts. Such tools provide visual aids to convey meaningful word-based patterns and relationships, promoting information discovery within textual content. DocuBurst is a web application originally presenting hierarchically structured nouns conveying a hyponymy relationship and revealing the noun occurrences in the document. The main purpose of DocuBurst is to present a visual synopsis of a document and assist the user in further exploration of its content. The limitations of restricting DocuBurst to presenting only general nouns is seen through visualizing entity-rich texts such as biographies. Nouns encapsulate broader concepts usually referring to several specific instances. A proper noun can overcome these limitations by providing sufficient levels of detail to expose more specificity.

Relevant entities have been integrated into the existing interface as an additional visualization which is coordinated with the current visualizations. Several design techniques are used as methods of coordination to maintain visual consistency of the interface and introduce a stronger sense of connectivity among separate data elements. Such techniques include weighted brushing to indicate co-occurrences among terms in the document and dynamic filtering capabilities for the entity set presented in the view. Correspondingly, the user interactions implemented are mouse hovering to temporarily reveal co-occurrence highlighting and mouse clicking to select terms of interest as well as filtering entity sets. The goal of the new interface design is to present the user with a visual summary of the document consisting of separate language components and unifying these components to allow quick interpretation of relatedness among data elements in a single view, prior to deciding on areas of interest in the document. With the enriched interactivity to alter visual state of the interface and additional informative data elements, DocuBurst can emerge on the web as a multi-purpose document visualization tool.

Table of Contents

List of Figures

Section 1: Introduction

With the expansion and popularity of the internet, the online community grows each and every day. To compliment this growth, a diverse collection of online texts ranging from e-books to everyday newspapers are becoming readily accessible. Prior to reading these texts, a reader may use a variety of different methods to obtain an overview or summary of the text, the most common being a quick scan through the text to single out key words or phrases that may trigger the readers interest. Readers are generally interested in seeing the overview or a synopsis of the text before they are willing to explore it more in detail [MN97]. The DocuBurst web application is designed to provide the user with an interactive summary, emphasizing commonly occurring words in the text. Potential uses of this application include literary analysis by digital humanities researchers and document collection summaries. Access to the DocuBurst web application has been in demand by several interested users ranging from the members of the RCMP to eager researchers belonging to the online information visualization community. With such an assorted group of potential users, deploying this web application for public use is greatly desired.

Using the web as a visualization platform opens several opportunities for DocuBurst due to its widespread nature, resulting in a diverse space of users. Diversity in users implies variation in the types of visualized content, uses of the application and in general, main interests or intentions of the users. In correspondence to this miscellany, the DocuBurst web application should provide the tools necessary to visualize and analyze document content from a broad text-genre space and appeal to users with varying expertise or abilities. The existing application can be altered to create a more detailed synoptic view of the document, providing additional exploration starting points to spark new insights and discoveries of intra-document word patterns and relationships. Specifically, in terms of text-based analytical capabilities, DocuBurst is limited to noun extraction. A noun-based

visualization of the biography of a famous hockey player would display terms such as “player” or “person”, disregarding the more specific terms, namely hockey player or team names, which can carry valuable or significant information. Depending on the reader’s intentions, presenting the noun “person” may be meaningless, especially if the document contains multiple references to several people. The reader may be curious to know the specific people’s names to which the “person” noun is referring to. Here, a potential area of weakness for DocuBurst is demonstrated. In entity-rich texts, there may be a desire for more specificity. Named entities are descriptive terms classified into several broader categories such as people, geographical locations, institutions, companies etc. In contrast with named entities, nouns indicate a general form or classification of an entity. Extracting and displaying relevant entities from a text will present the user with richer information.

The main contributions of this work include: integrating named entities into the DocuBurst application, enhancing interactivity with weighted brushing and dynamic filtering design techniques and formulating a usability evaluation plan.

Section 2: Problem Statement and Motivation

As discussed in [Col10], proper nouns were not originally integrated into the DocuBurst visualization because they had no place in the current hierarchical structure. However, proper nouns carry valuable information pertaining to details unseen through nouns. Adding this new data type into the current application could enhance its utility [Col10]. Nouns expose more general terms representing a thematic characteristic of the text and entities represent the detailed references of nouns. Certain text genres benefitting from entity extraction are less fictitious and more real world oriented including travel guides, biographies, news articles, social media archives etc. In terms of more classical literature, such as novels, people related entity extraction can expose

additional information about specific characters. As a web application, it is important for DocuBurst to cover a wide range of input data types.

A challenge presented when incorporating multiple visual elements encoding different data types into a single view is maintaining simplicity, without sacrificing each data's individual importance. All data elements should operate well together to serve the user's desired task. Therefore, placing a visualization containing new data elements into the interface requires some type of harmonization with respect to visual presentation and connectivity across data elements. Currently the DocuBurst operates synchronously with the paragraph viewing visualization (Figure 2, component A). The added entity visualization must maintain this synchronization. For the entities to have any significance in the existing application, a useful relationship between entities and the synsets of the DocuBurst must also be portrayed.

Section 3: Goals

An additional visualization used for displaying relevant named entities can be added to the interface and coordinated to work synchronously with the existing visualizations. When multiple visualizations are coordinated, any action performed on a single visualization which induces a change of state (i.e., a filtering operation) causes the other visualizations to be updated accordingly. The method used for visually displaying connectivity among the visualizations is co-occurrence based coordinated highlighting and related entity filtering via user interaction. Thus, the user is granted control of any new visual state change occurring within the interface. The new design for DocuBurst should carefully consider usability by incorporating uniform action sequences, easily escapable modes and place the user in a situation free of confusion or uncertainty.

In summary, incorporating named entities into the existing application involves:

- Adding automated entity extraction
- Enhancing usability of the existing application with design and interface layout modifications
- Integrating and visually connecting the entities with the existing visualizations using multiple view coordination and a weighted brushing technique representative of co-occurrences among data elements.
- Dynamically filtering the entity data set based on co-occurrences
- Implementing user interactions for controlling filtering and co-occurrence highlighting

Section 4: Background on DocuBurst

The current DocuBurst web application provides an interactive visualization of user-selected texts that can be used for synoptic or comparative purposes. The user may upload text and interact with the generated DocuBurst visualization which displays a collection of words found in the text, emphasizing the more commonly occurring terms. This visualization component follows a radial space-filling tree layout which demonstrates the hierarchical IS-A (hyponymy) relationship between the nouns displayed on each wedge. These nouns are extracted from WordNet, a lexical English database consisting of nouns which are grouped into "synsets" based on a shared concept or meaning. The underlying linguistic processing of the document involves dividing the text into segments, identifying the nouns and verbs (referred to as *parts of speech*), stemming these parts of speech (i.e., sailors->sailor, growing->grow) with the assistance of a morphology database table and lastly searching for these words in the WordNet database [CCP09]. A lemma represents the canonical form of a set of related words. For example, in the set {walking, walked, walk}, the lemma is "walk". In the DocuBurst visualization, a lemma is the member of the synset selected and displayed on the node's label.

Depending on the occurrence of the extracted words in the document, a count is computed. The resulting word, part of speech and count value is then sent to the web application's front end for rendering into a radial space-filling diagram, also referred to as a "sunburst" diagram. The visual encoding of each node varies in both angular width and colour transparency. Angular width can be proportional to the number of leaves in the sub-tree of the node or proportional to the sum of word counts in the sub-tree of the node [CCP09]. The transparency of the node colour is related to how prominent the corresponding synset is in the document.

There are two layouts for which the user may view the DocuBurst. The first is a single-node view, where only the words contained in the document are coloured. The second view is cumulative, where word counts of each leaf node are propagated towards the parent node. There is also a comparative view where the user selects two texts which are both integrated into a single DocuBurst visualization and colours are used to differentiate which document the synsets exist in. Other interactive features of the web application include node selection to view the usage of the synset in the text, a dynamic keyword filtering tool, layout reconfiguration to a different root node and browsing the document's content one text-segment at a time (Figure 1). An additional visualization is provided which works synchronously with the DocuBurst and is intended to be used for navigation through the document. It is a collection of tiles representing text segments or approximate "paragraphs". This visualization is also used to display the locations and occurrence frequencies of selected DocuBurst synset nodes in the text.

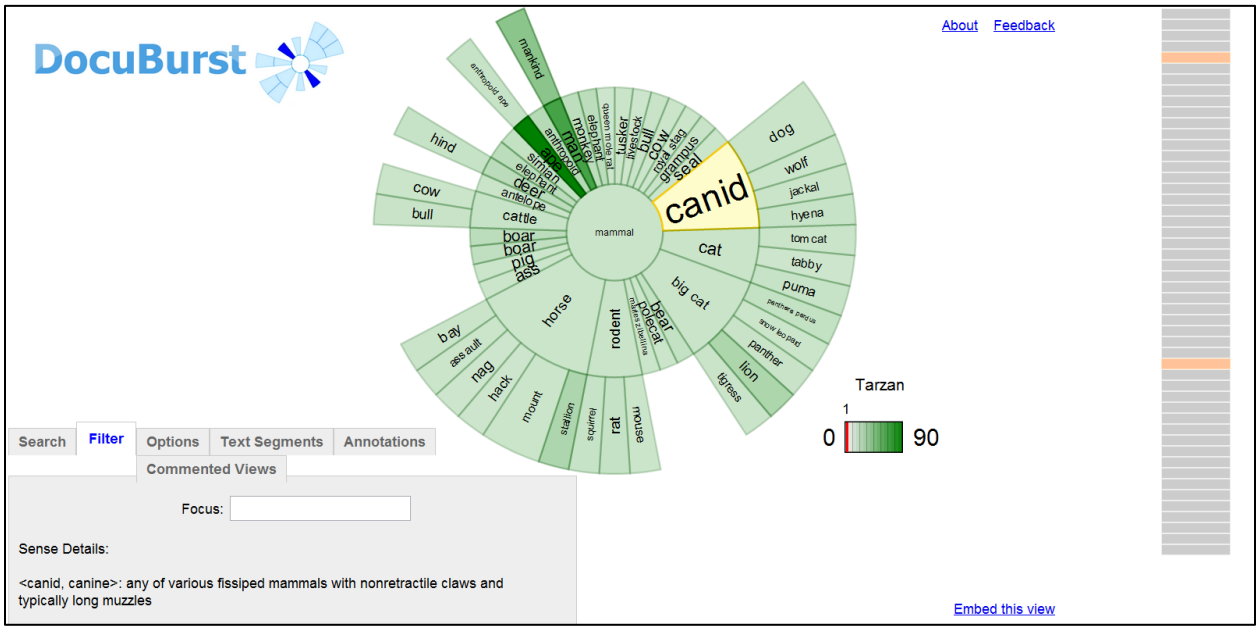

Figure 1: A screenshot of the original DocuBurst application

Section 5: Related Works

More and more visualization applications are migrating to the web. The web enables rapid sharing and exposure to a wider audience than what would be possible with a non-web based application. The web implementation of "Document Arc Diagrams" accepts user inputted text and generates an interactive arc diagram showing the internal similarity structure of the document by joining segments of text containing similar words with an arc [CW07]. "A Text Visualization Tool" is another existing online document visualization tool which uses an interactive graph structure to display relationships between key words in a document along with other widgets included to provide overviews, usage of words in the actual text and distribution of the words across the document [Cla07]. Comparably, DocuBurst extracts and visualizes meaningful language components from user provided texts, permitting the user to explore occurrences of multiple terms in the document.

VisGets is a web-based application used for visual exploration of large collections of web resources through dynamic query construction performed by user interaction across three dimensions, namely time, location and topical tags [CCDW08]. The information space of Vis-Gets is a collection of information residing within the web, whereas DocuBurst currently relies on information provided directly from the users. The design and interactivity techniques presented in VisGets have been incorporated into the new design of the DocuBurst application (see Section 6: Design). Some notable techniques include: weighted brushing to visually highlight relatedness among data elements across the multiple visualizations and filtering the data presently displayed. By incorporating these design techniques, DocuBurst is transformed into an enhanced interactive visual tool for document content analysis. In [CCDW08], "exploratory search" and "berry-picking", two concepts describing the initial state of possessing minimal information needs followed by engagement in a learning process, are discussed. The interactive nature and initial synoptic view of DocuBurst can assist in beginning exploration and promoting information discovery within the document, formulating new learning experiences.

Many Eyes is another web-based discovery tool taking advantage of the web's inherent collaborative nature by allowing users to upload data, create visualizations, save and share specific views of the visualizations and contribute comments or annotations on visualizations [KMVVW07]. Similarly, DocuBurst provides functionalities for anonymously commenting on and annotating specific views of the visualization and an embedding functionality used for saving a view and possibly sharing it with others. In [KMVVW07], there is mention of "collective intelligence" as an important social feature, allowing users to gain new perspectives on data sets by interpreting the visualization under the influence of others. Collaboration in visualization creation is an important requirement of a web-based application such as DocuBurst because it can help with analysis of content, introducing variation under the influence of interpretations from different perspectives. Using the web as a medium for deploying applications, automatically results in widespread data

exchange and sharing due to its massive scale, popularity and social networking components. The Many Eyes web application provides visualizations which are both useful and appealing to all different users, even those without visualization expertise [KMVVW07]. Comparably, in DocuBurst, the additional step of entity extraction provides a more detailed synopsis applicable to a broader range of text genres, extending its coverage of potential user interests.

Moving away from the web, [CKPV09] discusses the importance of automated entity extraction from texts for literary analysis mainly for the purpose of studying relations among characters or topics and vocabulary discussed near character references in texts. POSVis is a tool designed to facilitate this type of analysis [CKPV09]. Unlike DocuBurst, it does not visualize hierarchically structured parts of speech, but rather places all parts of speech in an interactive word cloud. Since the focus of POSVis is on character references, a network diagram centralized at a selected entity is used to visualize entity co-occurrences explicitly showing character relationships. Similar to DocuBurst, the user is able to freely explore the raw text segments and view the occurrences of all selected words. POSVis is limited to people-related entities in order to appropriately serve the intended purpose of character-based analysis. DocuBurst extends this limitation by extracting entities ranging across different categories, where the highest occurring or most relevant entities are displayed in the view. This way, the entities extracted in DocuBurst, can support several different types of texts and be used for examining a variety of relationships among words.

Section 6: Design

Section 6.1: Automated Entity Extraction with Open Calais

One limitation of the WordNet database is that it does not contain named entities. Open Calais is a web service which automatically extracts a wide range of named entities from user

submitted texts. The main advantage of using Open Calais is that it provides automatic and cost-free entity extraction. For each entity extracted, some useful information is returned including an occurrence count and the category the entity is classified under (See Appendix A). The occurrence count is used to compute the overall occurrence score of each entity in the document as well as the occurrence score in each paragraph it occurs in. These scores are required for implementing weighted brushing. Another automated feature offered by Open Calais is entity disambiguation, an attempt to resolve the identity of an entity instance. An entity is considered ambiguous if it is referenced in multiple different ways or if a group of entities take on similar names [Reu11]. Open Calais uses a set of unique entities to map each ambiguous entity. If an entity can't be precisely mapped to a word in the set, then it is assigned a relevance score and mapped to the word it is most related to. To aid with disambiguation, the surrounding text can be searched for helpful contextual clues. For instance, resolving an ambiguous company name can be done by searching for the surrounding text for a reference to the company's location [Reu11].

The limitation of entity disambiguation provided by Open Calais is that it performs disambiguation on only three types of entity categories: company names, geographical locations and product (electronics) names. Entity disambiguation is important because it increases the accuracy of entity extraction. However, as with word sense disambiguation, it can be a challenging task because it requires interpreting the text's meaning and an exhaustive entity identifier reference set.

Section 6.2: Visualization

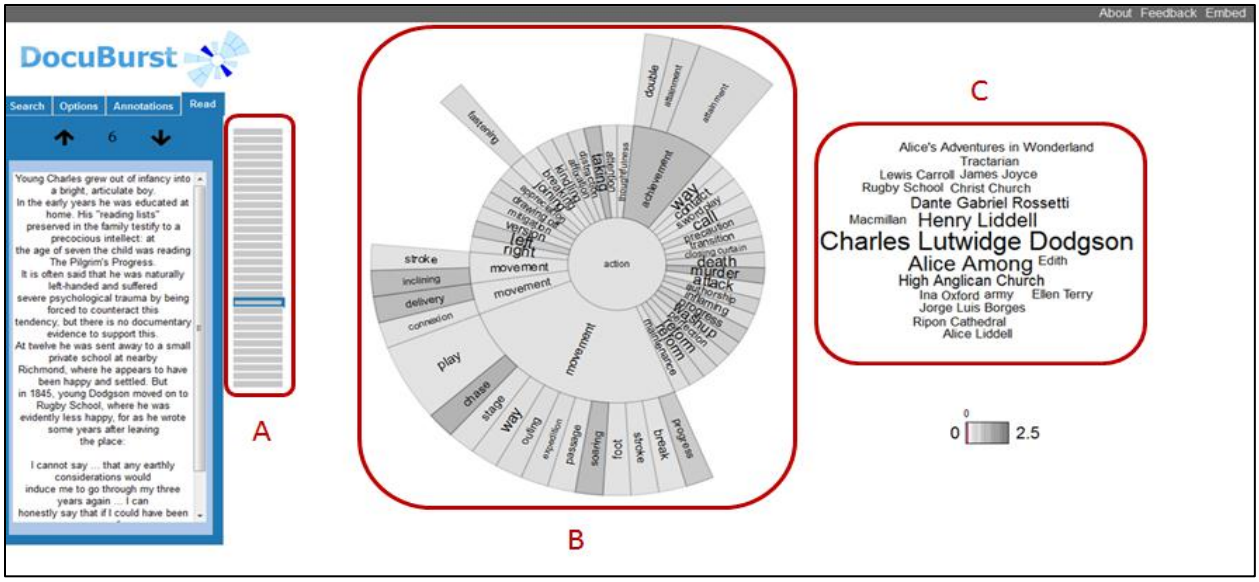

Figure 2: Three visualizations: A) The tile browser, B) The DocuBurst (sunburst) and C) The entity word cloud.

Section 6.2.1: Preliminary Usability Improvements

Prior to incorporating entity extraction and new interactive features, some usability enhancing modifications were made to the existing interface (Figure 2). First, a variety of aesthetic improvements were done to make each portion of the website more visually appealing. The next modification was re-designing the tabbed widget which previously was located at the bottom of the screen (Figure 1). An important use of this widget is reading the individual text segments of the document which are closely coupled with the tile browser, located on the right hand side. To remove the physical separation of these two related widgets, the tabbed widget and the tile browser were moved to the left hand side which eases the task of reading and navigating through the document using the tile browser. A reading indicator was added to the tile browser to display

the user's current location in the document to improve visibility of system's status [Nie05], a violated usability heuristic.

Section 6.2.2: Applying Visual Design Principles

The new design of the interface abides by Shneiderman's visualization mantra: "overview first, zoom and filter, then details on demand" [Shn96]. Initially, the user is presented with a synoptic view of the document. By filtering co-occurring entities in the entity word cloud and hovering over interested terms in all three visualizations, the user is able to locate and concentrate on an area of interest. Through interaction and selection of entities or nodes on the sunburst visualization, the user is able to closely examine specific details of the selection in the document's text by locating the actual occurrence of a selected word.

To visually encode co-occurrences among data elements across the three visualizations, colour with varying transparency is used. The pre-attentive effect of using colour as a visual variable places the desired emphasis on co-occurrences. Varying the transparency of colour coincides with the strength of co-occurrence. Since the added highlights are coloured, a decision was made to re-colour the sunburst visualization to a grey scale. The main reason for this change was to limit the number of different colours present in the interface, reducing confusion for the user and drawing more attention to co-occurrence highlighting as an important change of visual state. Additionally, a different colour is used to indicate a selected data element, to clearly remind the user of which element is currently selected, making it easier to undo selections and restore the system to its original visual state.

For a user interface to be predictable and easily controllable by the user, it must provide direct manipulation and immediate feedback [Shn97]. Satisfied users wish for mastery of the interface, ease of use and a desire to continue exploring more powerful features of the system

[Shn97]. To attain this, the DocuBurst application provides constant visibility and highlighting of data elements of interest, includes easily reversible actions and allow the user to instantiate simple action sequences to accomplish tasks.

Section 6.2.3: VisGets-Inspired Design Techniques

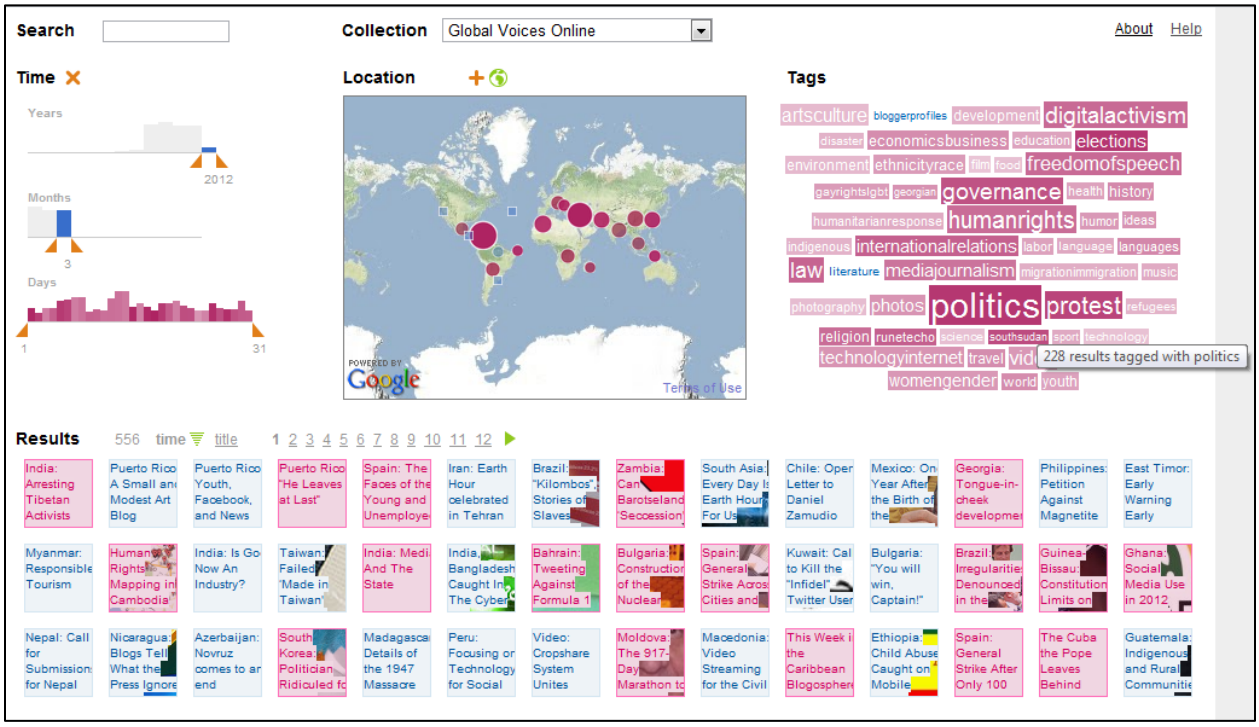

Figure 3: Screenshot from the live demo of VisGets [Dor08]

Section 6.2.3.1: Faceted Navigation

Faceted Navigation is typically used to instantiate and then facilitate search by presenting the user with higher level topics representative of the data, allowing the user to select a topic of interest to narrow down their search. In the DocuBurst application, this technique is used in the entity word cloud as a mechanism for filtering the entity set to contain only related entities. Although used in a less prominent form here, faceted navigation provides a way for the user to physically filter entities based on an entity of interest.

Section 6.2.3.2: Weighted Brushing

As in VisGets, weighted brushing in DocuBurst is used to reveal degree of relatedness between data elements (Figure 3). The main reason for using weighted brushing is to temporarily reveal relations within a visualization and across the other visualizations. To achieve the effect of weighted brushing, the strength of co-occurrence (representing relatedness), is encoded by varying the transparency of the highlighting colour. The hovered element is automatically assigned full opacity, because it is the most related to itself. All other elements within the visualization and across the other visualizations are re-coloured with a transparency based on the strength their co-occurrence with the hovered element. Therefore, if an element never co-occurs with another element it will not be highlighted with colour. The variance in transparency allows the user to instantly view the data elements which are related and their strength of relation to the hovered element of interest.

Section 6.2.3.3: Multiple Coordinated Views

Having three disjoint visualizations in the DocuBurst application would restrict the user to viewing only relations between data elements within a single visualization. Providing visual coordination across multiple views shows interdependencies among the data sets contained in each view [CCDW08]. Effective multiple view coordination is dependent on instantaneous changes across all views reflective of a user interaction, such as selection, applied to an element. VisGets presents the dynamic coordination of multiple visualizations through synchronous visual changes as the user interacts with any of the visualizations. Similarly, hovering over any visualization data element in the DocuBurst application, results in highlighting all co-occurring elements present on the screen.

Section 6.2.4: The Word Cloud

A word cloud was selected to display the most frequently occurring entities related to the current root and its children presented in the DocuBurst diagram. For practicality, only the top twenty entities from varying categories are selected based on frequency of occurrence in the document. A word cloud is a simple visualization for displaying an overview or summary of a collection of words with varying frequencies. Conventionally, the font size of the displayed entity word encodes the occurrence count of the entity and the word placement within the layout is in no particular order. The word cloud layout chosen for this application is correspondent to a popular text visualization called Wordle. Although seemingly similar to a generic tag cloud, Wordle's placement algorithm is based on the approximate bounding curves of the text, resulting in a tightly packed layout. This layout is more space efficient and exhibits enhanced visual appeal by eliminating unnecessary white space referred to the designers of Wordle as the "awkward ransom note" effect [FVW09]. The original Wordle word arrangement includes rotated orientations of the words. However, for this application readability of the text is very important, thus all words are horizontally orientated.

Section 6.3: Interaction

A useful portion of the DocuBurst application's new design is interactivity. The user interactions were designed with the intent of maintaining consistency and serving the purpose of dynamically connecting all visual elements on screen. In general, the user interactions available are mouse clicking for selection and mouse hovering for invoking co-occurrence highlighting. Mouse hovering on the word cloud and sunburst visualization is always active, regardless of any selections. Hovering over any data element highlights all co-occurring elements across the three

visualizations, temporarily revealing how related other data elements are to the hovered element of interest (Figures 4-6).

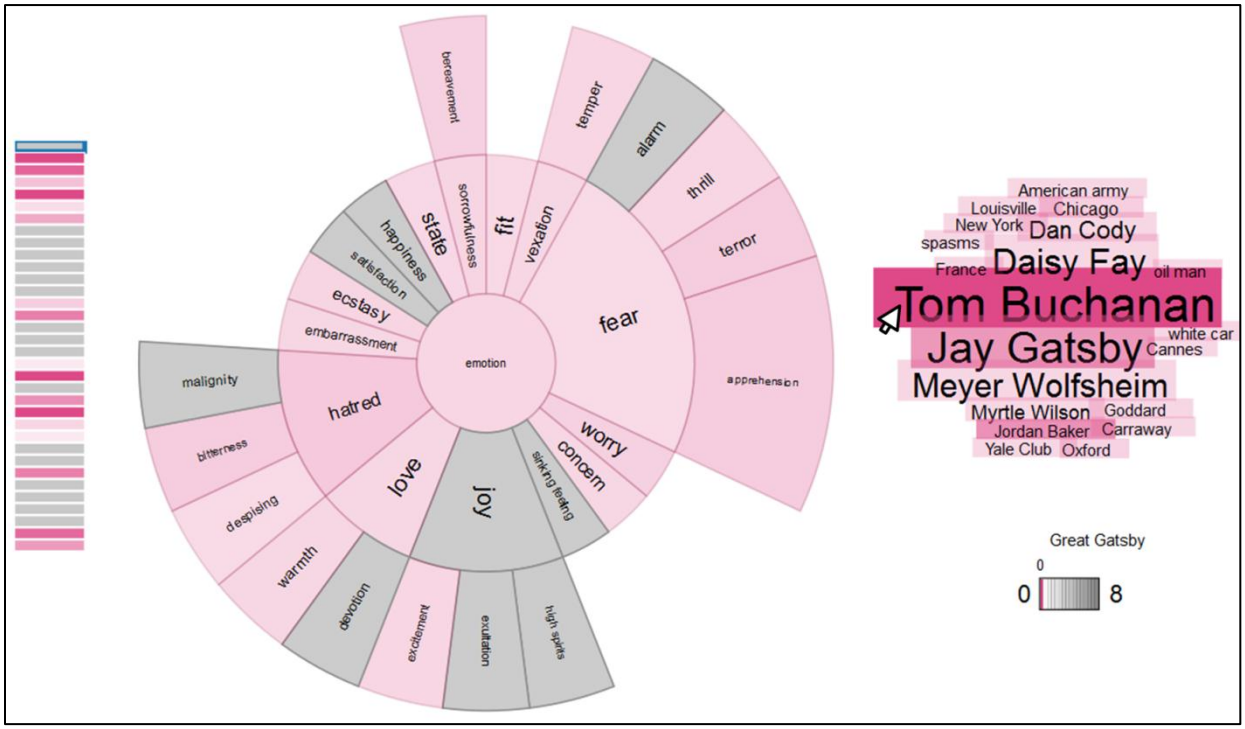

Figure 4: Hovering over "Tom Buchanan" to reveal co-occurrences between other entities, synsets and occurrences in paragraphs.

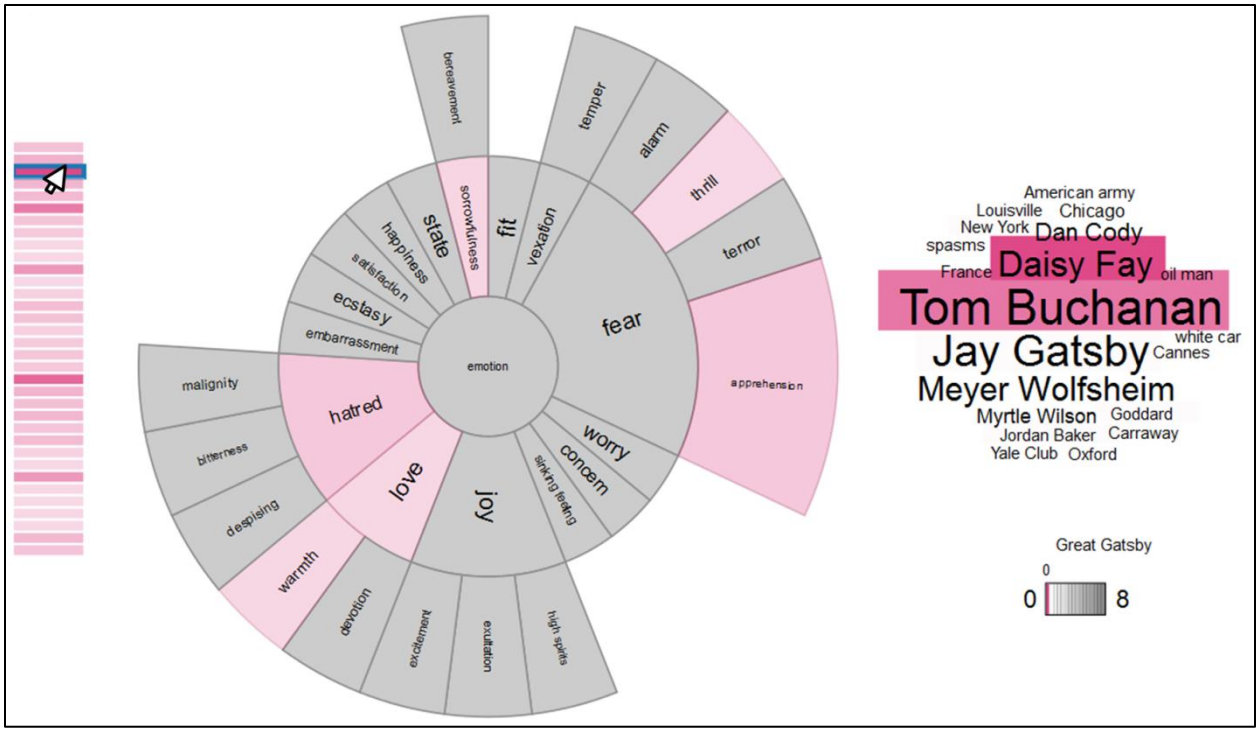

Figure 5: Hovering over the third paragraph to reveal other paragraphs with similar terms as well as the entities and synsets found in this paragraph.

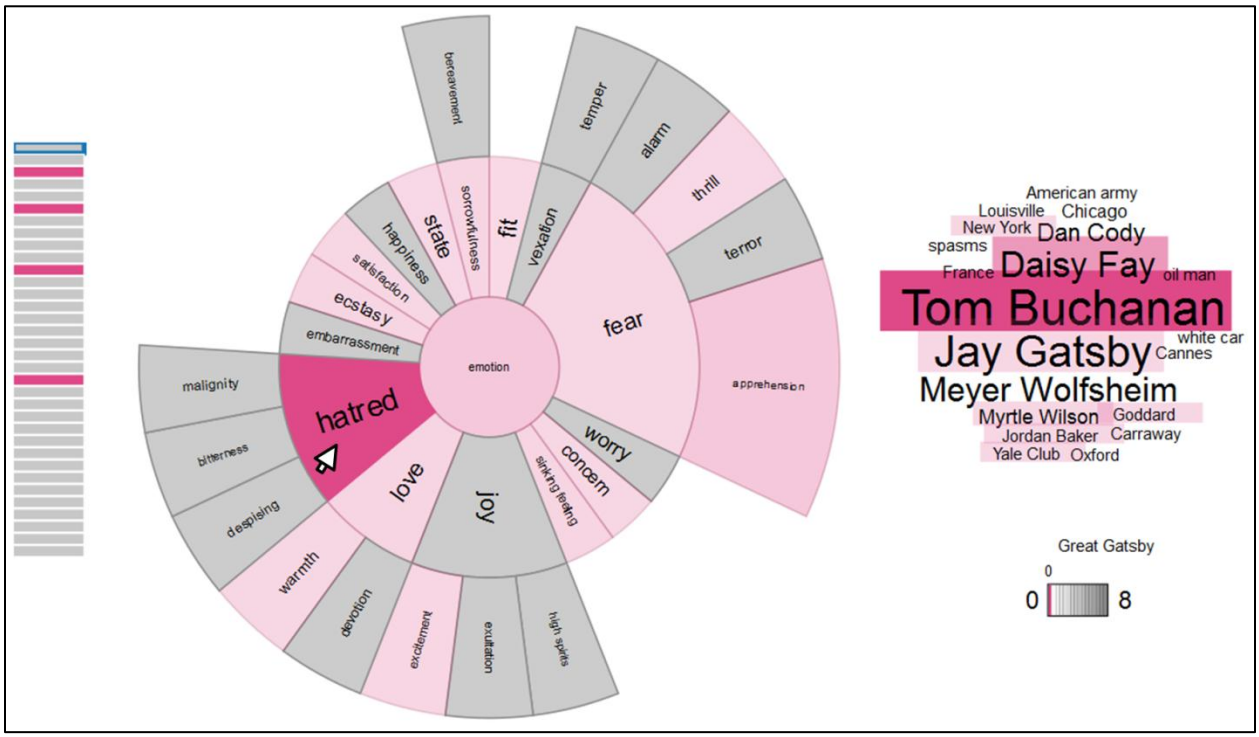

Figure 6: Hovering over "hatred" to reveal the paragraphs it occurs in as well as the co-occurring entities and other synsets.

Selecting an element in the word cloud or sunburst results in slightly different visual effects. The word cloud is the only visualization enabling filtering, which involves removing elements from its data set based on selection(s). Filtering on the DocuBurst is not possible due to its hierarchical structure. Similarly, filtering the tile browser is unsuitable due to interference with reading the document's content. Selecting an entity on the word cloud will result in repopulating its content to contain all other co-occurring entities. Therefore, selection is used for viewing only related entities. Entity selection does not affect the sunburst, however since hovering is never de-activated, the user is still able to view co-occurrences via coloured highlighting. The tile browser widget always serves as a result set display for selections. Whenever an entity is selected, the tile browser is re-coloured to show its occurrences in the document assisting the user in easily locating and browsing text segments containing words of interest.

Selecting a lemma from the sunburst will trigger a filter on the word cloud to contain only entities co-occurring with the selected lemma. Similar to entity selection, a lemma selection causes the tile browser to colour all paragraphs the lemma occurs in. Additionally, the sunburst enables right mouse click for selecting a lemma sub-tree. A right click will cause the other two widgets to behave in the same way as a left click, except the co-occurrence filtering criteria is expanded to the multiple lemmas in the sub-tree (Figure 7).

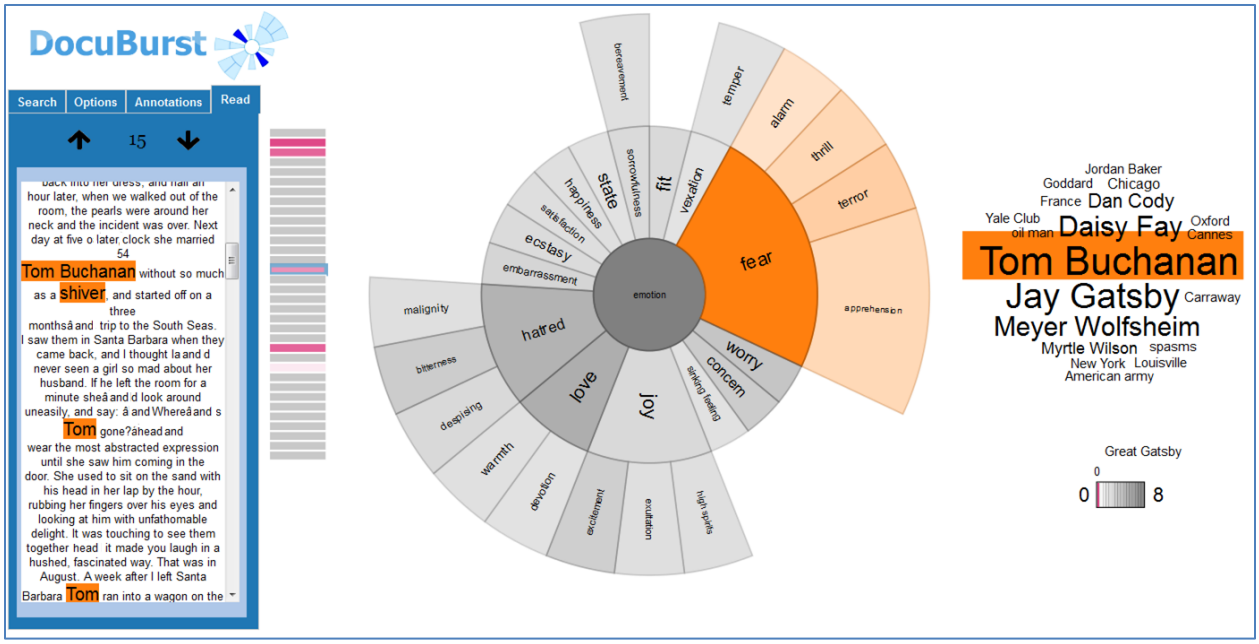

Figure 7: Selections of the "fear" sub-tree and "Jay Gatsby" to reveal paragraphs where the synsets of the "fear" sub-tree and the entity co-occur within. The occurrences of these terms in the document are viewed in the reading widget at the left hand side.

As mentioned previously, the tile browser is used for displaying results. Therefore, a selected paragraph will not trigger any physical filters. The result of left clicking a tile is displaying the actual text contained in the paragraph. A user may still view entities and lemmas contained in a paragraph through hovering over a tile which provides individual content summaries of each paragraph. At most, one entity and one lemma (or lemma sub-tree) may be selected at a time (Figure 8). If this is the case, then the tile browser displays only paragraphs where the selected words co-occur within. Lastly, at any time, the user may restore the entire application to its original visual state by de-selecting all clicked elements and removing the mouse from hovering over any element.

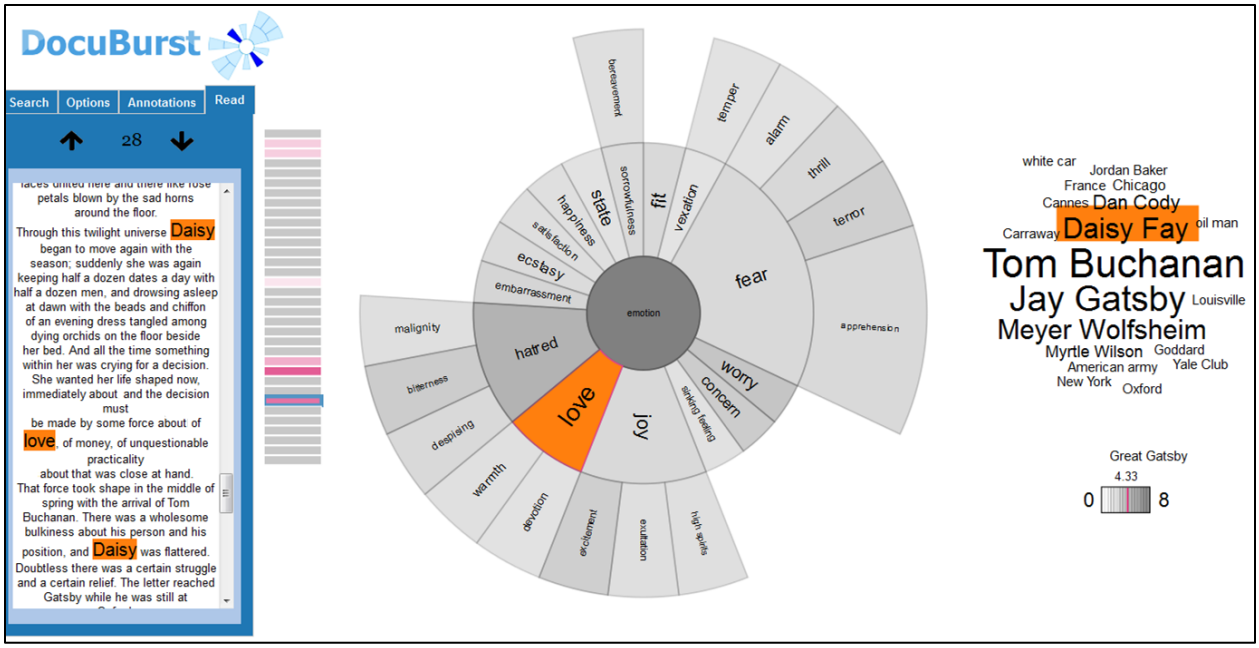

Figure 8: Searching for co-occurrences of the synset "love" and the entity "Daisy Fay" by selecting both "love" and "Daisy Fay".

Introducing interactivity in a user interface is crucial for applications such as DocuBurst whose content manipulation requires user input and encourages further exploration of the generated visual content. The new design strives to maintain consistency across all interactions [PS86], associating a natural feeling with the flow of interactions and minimizing unexpected results whenever an action is performed. By creating instant visual feedback through hovering, it places the user in control of the system by using their actions to trigger changes of visual state [PS86]. Another important characteristic in an interactive visualization design is the flow of interaction sequences, where interactions are carried out naturally and state changes in the system are not distracting but are more subtle. By using mouse hovering induced universal highlighting and minimal movement of data elements during filtering, transitioning between changes in the system's visual state do not draw the user's attention away from the intended task.

Section 6.4: Advantages of the New Design

The new design of the DocuBurst application allows distinct views representative of different data types physically separated but visually connected under the control of the user. The added filter-enable word cloud makes the application comparable to a query-based application, where the user may select a term of interest to narrow down the results. Having user interactions produce similar results across all visualizations is an important heuristic to consider when designing interactive applications. The interface demonstrates uniformity of actions, where a mouse hover always results in revealing co-occurrences, mouse clicking is always a form of selection or filtering and the tile browser always contains the results of a selection, allowing the user to view the outcomes of the actions performed. The advantage of using a transient weighted brushing technique is the user can quickly explore all possible co-occurrences before deciding on a filter action to navigate towards an area of interest. If the highlights were permanent, another selected state is added which may be difficult for the user to identify with and escape from.

The simplicity of the word cloud does not overcomplicate the interface. The sunburst displays a hierarchical relationship among nouns in the text, which is not complex as a stand-alone visualization, but can become complex when alongside other visualizations. This new design is not only intended to provide additional useful data elements, it must constitute as an interface used for quick interpretations of the three coordinated visualizations upon a single glance. A user must be able to see the multiple visualizations as both one large picture as well as individually segmented views. If this interface were extended to portray a more complex entity-based visualization, such as in the POSVis [CKPV09] interface where a node-link diagram is used to explicitly show entity relations, a clear switch of mode must be invoked in order to maintain the ability of having a "single-glance" visual interpretation. Otherwise, the sense of unity among the multiple visualizations would be replaced with the viewer's attempt to first interpret the visualizations separately and then adjoin these interpretations.

Section 7: Implementation

Section 7.1: Data Processing

Section 7.1.1: Integrating the Open Calais Web Service

Any user can register for a free API key to obtain service from Open Calais. The web service is invoked using an HTTP POST Request and the results are obtained from an HTTP Response. The results returned are parsed to extract the required information. For convenience, entity extraction is performed on each text segment of the document. In terms of processing time, this method may not be the most efficient, especially for larger texts. However, since this processing is only performed once when the document is uploaded, the slow-down is not too significant and will not affect performance of the visualization portion of the application.

Section 7.1.2: Additional Entity Grouping

Redundancies in the returned extracted entity set exist and become obvious in heavily occurring entities displayed in the word cloud. Approximate entity grouping was performed in order to establish higher categories of entities and reduce the number of separate entities referring to the same entity. This is done by using regular expression matching to find entity words contained as substrings in larger entity words. The largest word is selected as the "group name" for all other entities occurring as substrings. For example, two occurrences of the same name "Charles Smith" and "Smith" are extracted separately and would exist on the word cloud as separate words if they were heavily occurring in the document. Entity grouping would cause "Smith" to be grouped under "Charles Smith", assuming the two entities refer to the same person. This method is not flawless and will sometimes group unrelated entities under the same entity category, this problem being particularly obvious in people-related entities. A potential solution would be to localize the

occurrences of entities when they are grouped. Assuming if "Charles Smith" and "Smith" occur within the same sentence or small text fragment, they refer to the same person.

Section 7.2: Word Scoring

Section 7.2.1: Entity Occurrence Scoring

Prior to loading the word cloud on the screen, the overall occurrence score of each entity is pre-computed and stored in a database table. This score is obtained by taking the sum of all occurrences of the entity in the entire document and dividing it by the maximally occurring entity's count. This score indicates how frequently an entity occurs in the document and is used for sizing the font of the words in the word cloud. Another pre-computed score is the occurrence of an entity to each paragraph it occurs in. This score is computed in a similar manner by taking the occurrence count of the entity in a paragraph and dividing it by the maximum occurrence count found amongst all entities in the paragraphs they occur in. Unlike the overall occurrence score, this score indicates how heavily an entity occurs in a specific paragraph. Therefore, it is possible for an entity to have a very high overall occurrence score, but not necessarily a high occurrence score in each paragraph it occurs in. In the word cloud, since only a subset of entities are selected, the scores are re-normalized to lie within the range of the maximum score with respect to the words shown in the word cloud.

Section 7.2.2: Co-occurrence Scoring

In linguistics, the term "co-occurrence" simply refers to the proximity of words in a text. The proximity here is defined as within a text segment. Whenever two words occur in the same text segment (approximate paragraph), they co-occur with one another. Two words co-occurring in the document are assumed to be interdependent. In this application, there are six different co-occurrence scores that must be computed and cached for quick retrieval and display on mouse

hover or click. Co-occurrence scores involving paragraphs and words actually indicate an occurrence, since paragraphs cannot co-occur with words. Also, a paragraph and paragraph co-occurrence indicates the amount of common entities and lemmas in the current visualization between the two paragraphs.

Co-occurrence	**Score Calculation**
$lemma_l \; and \; lemma_m \; in \; paragraph_p$	Minimum($lemmaScore_{l,p}$, $lemmaScore_{m,p}$)
$entity_e \; and \; entity_f \; in \; paragraph_p$	Minimum($entityScore_{e,p}$, $entityScore_{f,p}$)
$entity_e \; and \; lemma_l \; in \; paragraph_p$	Minimum($entityScore_{e,p}$, $lemmaScore_{l,p}$)
$entity_e \; and \; lemmaSubtree_s \; in \; paragraph_p$	Minimum($lemmaSubtreeScore_{s,p}$, $entityScore_{e,p}$)
$entity_e \; and \; paragraph_p$	$entityScore_{e,p}$
$lemma_l \; and \; paragraph_p$	$lemmaScore_{l,p}$
$lemmaSubtree_s \; and \; paragraph_p$	$lemmaSubtreeScore_{s,p}$
$paragraph_p \; and \; paragraph_q$	$\frac{entityCount_{p,q} + lemmaCount_{p,q}}{2}$

The occurrence scores used above are calculated for each paragraph the lemma or entity occurs in:

Let $C_{l,p}$ *be the number of occurences of* $lemma_l$ *in* $paragraph_p$ *and* S_l *be the number of synsets containing* $lemma_l$:

$$lemmaScore_{l,p} = \frac{C_{l,p}}{S_l}$$

Let $C_{e,p}$ *be the number of occurences of* $entity_e$ *in* $paragraph_p$ *and* $MaxC_{e,p}$ *be the maximum occurrence count across all* $C_{e,p}$:

$$entityScore_{e,p} = \frac{C_{e,p}}{MaxC_{e,p}}$$

Let L be the set of all lemmas in subtree s and D be the set of all distinct synsets containing the lemmas in L:

$$lemmaSubtreeScore_{s,p} = \frac{\sum_{l \in L} lemmaScore_{l,p}}{|D| \cdot |L|}$$

The counts used above are calculated for each paragraph and all other paragraphs containing common entities or lemmas (or both):

Let E be the set of all entities occurring in $paragraph_p$ *and* $paragraph_q$:

$$entityCount_{p,q} = \frac{\sum_{e \in E} Min(entityScore_{e,p}, entityScore_{e,q})}{|E|}$$

Let M be the set of all lemmas occurring in $paragraph_p$ *and* $paragraph_q$:

$$lemmaCount_{p,q} = \frac{\sum_{l \in M} Min(lemmaScore_{l,p}, lemmaScore_{l,q})}{|M|}$$

Section 7.3: Visual Implementation

Section 7.3.1: Visualization with D3

The D3 visualization toolkit is used for implementing the word cloud as well as the tile browser. D3, known as "Data Driven Documents", allows direct manipulation of a shared representation of a web page known as the document object model (DOM) [BHO11]. In web-speak, the DOM is a platform and language independent interface allowing dynamic access and modification of the content, structure and style of elements on a web page [W305]. Comparable visualization toolkits such as Protovis [BH09] encapsulate the DOM, providing limited access and flexibility to the developer [BHO11]. Some of the features offered by D3 which are advantageous in creating dynamic visualizations include: the selection operator and immediate evaluation of visual attributes. Selection is used for filtering a set of elements within the DOM. In the DocuBurst application, it is particularly useful for selectively assigning colour to elements in the visualization and dynamically re-populating the word cloud's data when an entity or lemma node is clicked. The selection operator works nicely with a remove function by enabling selection of a data set, removal of selected elements from the screen and then repopulation of the visualization layout with new data elements. D3's also has an operator called "attr" which is used for immediately setting attributes of selected elements (i.e., the colour of a set of rectangles).

Section 7.3.2: D3 Word Cloud

The interactive word cloud layout implemented in D3 is used for displaying entities [Dav12]. To place the words, a randomized greedy algorithm is used which involves arbitrarily placing words within the bounding layout's space and selecting the larger words first for placement. After a word is placed, it is moved in a spiral direction outwards until it does not intersect with any of the other words currently lying in the layout. To fill the word cloud with data, entities are selected based on a higher number of co-occurrences with the synsets currently a part of the DocuBurst. Therefore, the entities within the word cloud will always relate to the currently displayed synsets.

Section 7.3.3: Multiple Visualization Coordination

An important requirement of visualization coordination is responsiveness. To achieve quick response and instant visual feedback triggered by mouse hovering, all required data is cached in a hash table data structure. This allows for relatively fast access to the hovered data element's co-occurring elements. The word cloud is limited to displaying twenty entities and the DocuBurst, with no concrete limitation, will display a controllable number of synsets. Therefore, the size of the hash tables will not require problematically large amounts of memory. One future improvement to optimize memory usage could be to cache all data in a single table, as opposed to separate tables for each co-occurrence relationship.

In response to mouse hovers, visual feedback is expressed through weighted brushing across all data elements in each of the three visualizations. Event listeners are implemented to detect mouse activity and the visual attributes of the data elements are altered accordingly (i.e., elements are re-coloured by changing their "fill" property). The amount of transparency in the highlighted colour is directly determined by the corresponding co-occurrence or occurrence score with the addition of a small offset value (0.15) to ensure lower scores do not result in transparencies that are almost invisible against the white background.

Section 8: Potential Uses

Section 8.1: Literary Analysis

The interactive DocuBurst application engages the user in new learning experiences through exposure of relevant extracted terms allowing inference of intra-document word relationships and patterns. Visualizing document content in this way can facilitate the task of "reading with numbers", an expression used to describe a quantitative approach to literary study [Eak04]. The digital humanities is an area of research involving the use of digital materials by merging traditional humanities research methods with analysis tools provided by the field of computing. Applicable to DocuBurst, such researchers are interested in formulating vague hypotheses and studying texts to trace and identify word-related patterns that can be used as evidence for proving, extending or refuting the original hypotheses. Tools to facilitate the innovative search tasks of these researchers generally consist of four operations: applying natural language processing to extract words characterized by parts of speech, computing useful word statistics, extracting named entities, and then present this data effectively with a visualization [HM11]. This demonstrates that there may be place for DocuBurst in the field of digital humanities research. DocuBurst enables search through "neighbourhoods" of related words to explore selected words of interest and potentially uncover additional terms that may be meaningful to the analyst [HM11].

Section 8.2: Widespread Web Application

In general, a rewarding goal for visual designers is creating work that is both admirable and influential to the public eye. Inspired by Many Eyes' active membership of the web ecosystem [KMVVW07] and Wordle's continually growing popularity, enhanced collaboration through goals for

social integration and public online deployment will allow DocuBurst to emerge on the web as a multi-purpose document content visualizer and analysis tool.

Section 9: Future Work

Section 9.1: Proposed User Study

Public deployment of DocuBurst on the web presents an opportunity for user evaluation and conducting usability related studies on the interface. A technique used for Wordle called the "Wordle in the Wild" describes a methodology where the visualization is studied in its "natural habitat" by searching for references to the visualization and uncovering its use cases [FVW09]. Conducting a similar study on DocuBurst would allow investigation of where DocuBurst is currently residing in the web, revealing its uses and the different text genres users are interested in visualizing. Another area for investigation is classifying the main types of users. For example, non-expert users using DocuBurst for personal use or readers making use of DocuBurst's analytical tools. Using this information can help in future design iterations of the interface involving usability related improvements or additional helpful features. As demonstrated with Wordle, this type of study can reveal interesting, unexpected use cases [FVW09].

Additionally, all user activity is currently logged by the application. Logging data is an inexpensive, unobtrusive technique in human-computer interaction for tracing the actions and events of multiple users on an interface [BGGHS94]. Collecting this empirical data is particularly beneficial in evaluating usability of an interface and effectiveness of design choices [BW01]. Tracking usage can be an asset for uncovering which tools the users are frequently using and how they are exploring the visualizations. For example, using DocuBurst in an educational setting for book summaries or using it to display social network textual data, such as twitter feeds. It is also both valuable and desirable to know the types of text users are interested in visualizing. Whether

they are uploading personal texts or interested in exploring existing texts. The purpose of knowing this information is to reveal apparent trends and decide whether or not the interface can be improved to encourage or evolve these trends.

Specific areas to concentrate on when carrying out a user evaluation which may be helpful in enhancing the current interface include the user's response to aesthetic quality, ease of use and usefulness of the interactions provided and first glance interpretations of the display. To obtain an in-depth focus on specific users, semi-structured interviews can be conducted. To obtain more quantitative results, questionnaires can be distributed to a large sample of users. Observing how users make use of the website could give some perspective on how the current design could be altered or identify any useful additions that the interface does not currently provide. Another area of interest is in the visual presentation. It can be valuable to know the user's preference and the reason for this preference of single node view versus cumulative or single text versus a comparative view of multiple texts. The effectiveness and usefulness of presenting extracted entities is also important. An outline created for questionnaires and interviews are presented in Appendices B and C.

Section 9.2: Entities and Suggested Roots

Section 9.2.1: The Importance of Suggested Roots

In DocuBurst, a suggested root indicates a potential synset for rooting the sunburst visualization at, which will produce an informative and relevant representation of the document. More abstractly, suggested roots are comparable to the auto-complete function of a search engine. It is likely for a new user to be unsure of which root to begin with. Intuitively, they consult the suggested roots to help them decide. This action being similar to half typing a search query into Google and using the suggestions as a way to refine, complete or reformulate an initial query. Suggested roots are intended to guide the user towards helpful starting points of their exploration.

Therefore, they have the ability to influence the path of a user's information seeking journey. Supporting exploration based on interesting examples can be a way to keep the user engaged with the system [HM11].

Section 9.2.2: Current Techniques for Selecting Suggested Roots

Suggested roots should contain a high number of children and a decent occurrence score producing a detailed DocuBurst visualization which exposes as many relevant terms as possible. Suggested roots should strive to be representative of the major themes expressed in the document. The DocuBurst application is currently suggesting roots based on selecting synsets with a large number of children and an overall high occurrence score within the document. The number of children with non-zero scores are also considered because a synset with a large number of children may have very few children existing within the document. There has been previous work in experimenting with different calculations to decide the optimal method of suggesting roots [Chi10]. This experimentation mainly involved adjusting the weights of each of the three terms in the formula for the *root score* presented below. There is no perfect solution to finding ideal root suggestions. However, through trial and error, some methods are proven better than others. For each synset found in the document, a *root score* is computed which is a measurement of how qualified the synset is for being a suggested root. Higher scores indicate a better root candidate and the top scores are selected as the suggested roots for the document. For each synset found in the document, a *root score* is pre-computed in the following manner:

If the candidate root has at least one child and at least one child has a non-zero score:

$$root\ score = A \cdot \sqrt{\frac{score}{children}} + B \cdot nonzero + C \cdot (children - nonzero)$$

Where *A, B, C* and *D* are parameters used as weight factors to determine how heavily each term contributes to the *root score, score* is the occurrence score of the synset indicating its strength of occurrence in the document, *children* is the total number of children of the synset and *nonzero* is the number of children with a non-zero occurrence score.

This calculation considers three factors for determining whether or not a synset is well suited for a suggested root: number of children found in document (i.e., having a non-zero score), the number of children not found in the document and frequency of the synset occurring in the document averaged over all of its children.

Section 9.2.3: Incorporating Entities into Root Scores

Introducing entity extraction alludes to an interesting research question of whether or not entities can make a positive contribution to determining useful suggested roots assisting the user in beginning their exploration. More specifically, examining the effect incorporating entities into the initial *root score* computation has on the quality of suggested roots. High-quality suggested roots should produce DocuBurst visualizations clearly representing the main topics of the document and display sufficient levels of detail (i.e., around three levels of the noun hierarchy present). Entities and synsets are essentially two disjoint pieces of information, where a connection is established only through examining co-occurrences. Since all co-occurrence scoring is already known, these scores can be incorporated into a new root score calculation:

Initial suggested roots based on high co-occurrence with entities:

$$modifiedRootScore = rootScore + D \cdot entityScore$$

Where *D* is a parameter to determine how heavily each term contributes to the *modifiedRoot Score* and *entityScore* is an average of all co-occurrence scores between the candidate root and entities.

The equation above will assign higher *root scores* to synsets co-occurring with more entities extracted from the document.

The next approach involves suggesting roots based on entities of interest while the user is viewing a specific root. Using the entities displayed with respect to the current root, roots can be suggested based their co-occurrences with the entities presented in the word cloud. A useful feature could be to suggest roots dynamically based on an entity of interest. When an entity is selected, it's reasonable to assume the user is interested in exploring other data elements closely related to the entity. Therefore, suggesting roots based on a selected entity or entity set may help with further investigation of that particular entity. For instance, if an entity is clicked and the word cloud is filtered accordingly, roots can be suggested based on their co-occurrence with the selected entity and its co-occurring entities. Each time an entity is selected, different entity-based suggested roots could be displayed. Therefore, the user is always presented with new suggested roots each time they explore different entities. The formula below computes a *root score* with respect to a sub-set of entities presented in the word cloud:

$$entity\ root\ score = rootScore + D \cdot cooccurScore$$

Where D is a parameter used to determine how heavily each term contributes to the *entityRoot Score* and *cooccurScore* is an average of all co-occurrence scores between the candidate root and entities present in the word cloud.

The above calculation can also be applied for when an entity is selected and the word cloud is filtered. In this case, the *cooccurScore* would only consider co-occurrences with entities in the filtered set.

Section 9.2.4: Experimentation

The next step would be to test these calculations and decide whether or not they are improving the quality of computed suggested roots in terms of producing DocuBurst visualizations which are well representative of the document or correspond to the entity of interest. An ideal weight factor (parameters *A, B, C* and *D*) assigned to each added term in the *modified root score* formula will also need to be determined empirically through trial and error to uncover how much entity co-occurrence should contribute to *root score* calculations. Suggesting roots based on entities may depend on how heavily users rely on entities for exploration. Either entities are a contributing factor for when the user selects new roots or users prefer to select new roots without considering related entities. Determining these intentions of users may require a user study or observation through public deployment of the application.

Section 9.3: Integration with Online Document Repositories

Depending on the interests of the users, it may be beneficial to integrate DocuBurst directly with existing online document repositories. Organizations such as Project Gutenberg offer free access to copies of digital books. Other newspaper companies provide retrievable online archives of current and past news articles, such as the New York Times or the Toronto Star. Integrating DocuBurst with such websites allows a user to make use of the application without initially providing a document of interest, allowing the user to casually browse and explore thought-provoking visualizations.

Section 9.4: Enhanced Online Collaboration

The key to creating a widespread application with high popularity on the internet is through integration with social networking sites. Websites such as Facebook and Twitter offer simple embedding features to link an application directly with their website. Placing DocuBurst in the social networking environment introduces a larger audience of potential collaborators willing to

explore and share texts with DocuBurst. It is important for DocuBurst to have collaborative capabilities in order for it to become an active member of the web.

Section 9.5 Entity Extraction with Document Comparison View

A powerful feature offered by the DocuBurst application is the ability to create comparative views of two documents. In this view, the two documents rooted at the same synset are visualized in a single DocuBurst where differences in colour represent common synsets between the two documents and conversely, synsets existing only in one of the documents. Relative frequencies are encoded by colour transparency, similar to a single-document DocuBurst. Integrating entities with this comparative visualization should maintain the consistency and simplicity of the existing interface, easing the transition between different views of the application. Therefore, introducing different entity visualizations or applying a different layout is unsuitable.

A naïve approach could be to extract the co-occurring entities of both documents and then display them in a word cloud. However, with highly unrelated documents, there is a chance of very little or no entities at all matching this criteria. To resolve this, another possibility is to extract the top occurring entities in each document, displaying them in separate word clouds. When a user hovers over an entity in either word cloud, if the same entity occurs in the other document it will also be highlighted in the corresponding word cloud. Keeping a similar colour scheme, different colours portray which document the entity exists in. Unfortunately, this leads back to the original problem of having too many different colours present in a single view. To decide on effective visual design choices, some experimentation and possibly verification with real use would have to be performed. Presented here is simply a starting point for the new design of the comparison view.

Using entity extraction with comparison view in DocuBurst allows for tracking entity usage in multiple texts, enhancing the application's cross-document analytical capabilities. Consider the

comparative view of biographies of two sociologists who lived during overlapping time periods (Figure 9 and 10), extracted entities could reveal how the author's lives were related or influenced by similar people or locations. Using the original colour scheme, blue and green is used to differentiate between the two documents and red indicates the existence of a term in both documents. The varying transparency of colour represents the strength of occurrence (Figure 9) or co-occurrence (Figure 10) of each coloured word.

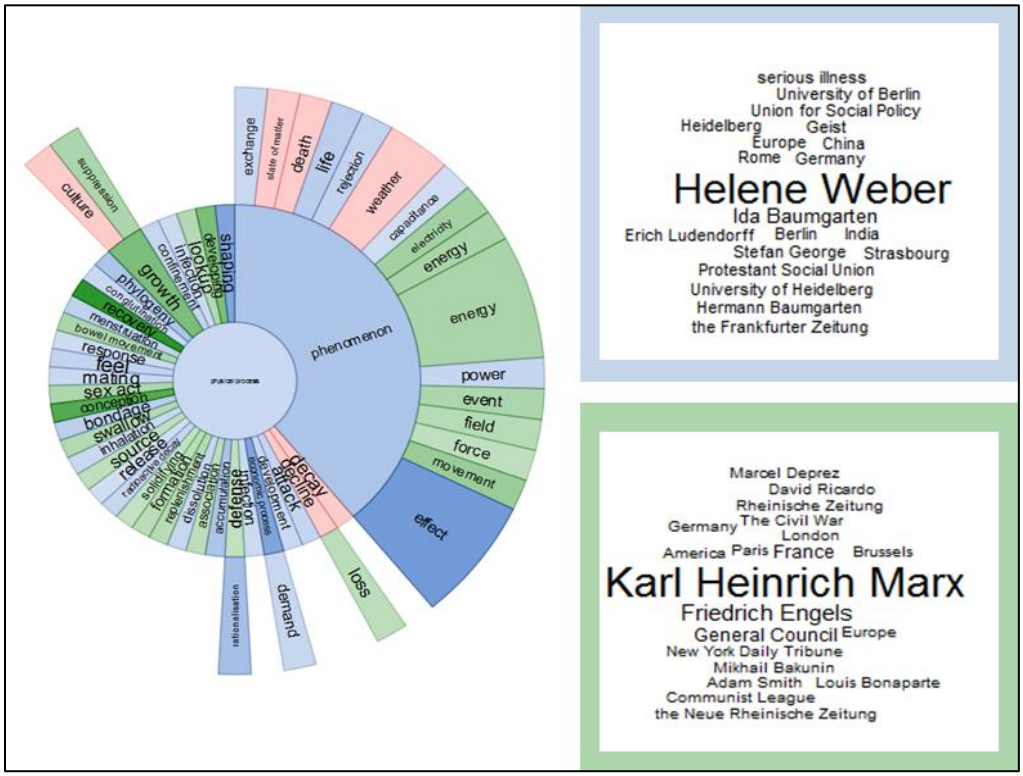

Figure 9: A sketch of the comparative view of two biographies about Karl Marx and Max Weber.

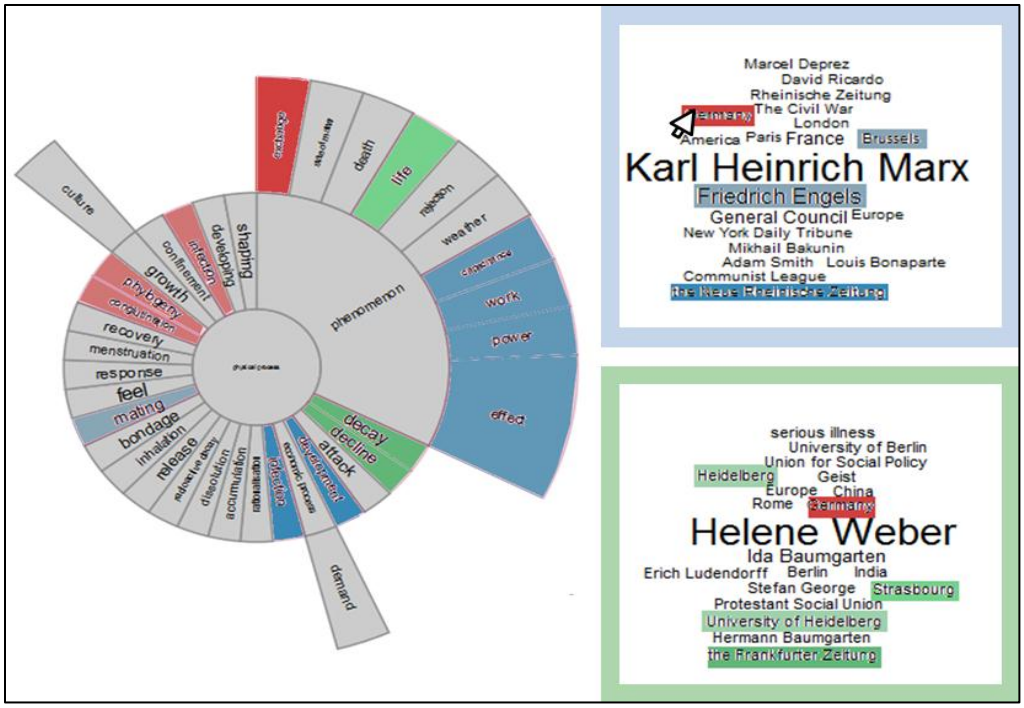

Figure 10: A sketch of the comparative view visualizing two biographies of Karl Marx (blue) and Max Weber (green). The mouse is hovered on "Germany". As a result, all co-occurring entities with "Germany" are highlighted in both word cloud. Since "Germany" occurs in both biographies, it is distinguished by the colour red.

Section 10: Conclusion

The intent of DocuBurst is not only to visualize the language components contained in documents, but to offer a learning experience to each user by providing rich interactivity and aiding content analysis with relevant key term extraction. With the help of DocuBurst as a visual analytical tool, a user is able to observe noun hierarchies, explore related named entities, view occurrences of any selected term by reading the document in text segments and temporarily reveal relatedness across all data elements presented on the screen. These features are used for guiding the user towards exploring sections of the document of interest by relating general nouns to more comprehensive, co-occurring entities. Adding automated entity extraction improves DocuBurst's analytic capabilities and the new interface design enhances unity among visual elements. The named entity extraction expands the document type space to contain media-related, biographical or

geographical themed texts. It also overcomes the original limitation of noun-based visualization, where documents are under-represented by neglecting proper nouns which provide the required details to further comprehend specific instances of nouns. Furthermore, potential future work and additional design considerations have been premeditated.

References

[BGGHS94] Albert Badre, Mark Gray, Mark Guzdial, Scott Hudson and Paulo Santos. *Analyzing and Visualizing Log Files: A Computational Science of Usability*. Presented at HCI Consortium Workshop. 1994.

[BH09] Michael Bobstock and Jeffrey Heer. *Protovis: A Graphical Toolkit for Visualization*. IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis). 2009.

[BHO11] Michael Bobstock, Jeffrey Heer and Vadim Ogievetsky. *D3: Data-Driven Documents*. IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis). 2011.

[BW01] Mary C. Burton and Joesph B. Walther. *The Value of Web Log Data in Use-Based Design and Testing*. Journal of Computer Mediated Communication, Volume 6, Issue 3, pages 41-93. April 2001.

[CCDW08] Sheelagh Carpendale, Christopher Collins, Marian Dork and Carey Willamson . *VisGets: Coordinated Visualizations for Web-Based Information Exploration and Discovery*. IEEE Transactions on Visualization and Computer Graphics In Visualization and Computer Graphics, IEEE Transactions on, Vol. 14, No. 6. November 2008.

[CCP09] Sheelagh Carpendale, Christopher Collins and Gerald Penn. DocuBurst: Visualizing Document Content Using Language Structure. Eurovis '09. 2009.

[Chi10] Bradley Chicoine. *Chicoine DocuBurst*. http://cslab.pbworks.com/w/page/25871701/Chicoine%20DocuBurst. PBWorks. 2010.

[CKPV09] Tanya Clement , Amit Kumar , Catherine Plaisant and Romain Vuillemot . *What's Being Said Near "Martha"? Exploring Name Entities in Literary Text Collections*. IEEE Symposium on Visual Analytics Science and Technology, VAST 2009. 2009.

[Cla07] Jeff Clark. A Text Visualization Tool. http://www.neoformix.com/2007/ATextExplorer.html. March 2007.

[Col10] Christopher Collins. *Interactive Visualizations of Natural Language*. University of Toronto. 2010.

[CW07] Jeff Clark and Mark Wattenberg. Document Arc Diagrams. http://www.neoformix.com/Projects/DocumentArcDiagrams/index.html. April 2007.

[Dav12] Jason Davies. *Word Cloud in D3*. www.jasondavies.com/wordcloud. February 2012.

[Dor08] Marian Dork. Live Demo of VisGets. http://pages.cpsc.ucalgary.ca/~mdoerk/view/explore. 2008.

[Eak04] Emily Eakin. *Studying Literature By the Numbers*. The New York Times. January 10th, 2004.

[FVW09] Jonathan Feinberg, Fernanda B. Viégas and Martin Wattenberg. *Participatory Visualization With Wordle*. IEEE Transactions on Visualization and Computer Graphics Volume 15 Issue 6. November 2009.

[HM11] Marti A. Hearst and Aditi Muralidharan. WordSeer: Exploring Language Use in Literary Text. HCIR. 2011.

[KMVVW07] Jesse Kriss, Matt Mckeon , Frank Van Ham, Fernanda B Viegas and Martin Wattenberg . *Many Eyes: A Site for Visualization at Internet Scale*. IEEE Transactions on Visualization and Computer Graphics In Visualization and Computer Graphics, IEEE Transactions on, Vol. 13, No. 6. November 2007.

[MN97] John Morkes and Jakob Nielsen. *Concise, Scannable, and Objective: How to Write for the Web.* http://www.useit.com/papers/webwriting/writing.html. 1997.

[Nie05] Jakob Nielsen. *Ten Usability Heuristics*. http://www.useit.com/papers/heuristic/heuristic_list.html. 2005.

[PS86]Catherine Plaisant and Ben Schneiderman. *Designing the User Interface: Strategies for Effective Human-Computer Interaction, 5th Edition*. Addision-Wesley Longman Publishing Co., Inc. Boston, MA, USA. 1986.

[PU10] Princeton University. *About WordNet.* http://wordnet.princeton.edu. 2010.

[Reu11] Thompson Reuters. *Open Calais Documentation*. http://www.opencalais.com/documentation. 2011.

[Shn97] Ben Shneiderman. Direct Manipulation for Comprehensible, Predictable and Controllable User Interfaces. IUI '97 Proceedings of the 2nd international conference on Intelligent user interfaces. 1997.

[Shn96] Ben Shneiderman. The Eyes Have it: A Task by Data Type Taxonomy for Information Visualizations. VL '96 Proceedings of the 1996 IEEE Symposium on Visual Languages. 1996.

[W305] W3C. *Document Object Model (DOM)*. http://www.w3.org/DOM/. 2005.

Appendices

Appendix A: Entity Categories Extracted from Open Calais

Descriptions of the Categories listed below can be found in the Open Calais documentation at: http://opencalais.com

Category Names
City
Continent
Country
Province or State
Region
Natural Feature
Person
Organization
Facility
Company
Entertainment Award Event
Political Event
TV Show
TV Station
Radio Program
Radio Station
Movie
Music Album
Music Group
Published Medium
Sports Event
Sports Game
Sports League
Technology
Operating System
Programming Language
Medical Condition
Medical Treatment
Industry Term

Appendix B: Interview Outline

Demographical/Background Questions:

- What is your name?
- How did you find or hear about DocuBurst?
- What was the technology used for accessing the DocuBurst web-application (i.e., laptop, desktop, Ipad)?
- What are some of your main intentions or goals while using DocuBurst?

- Were you using DocuBurst for any specific purpose other than recreationally?

Visualization Interpretation Questions:

- After exploring the visualization, are there any interesting observations or insights gained about the document you explored?
- What do you think the DocuBurst diagram is conveying in terms of its relation to your document?
- What do you think the entity word cloud is conveying in terms of its relation to your document?
- Which visual attribute do you find to be the most pre-attentive in the entire interface and why?
- Which data characteristic do you think the pink coloured highlights intended to portray?
- Describe your interpretation of the varying colour transparencies used across the interface.
- Describe your interpretation of the varying sizes of each pie segment in the Docuburst diagram.
- Did you try the comparative view? If so, comment on its effectiveness of comparing two documents.
- Did interacting with the connected visualizations assist you in exploring the document? If so, how?

Usability Questions:

- Which tools did you find the most helpful or useful in completing desired tasks? Why?
- Were there any interaction features or tools you found difficult to use? If so, describe the steps taken in resolving these difficulties.
- Did you encounter any errors or unexpected behaviour when using the web application? If so, please explain.
- Did the sequence of interactions provided appear to flow smoothly? Please explain.
- Do the mouse interactions feel intuitive and natural to you? Please explain.

Appendix C: Questionnaire Outline

General Usability Questions:

	Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree	Not Applicable
I find this website visually appealing.						
It is easy to move back and forth between web pages.						
There is sufficient help documentation provided.						
I did not feel frustrated while using this website.						
The terminology used on this website (dis-regarding the content on the visualizations) is clear and straightforward.						

I did not encounter any errors while using this website.						
I would be likely to use this website in the future.						

Specific Design Questions:

	Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree	Not Applicable
Looking at the DocuBurst helped me learn more about the nouns within the document.						
Incorporating named entities into the view as a separate visualization was helpful.						
I prefer the single-node view over the cumulative view.						
I made use of the comparative view to compare two documents.						
Interacting with all three visualizations felt natural and easy.						

The type(s) of documents that I am most interested in visualizing is (are) (Check all that apply):

- Novels and Poetry
- Personal Information
- Academic Papers, Reports or Textbooks
- Scripts and Speeches
- Online Social Media
- Newspaper Articles
- Corporate/Business Reports

Additional Comments

Please provide any feedback or suggestions for improvement below: