WordBridge: Using Composite Tag Clouds in Node-Link Diagrams for Visualizing Content and Relations in Text Corpora

KyungTae Kim
kimk@purdue.edu

Sungahn Ko
ko@purdue.edu

Niklas Elmqvist
elm@purdue.edu

David S. Ebert
ebertd@purdue.edu

Purdue University
West Lafayette, IN, USA

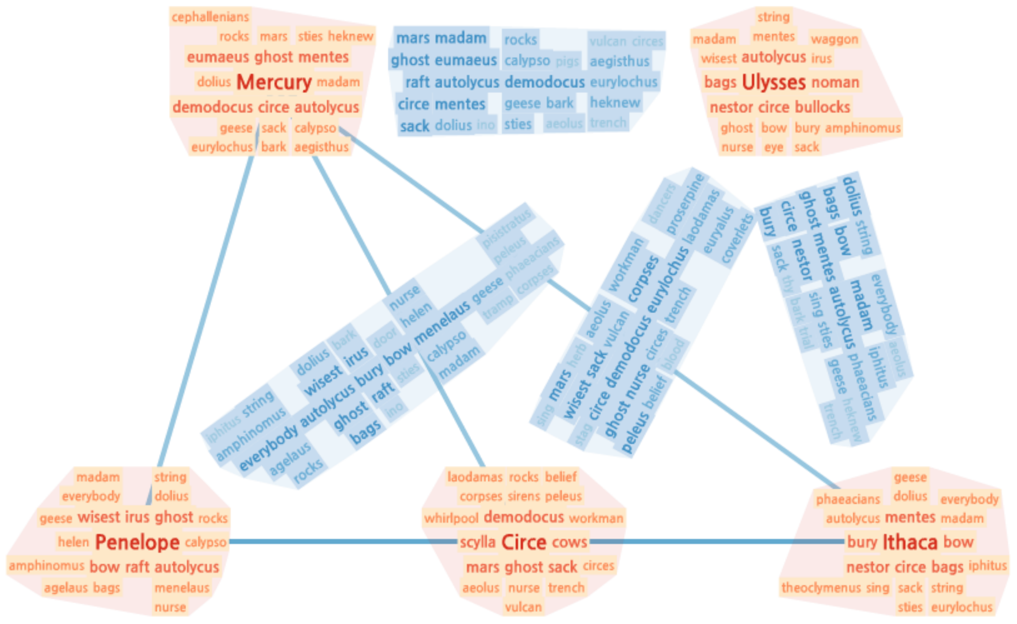


Figure 1: Important entities in Homer's *Odyssey*. Entities were identified using OpenCalais [20]. Links represent co-occurrences in a book (part) of the story. Orange *node clouds* represent terms that characterize entities and blue *link clouds* characterize their pairwise relationships.

Abstract

We introduce WordBridge, a novel graph-based visualization technique for showing relationships between entities in text corpora. The technique is a node-link visualization where both nodes and links are tag clouds. Using these tag clouds, WordBridge can reveal relationships by representing not only entities and their connections, but also the nature of their relationship using representative keywords for nodes and edges. In this paper, we apply the technique to an interactive web-based visual analytics environment—Apropos—where a user can explore a text corpus using WordBridge. We validate the technique using several case studies based on document collections such as intelligence reports, co-authorship networks, and works of fiction.

Index Terms: H.5.2 [Information Interfaces and Presentation]: User Interfaces—Graphical User Interfaces (GUI)

1 Introduction

The sheer size of digital text data in many domains and media today makes it difficult for people to quickly grasp what data is important, and what data is not. Exhaustively reading the full text corpus relating to a particular area—such as news articles, financial documents, or intelligence reports—is often not feasible given time constraints imposed on an analyst. To give a concrete example, news broadcasting services such as CNN, Fox News, and MSNBC produce an average of 50,000 unique words per month—barring trivial stopwords of the English language and not counting the full texts [11].

In other words, there is clearly a need for summarizing and abstracting large text corpora into a representation suitable for overview. Using this representation, an analyst could quickly decide which documents (or reports) are important or not, and retrieve and read the full text only for those documents. Of course, the key question is "How can we design a representation that both aptly summarizes the underlying content, yet provides a much more compact view than the full text of the documents themselves?"

While there have been many attempts at answering this question in fields such as information retrieval [12, 23], text mining [14], and knowledge discovery [3], we focus on visualization approaches here. Examples of text visualization methods include tag clouds [2, 28], Wordle [29], WordTree [32], and phrase nets [26]. However, powerful as these all are, few explicitly focus on showing *both* the content as well as the relations within the text corpus.

In this paper we introduce a novel relationship visualization technique called **WordBridge** that uses a composite tag cloud representation shaped as a node-link diagram to display both the context and the relationship between entities (Figure 1). The metaphor is that of a "bridge" of words that connect one entity to another. WordBridge builds on existing text visualization approaches by providing a hybrid between a simple term co-occurrence graph and a tag cloud of the text of a document. It uses a dynamic and deterministic layout algorithm inspired by the Wordle [29] tag cloud. In this fashion, we can strike a balance between overview and detail, while providing summaries of documents without the need to read them all.

To showcase the utility of the WordBridge technique, we have implemented a web-based visual analytics system for investigative analysis that we call Apropos. Apropos is designed to show relationships between entities, documents, and important keywords in a text corpus using the WordBridge representation. In an offline stage, Apropos first identifies entities and extracts keywords using freely available text retrieval tools. It then provides an interactive web-based visualization environment where a user can explore the corpus using the WordBridge technique. We validate our new technique and the Apropos system through a set of examples for investigative analysis and relationship data derived from fiction.

2 Related Work

There exists a multitude of text visualization techniques in the literature. Collins et al. [5] present and classify a number of the most popular techniques in terms of the feature sets they provide. In this paper, we take a slightly different view, opting to primarily focus on **textual relations** within a text corpus.

In the following treatment, we will first give a brief background on some of the most relevant text visualizations for general use. We will then focus on techniques that support comparison and seeing linkages between documents. We will begin, however, by studying the state of the art in text mining, retrieval, and extraction.

2.1 Text Mining, Retrieval and Extraction

Text mining is the process of deriving high-quality information from text. The literature on text retrieval and extraction is large and colorful [14], and includes as diverse areas as text categorization, text clustering, entity identification, sentiment analysis, document summarization, and entity-relation modeling.

In this work, we are particularly interested in extracting representative keywords from documents, i.e., a form of *document summarization*. There exists a large amount of work in this domain; see [1, 17, 19] for surveys and overview. Some of the representative techniques include relevance weighting schemes [21], idf [16], and tf-idf and its family of techniques [22, 23].

Among these techniques, the latter, "term frequency-inverse document frequency" (tf-idf), is one of the most popular term weighting methods in the field of information retrieval [1]. It is also relatively easy and efficient to implement, which is why we focus on it here. In the tf-idf scheme, once a "term" is chosen in a document, a count for its occurrences is computed for each document in the corpus. After proper normalization, this term frequency count is compared to an inverse document frequency count, which means the number of occurrences of the term in the entire corpus.

There are two empirical intuitions in using tf-idf: If a term t has a high frequency in a document d (called the *term frequency*, or tf), it is highly possible that t is important in d. On the other hand, if t has high frequency over all documents (called the *inverse document frequency*, or idf), the term probably is not the powerful to differentiate importance between documents. For a given document, these tf and idf are combined to derive weights by dividing the tf by the idf for each term. Generally, a log-scale is taken to reduce the impact of larger tf-idf values.

For example, under the tf-idf scheme, a common word such as "the" is not important in a document collection; while it may have a high frequency in a specific document, it will also have a high frequency for the whole document collection. Therefore, the tf-idf value for the word will be low. On the other hand, for a word that is common in a single document, but uncommon in the whole corpus, the intuition is that the word is indeed important to that document.

2.2 Visualizing Text

Text visualization has lately become a mainstream form of visualization for the masses through the *tag cloud* visualization technique [28]. Tag clouds (also known as *word clouds*) are popular on the Web for all kinds of social media sites, and visualize text by laying out words from the text corpus in a visual space (often in alphabetical order) and encoding data about the relative frequency, popularity, and preference of each term using its graphical attributes ike color, size, and weight. While studies have shown that font size and weight are particularly effective for making words in a tag cloud "pop" out on the canvas [2], it is a fact that standard tag clouds do not make use of the spatial dimension at all.

This is also the motivation for the Wordle [29] tag cloud technique. Wordles are essentially beautified tag clouds that make more efficient use of display space by packing terms together, even inside other terms. However, the standard Wordle layout algorithm tends to be slow and is not designed for interactive visualization.

Beyond tag clouds, there exist a number of additional forms of text visualization. ThemeRiver [13] visualizes temporally changing themes extracted from document collections. Document Cards [25] are visual document summaries built by extracting text and images from PDF documents, but focuses on individual documents as opposed to showing relations between them. IN-SPIRE [33] spatializes text into theme views that preserve important characteristics from documents, and also begins to show relations in the text.

2.3 Visualizing Textual Relations

Visualizing the structure within a text corpus is becoming increasingly important. Some techniques focus on structure rather than textual content. Arc Diagrams [31] show repetition in string data; one of its applications is for text documents. FeatureLens [8] supports visual exploration of frequent text patterns in document collections, but also allows drilling down to marked-up text.

Including actual words in the visual representation helps users understand the text corpus. DocuBurst [5] uses the existing WordNet ontology to group similar words into a space-filling radial hierarchy. The WordTree [32] visualizes relations within a document on a per-word level, constructing an interactive hierarchy of the context of a particular starting word. Users can explore the hierarchy, causing the layout to change dynamically. Phrase nets [26] take this a step further by constructing a graph of related words in text documents (i.e., again on a per-word level) based on a user-specified relation. The result is a kind of structured tag cloud that allows for extended refinement of the word relation.

In that same vein, Parallel Tag Clouds [6] (PTCs) are exactly that—a tag cloud supporting faceted browsing of text corpora. PTCs are interesting for our purposes because they allow comparison between documents and parts of documents. PTCs are inspired by Themail [27], a system for extracting characterizing keywords in e-mail conversations to show temporal relationships for different recipients. Most recently, the POSvis [30] system supports literary analysis through a combination of filtering and visual representations to study vocabulary and relations within documents.

Perhaps the technique that is most related to WordBridge is GreenArrow [34], which uses a graph representation with labels as links between nodes. While Wong et al. discuss the concept of a dynamic node label using the same concept as their link labels, they do not explore this approach further, and do not incorporate both techniques into the same visualization.

3 The WordBridge

The WordBridge is a novel graph visualization technique for showing relationships between entities. Instead of merely showing the relationships as visual links in a node-link diagram, WordBridge is able to reveal the *nature* of the relationship using a text representation. This is possible because both nodes and edges in WordBridge are represented by constrained-layout Wordle [29] tag clouds that can express selected keywords that characterize the entities (for nodes) as well as their relationships (for edges). The metaphor is that of a "bridge of words" connecting one entity to another.

3.1 Data Model

The WordBridge visualizes graph data where both nodes and edges are associated with a set of ranked keywords. More specifically, for a standard graph structure $G = (V, E)$ where V and E are simple sets of vertices and edges, respectively, the WordBridge vertex and edge sets also have, for each vertex $v \in V$ and edge $e \in E$, an associated set of word-rank tuples $T \subset \mathbb{W} \times \mathbb{R}$, where $\mathbb{W}$ is the set of all permissible alphanumeric words. In the tuple (w, r), accordingly, the rank r for each word w is in the interval $[0, 1]$, and communicates how well the word represents the node or edge it is associated with.

For the concept of the WordBridge bridging entities in the dataset to hold true, we impose the following constraints on keyword sets:

- For a vertex v, the keyword set $T_v(v)$ should characterize the vertex **itself**.
- For an (undirected) edge e between two vertices v_1 and v_2, the keyword set $T_e(v_1, v_2)$ should characterize the (bidirectional) **relationship** between the two vertices.

Despite these constraints, there is some latitude in the WordBridge model in defining what characteristic keywords entail, i.e., the content of the T_v and T_e sets for nodes and edges. In the following section, we describe some useful interpretations.

3.2 Example Datasets

We originally designed the technique to use a graph dataset derived from entity-relationships extracted from text corpora. Such datasets could be analyzed through entity identification techniques [4, 14, 15]. In such circumstances, the nodes would be the entities themselves, and the edges would be the co-occurrence of these entities in documents. We then use text extraction algorithms [14] to retrieve keyword sets T_v and T_e for all vertices and edges, respectively.

However, there are other ways to form WordBridge graphs. Similar data models can be used for graph data such as co-authorship in a publication database, collaboration for a network of organizations, or even relationships in a social network. In such instances, however, the set of keywords describing as well as bridging entities must be carefully defined. Here are some examples:

- For a **co-authorship network**, authors in the publication database are vertices (V), and edges (E) connect authors who have written an article together. The full text of all articles that an author A has written could be used to extract representative keywords for the author ($T_v(A)$), and the specific articles that two authors A and B co-authored would be used for characterizing the relationship ($T_e(A, B)$). Figure 2 gives an example.
- For a **collaboration network**, organizations are represented by vertices (V), and edges (E) are their collaborations. Documents describing each organization, such as a webpage, business plan, or mission statement, can be used to derive the keywords for the organization (T_v), and any documents describing the collaboration can be used for the relationship (T_e).
- For a **social network**, such as from a social networking site such as Facebook, the actors are vertices (V) in the graph, and their acquaintances are represented by edges (E). For the Facebook example, representative keywords for an actor (T_v) could be derived from the personal wall posts of each actor, and the keywords for the relation (T_e) could be extracted from the union of wall posts exchanged between the two actors

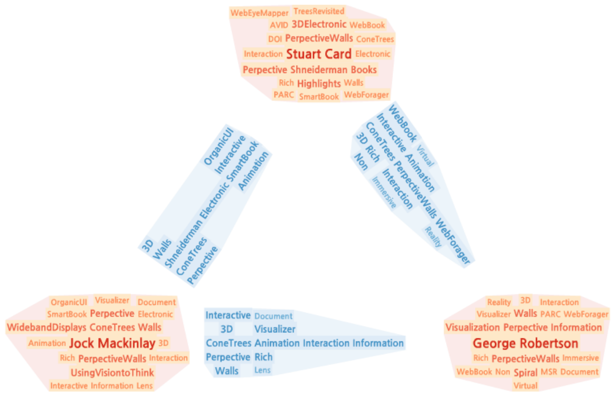

WebEyeMapper TreesRevisited
AVID 3DElectronic WebBook
DOI PerpectiveWalls ConeTrees
Interaction Stuart Card Electronic
Perpective Shneiderman Books
Rich Highlights Walls
PARC SmartBook WebForager
OrganicUI Interactive SmartBook Animation
3D Walls Shneiderman Electronic ConeTrees Perpective
WebBook Virtual
Interactive Animation
ConeTrees PerpectiveWalls WebForager
3D Rich Interaction Reality
Non Immersive
OrganicUI Visualizer Document
SmartBook Perpective Electronic
WidebandDisplays ConeTrees Walls
Animation Jock Mackinlay 3D
Rich PerpectiveWalls Interaction
UsingVisiontoThink
Interactive Information Lens
Interactive Document
3D Visualizer
ConeTrees Animation Interaction Information
Perpective Rich
Walls Lens
Reality 3D Interaction
Visualizer Walls PARC WebForager
Visualization Perpective Information
George Robertson
Rich PerpectiveWalls Immersive
WebBook Non Spiral MSR Document
Virtual

Figure 2: WordBridge for three prolific InfoVis authors. Tags in each *node cloud* (orange) represent that individual's research, and tags in each *link cloud* (blue) represent joint projects.

3.3 Visual Representation

The basic visual representation of WordBridge is that of a node-link diagram of the graph $G = (V, E)$. Instead of standard visual nodes, a WordBridge node for the vertex $v \in V$ is a Wordle [29] tag cloud of the keyword set $T_v(v)$ centered around the node's position—a *node cloud*. Analogously, a visual link between two vertices v_1 and v_2 is no longer a line connecting two nodes, but rather a Wordle [29] tag cloud of the keyword set $T_e(v_1, v_2)$ centered around this connecting line—a *link cloud*. Unlike extended graph labels [34], which have the option of being directed, link clouds are always bi/undirected.

Both node and link word clouds use dynamic layouts to adapt to the available display space. This is particularly important for link clouds whose length may change as the graph layout changes. To clearly delineate the boundaries of both node and link clouds, they are surrounded by dynamically-computed convex hulls that adapt to their current size and layout.

Constructing a WordBridge relies on both high-level graph layout, as well as layout of individual word clouds. Below we discuss these two issues, as well as the use of color, scaling, and convex hull borders that all constitute the WordBridge visual representation.

3.3.1 Graph Layout

The WordBridge can use any graph layout algorithm, but because the links in the technique are actually word clouds themselves, layouts that minimize edge length will minimizing visual clutter. In our prototype, we have mainly used force-directed layouts [9, 18].

3.3.2 Cloud Layout Requirements

Our node and link clouds have several requirements that differentiate them from standard word clouds:

R1. Constrained layout: As opposed to standard word clouds, our node and link clouds will either be organized around a point (node clouds), or along a line (link clouds).

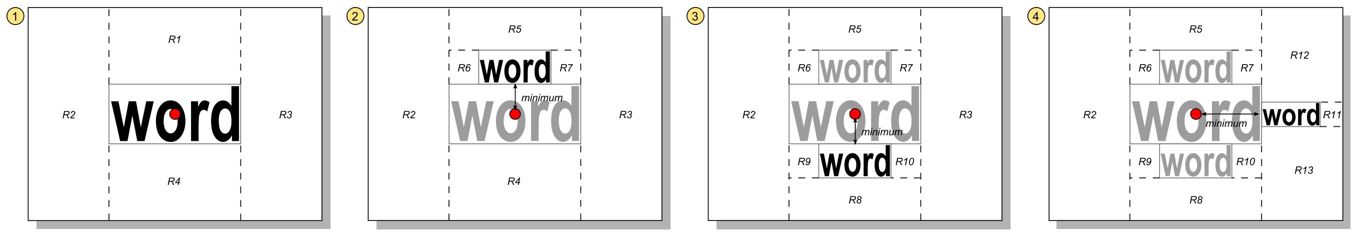


Figure 3: Cloud layout algorithm. Terms are added to the canvas iteratively, starting with the largest (or widest) term first. The position to place the term is determined by its distance to the optimal center point (red circle) for a node cloud, or, for a link cloud, the optimal center line.

R2. Space-efficient: Because our clouds are now going to be part of large, composite word clouds in the WordBridge visual representation, we must use a space-efficient layout that minimizes wasted space.

R3. Computationally efficient: For the same reason as above, our cloud layout algorithm must be sufficiently efficient to lay out a large number of individual clouds while retaining real-time rendering performance.

R4. Variable shape and size: Node and, in particular, link clouds must be able to adapt to varying cloud shape and size configurations as the graph layout changes, possibly dynamically.

R5. Deterministic: As a corollary of the above, our cloud layout must be deterministic so that two invocations of the algorithm for the same space configuration yield the same layout. Furthermore, it is desired that adding or removing terms to the end of the frequency list (i.e., as if increasing or decreasing the detail of the word cloud) yields a stable layout.

Standard tag clouds [2, 28] are computationally efficient (R3) and are deterministic (R5), but that is also the end of their virtues. They are not space-efficient and they do not allow for constrained layout or variable shape (only variable size).

The Wordle [29] layout is a much better fit. It allows for constrained (R1) and variable (R4) layout, and they are most definitely space-efficient (R2). However, the standard Wordle layout algorithm is random, violating R5, and it is, on the whole, not computationally efficient (R3) due to high-resolution hit checking. On the other hand, it *is* possible to design a Wordle layout implementation that also fulfills R3 and R5.

3.3.3 Cloud Layout Algorithm

Figure 3 shows a schematic overview of our Wordle [29]-inspired tag cloud layout algorithm that fulfills all of the above requirements. The algorithm accepts a set of ranked keywords $T = \mathbb{W} \times \mathbb{R}$, a bounding box B, and an optimal center c (a point for node clouds, and a line for link clouds). Its output is a size and position for all keywords $t \in T$. Figure 5 shows an example layout computed for William Shakespeare's play *A Midsummer Night's Dream.*

Below we sketch the general steps of the algorithm:

1. Create graphical representations of all keywords $t \in T$ and scale their size depending on their frequency/rank.

2. Sort the graphical keywords by descending width.

3. Initialize the priority queue of available free space with the full bounding box B (with a distance of zero).

4. For each graphical keyword (until unable to layout more):

   (a) Find the available free space in the priority queue that is closest to the optimal center c, yet which is big enough to fit the current keyword.

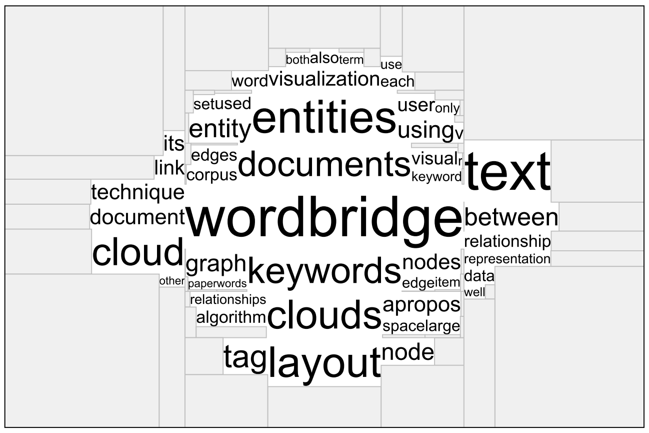


Figure 4: Our Wordle cloud layout for 50 keywords taken from this paper. Light gray rectangles represent the remaining free space.

   (b) Position the current keyword in the found free space so as to minimize the distance to the optimal center.

   (c) Create up to four rectangles of available free space by reclaiming whatever space is not used by the newly positioned keyword in the found free space.

   (d) Add the new rectangles of free space to the queue.

Because the above algorithm only uses the bounding box for the graphical keywords, the resulting tag cloud is not nearly as aesthetically pleasing as for a real Wordle tag cloud. Furthermore, the iterative subdivision of free space into smaller and smaller rectangles means that the algorithm will not make perfect use of the available space—the algorithm will not be able to lay out a word that fits across several adjacent free space rectangles, but which is too large to fit on any single one. Figure 4 shows the remaining free space after the layout algorithm has finished.

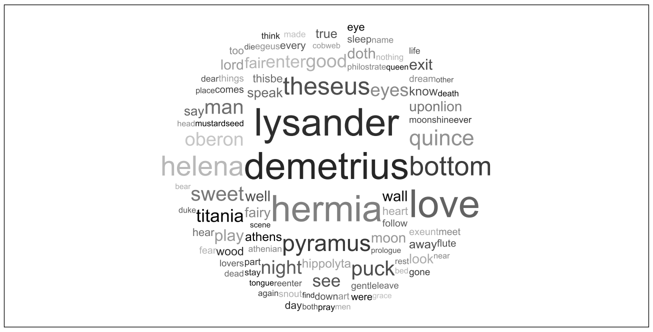
However, because of its simplicity and efficient use of a spatial index, the above algorithm is significantly faster than the full-fledged Wordle layout (R3), and is also deterministic (R5).

3.3.4 Borders

Unlike standard tag clouds [2, 29], as well as extended graph labels [34], a WordBridge needs explicit borders around both node clouds or link clouds to give firm delineations between each cloud. The simplest way of achieving this is to compute the convex hull for each cloud and use this as a border. Furthermore, filling the hull with a specific color (different for node and link clouds) will help users visually separate the clouds from each other.

3.4 Interaction

Graph Interaction. WordBridges are visually much more complex than standard node-link graphs, exacerbating many of the


(a) Optimal center point.

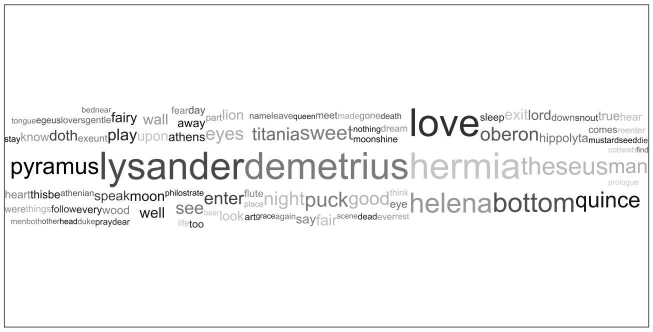

(b) Optimal center line (horizontal).

Figure 5: Example of our Wordle [29] layout for William Shakespeare's *A Midsummer Night's Dream* (point vs. line layouts, random colors).

problems [7] of graph visualizations even further. A large number of concurrently visible nodes and links quickly cause high visual clutter [10] and large amounts of overlap between the different clouds. Therefore, WordBridge visualizations work best with a query-based exploration approach where only a small subgraph of the whole graph is shown at any point in time. Left-clicking on a node will expand to add all of its neighbors to the currently visible set, and right-clicking will collapse the neighbors in turn.

Figure 2 shows a small subgraph of the InfoVis co-authorship network. We use an interactive force-directed graph layout for the high-level graph structure. Users can interact with the graph itself by dragging to move nodes around on the canvas.

Cloud Interaction. Due to the visual clutter introduced by link clouds, WordBridge links are by default collapsed into lines when they first appear on the screen. Figure 1 shows all links except those to Ulysses himself as being collapsed into lines. However, left-clicking on the link will expand it to a full link cloud. Analogously, right-clicking on an expanded cloud will collapse it back into a line.

Because our cloud layout algorithm efficiently supports variable size for a cloud, we can even implement a level-of-detail interaction technique for both link and node clouds. For example, rolling the mouse wheel over a cloud could "inflate" and "deflate" the cloud by incrementally adding or removing keywords. To ensure consistency, keywords should be added in order of descending frequency (most frequent non-visible keyword first), and removed in order of ascending frequency (least frequent visible keyword first).

3.5 Visual Scalability

The WordBridge visual representation is more visually complex than its constituent parts: node-link diagrams and tag clouds, respectively. The most serious effect of this more complex visual representation is naturally that its visual scalability is decreased. This is a common trade-off, and many of our interaction techniques above have been designed to cope with this by enabling people to expand and collapse individual link and node clouds. It is also the reason for the query-based interaction used by our implementation.

4 Implementation

The WordBridge technique was implemented in the context of *Apropos*, a web-based text analysis application designed to show relationships between entities extracted from a text corpus. In this section, we give an overview of Apropos, discuss our text analysis components, and give some notes on its implementation.

4.1 System Overview

Apropos (Figure 6) consists of an entity database, a graph visualization based on the WordBridge technique, and some user interface mechanisms for browsing entities and managing the analysis history. The entity database is accepted as a collection of XML documents that were created from a document collection in a preprocessing step (see the next section). The database consists of entities (i.e., a vertex set V), their relationships (i.e., an edge set E), and keywords for both entities and their relationships (i.e., $T_v(v)$ for all $v \in V$, and $T_e(v_1, v_2)$ for all $(v_1, v_2) \in E$).

After loading a database, the application populates its entity list with all of the loaded entities. This list is ordered by the degree of each entity on the assumption that highly connected nodes are often of high importance in a dataset. Users can select any of the available entities and choose to add them to the visualization canvas.

Adding an entity to the canvas will also add all of its neighbors in the form of a simple WordBridge (Figure 6). According to the discussion in Section 3.4, link clouds in WordBridge are initially collapsed into lines to reduce the visual clutter. Clicking on a link will expand it, however. Furthermore, node clouds are always expanded, showing the keyword set $T_v(v)$ for a visible vertex v. The layout of the graph is controlled using an interactive force-directed layout, meaning that the user can grab and drag any of the nodes to rearrange the layout, causing the rest of the layout to update.

The user can now freely expand the WordBridge to other neighbors of visible nodes by clicking an entity or collapse a branch by clicking on an expanded entity. Entities can also be removed entirely from the WordBridge network by double-clicking the node.

Once the user has explored the dataset and begun to grasp important entities, he or she can right-click on an entity to read the actual documents from where it was extracted. Because an entity often appears in many documents, Apropos will show a list of documents which are related to the entity. By the same token, when one of tags in a node cloud is clicked, Apropos will show a list of documents which are only related to that specific tag. When clicking on a tag in a link cloud, on the other hand, Apropos will list those documents that caused the edge to be created, i.e., documents where the keyword and both connected entities appear.

This iterative visual query approach supports exploring relationships in a text database with details-on-demand [24]. For instance, when exploring a newspaper dataset, if the user finds two entities *George W. Bush* and *Barack Obama* connected by a link cloud with the term *insurance*, the user can click on the tag to read the actual documents that mentioned both presidents together in that context.

4.2 Text Analysis

We use text analysis methods to create entity-relationship networks from a large text corpus.

4.2.1 Managing Document Collections

Our target dataset is document collections consisting of many, often smaller, text documents. This allows us to extract characteristic keywords from the smaller documents, and use co-occurrence of

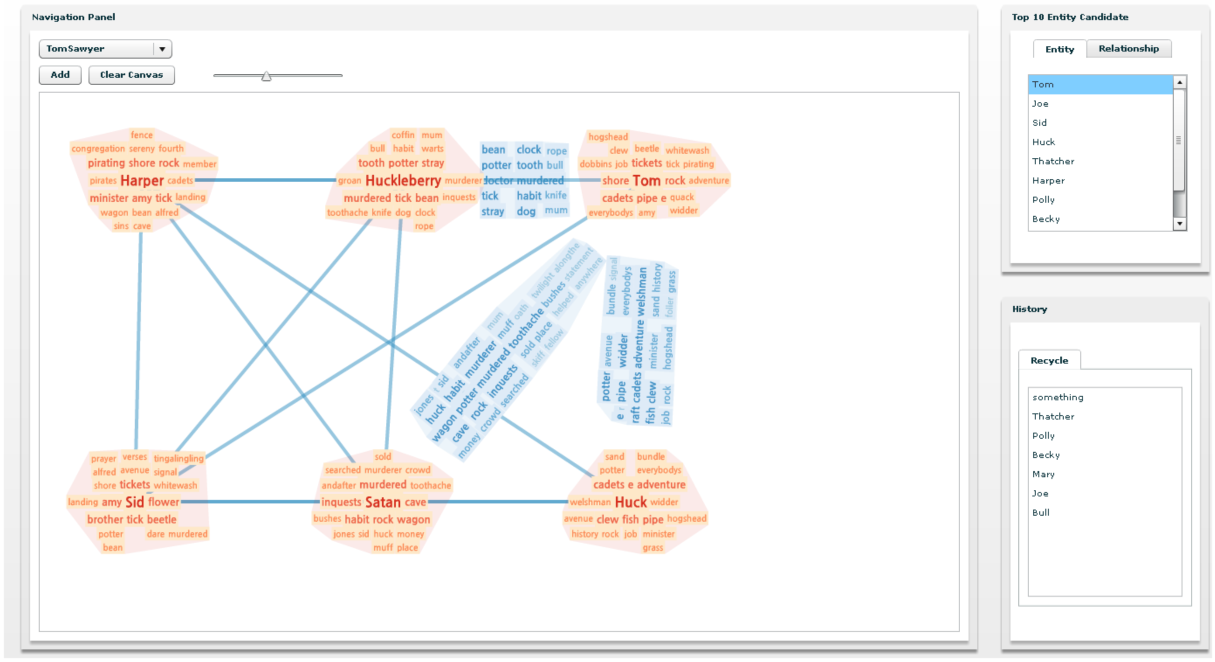


Figure 6: Screenshot of our Apropos web-based text analysis system. The main visualization window shows the state of the current subgraph visualized using WordBridge. An entity list (top right) and a history list (bottom right) support analytical tasks on the entity collection.

entities to build the graph (see below). However, some datasets are not of this form, but instead consist of a single large document (or a few large documents). For example, works of fiction (such as the examples discussed later on in this paper) often have this form.

To cope with this case, we use any inherent finer-scale structure that exists in the document. For example, a large report may be split into sections or subsections, and a novel may be split into chapters, paragraphs, or even sentences. These snippets of text (we will simply call them documents in the remainder of this paper) will then constitute the basic unit of analysis for the Apropos system.

4.2.2 Extracting Graph Structure

In our Apropos implementation, we use the concept of an *entity* [14] as nodes and extract the high-level graph structure from the co-occurrence of entities in a document. In other words, there is an edge between two entities A and B if they co-occur in the same document anywhere in the text corpus.

Of course, the problem now becomes how to identify the entities in the corpus. We use two different approaches for this problem:

- When the entities in a corpus are well-defined, such as for a list of press releases from Fortune 500 companies where the companies are the entities, we can manually list the entities.

- When entities are unknown or too numerous to specify manually, we use standard entity identification [4, 15] techniques for automatically deriving the entities in the corpus.

We currently use OpenCalais [20] for the latter approach.

4.2.3 Extracting Keywords

With the high-level graph structure $G = (V,E)$ in place, we must now find the keyword sets T_v and T_e for the nodes and edges, respectively. We use simple tf-idf [22, 23] to extract characteristic keywords for all documents in the corpus. However, these ranked keywords are specified on a per-document basis.

To generate keyword sets on a per-entity and per-relationship basis, we combine the top-ranked keywords for all documents where the entity occurs (T_v), or where both entities involved in the relationship occur (T_e), respectively. Because tf-idf values are meaningful for comparison across documents, we simply use the frequency of this union of per-document keywords to rank the sets.

4.3 Implementation Notes

Apropos is implemented in the Adobe Flex 3.5 environment and uses the Flare toolkit[1]. Using Flare and Flex allows the application to be easily exported to the Web as a Flash application. We have tested the system with document corpora consisting of up to 5,000 documents (3 MB of text), but this is not a hard limit. Because the system has a query-based approach and does not endeavor to visualize all documentss at once, we believe the system is capable of managing much larger corpora.

Data for the web application is constructed using a preprocessor implemented in Java. In terms of performance, the preprocessor is able to analyze the 5,000 document collection mentioned above in less than 10 minutes. However, this preprocessing step is done only once prior to deploying a dataset on the web.

5 Case Studies

Beyond the examples shown earlier in this paper, we implemented three in-depth scenarios to showcase the utility of the WordBridge.

5.1 The Adventures of Tom Sawyer

The first scenario was a visual exploration of the novel *The Adventures of Tom Sawyer* written in 1876 by Mark Twain. We divided the book into 36 documents by chapter and then used OpenCalais [20] to identify entities. We limited the entitiy identification to include only people, a total of 58 separate entities. We then extracted keywords using tf-idf, and sorted them by importance.

[1]http://flare.prefuse.org/

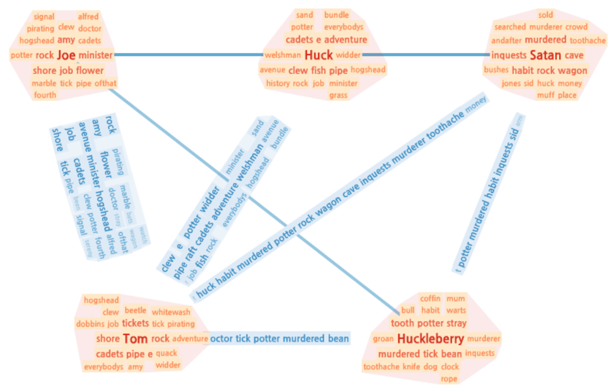


Figure 7: WordBridge visualization showing part of the social network from *The Adventures of Tom Sawyer* by Mark Twain.

The goal was to have an overview of main character in the novel. The degree-sorted list in Apropos not surprisingly suggested the entity “Tom” (Sawyer) as the most dominant entity, followed by (Injun) “Joe”, “Sid” (Sidney), and ’Huck’ (Huckleberry Finn). We start exploring from the ’Tom’ entity (Figure 7), and it characterizes ’Tom’ with terms such as “adventure”, “pirating”, and “pipe”. By clicking the link between Tom and Joe, the link is expanded into a link cloud that shows that the two entities are strongly related. The cloud also gives an idea why the entities are related through keywords such as “murdered”, “doctor”, and “potter” (Tom Sawyer and Huckleberry Finn become witness to Injun Joe murdering Dr. Robinson and blaming his partner Potter for the crime).

5.2 King James Bible

In this scenario, the goal was to find a brief notion of the relation between the entity “Lord” and other important entities. We used an XML-tagged version of King James Holy Bible with persons, places, and organizations identified.

“Lord” is—not surprisingly—the most dominant entity in this dataset. In Figure 8, we can see a few dominant tags that describe Lord, such as “David” (the second king of Israel), “spear” (possibly referring to the gigantic spear of Goliath, who David slew), and “Bethlehem” (the birthplace of Jesus). Furthermore, their link cloud displays tags which describe their relation, including names such as “David”, “Adonijah” (the fourth son of David and attempted usurper of David’s throne), and “Solomon” (another son of David and the chosen heir to the throne) that explain their connection.

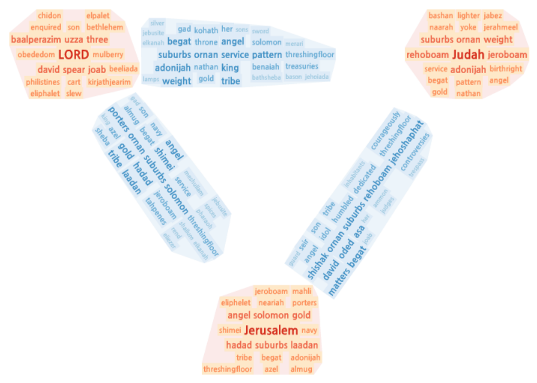


Figure 8: Three entities and their relations from the King James Bible.

5.3 Blue Iguanodon

The last scenario was a set of intelligence reports, and the goal of this scenario was to gain some idea about a slightly larger dataset without any prior knowledge. We used the *Blue Iguanodon* dataset from the VAST 2007 challenge with entities already identified.

Our analysis started with the most dominant entity “U.S.” and its neighbors, as suggested by Apropos. Figure 9 shows a large number of tags related to terrorism and food. The “Washington” node cloud includes *agroterrorism*, *feed*, and various animals. In addition, the link cloud for the relation between U.S and Washington include other animals as well as the tags *meat*, *usbeef*, and *tyson*, all food-related terms. WordBridge seems to suggedt that the dataset describes a terrorist threat involving food in the United States.

6 Conclusion and Future Work

We have presented a novel graph-based visualization technique for externalizing not only entities in a large document collection, but also their relationships beyond mere co-occurrence information. The technique is called WordBridge, and works by forming an actual bridge of words that connects one entity to the other using composite tag clouds in the shape of nodes and links. We have implemented the technique in the context of a web-based text analysis tool called Apropos, and show some examples of how to use the technique, including for intelligence reports, Mark Twain’s *The Adventures of Tom Sawyer*, and the King James Bible.

Text visualization is an exciting area of research with much potential and direct applicability to a wide array of domains. In particular, these techniques may become useful for settings where compact summaries of a large collection of documents are necessary—in other words, the type of settings studied in this paper. We are interested in applying these ideas to other types of document collections, including scientific articles, legal documents, and webpages. Finally, we have only presented case studies of WordBridge in this paper, but formal user studies may be necessary to evaluate the technique’s utility in comprehending large text corpora.

Acknowledgments

This material is based upon work supported by the U.S. Department of Homeland Security’s VACCINE Center under award no. 2009-ST-061-CI0001. Many thanks to the reviewers for their feedback.

References

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.

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

[3] M. J. A. Berry and G. Linoff. *Data Mining Techniques*. Wiley, 1997.

[4] M. Chau, J. J. Xu, and H. Chen. Extracting meaningful entities from police narrative reports. In *National Conference on Digital Government Research*, 2002.

[5] C. Collins, M. S. T. Carpendale, and G. Penn. DocuBurst: Visualizing document content using language structure. *Computer Graphics Forum*, 28(3):1039–1046, 2009.

[6] C. Collins, F. B. Viégas, and M. Wattenberg. Parallel tag clouds to explore faceted text corpora. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*, pages 91–98, 2009.

[7] G. DiBattista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall PTR, 1998.

[8] A. Don, E. Zheleva, M. Gregory, S. Tarkan, L. Auvil, T. Clement, B. Shneiderman, and C. Plaisant. Discovering interesting usage patterns in text collections: integrating text mining with visualization. In *Proceedings of the ACM Conference on Information and Knowledge Management*, pages 213–222, 2007.

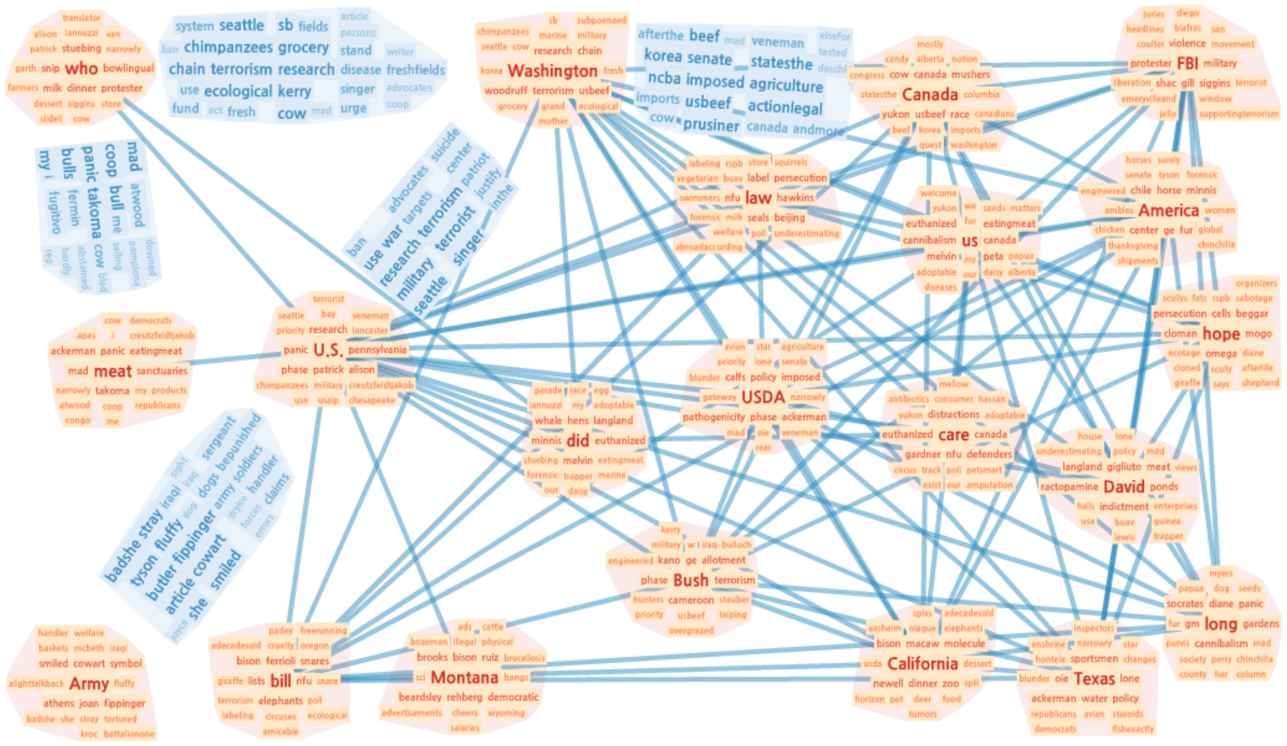


Figure 9: Part of the VAST 2007 competition dataset (Blue Iguanodon) visualized using WordBridge.

[9] P. Eades. A heuristic for graph drawing. *Congressus Numerantium*, 42(10):149–160, Jun 1984.

[10] G. Ellis and A. J. Dix. A taxonomy of clutter reduction for information visualisation. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1216–1223, 2007.

[11] M. Ghoniem, D. Luo, J. Yang, and W. Ribarsky. NewsLab: Exploratory broadcast news video analysis. In *Proc. IEEE Symposium on Visual Analytics Science & Technology*, pages 123–130, 2007.

[12] M. Grems. A survey of languages and systems for information retrieval. *Communications of the ACM*, 5(1):43–46, Jan. 1962.

[13] S. Havre, E. Hetzler, P. Whitney, and L. Nowell. ThemeRiver: Visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):9–20, Jan. 2002.

[14] A. Hotho, A. Nürnberger, and G. Paass. A brief survey of text mining. *LDV Forum*, 20(1):19–62, 2005.

[15] P. Jackson and I. Moulinier. *Natural Language Processing for Online Applications: Text Retrieval, Extraction & Categorization*. John Benjamins, 2002.

[16] S. K. Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.

[17] K. Kageura and B. Umino. Methods of automatic term recognition: a review. *Terminology*, 3(2):259–289, 1996.

[18] A. A. Kennings and K. Vorwerk. Force-directed methods for generic placement. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(10):2076–2087, 2006.

[19] C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, 1999.

[20] OpenCalais. `http://www.opencalais.com`. Access Mar 2010.

[21] S. E. Robertson and K. S. Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27:129–146, 1976.

[22] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[23] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw Hill, 1983.

[24] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the IEEE Symposium on Visual Languages*, pages 336–343, 1996.

[25] H. Strobelt, D. Oelke, C. Rohrdantz, A. Stoffel, D. A. Keim, and O. Deussen. Document cards: A top trumps visualization for documents. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1145–1152, 2009.

[26] F. van Ham, M. Wattenberg, and F. B. Viégas. Mapping text with phrase nets. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1169–1176, 2009.

[27] F. B. Viégas, S. Golder, and J. Donath. Visualizing email content: portraying relationships from conversational histories. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*, pages 979–988, 2006.

[28] F. B. Viégas and M. Wattenberg. Tag clouds and the case for vernacular visualization. *interactions*, 15(4):49–52, 2008.

[29] F. B. Viégas, M. Wattenberg, and J. Feinberg. Participatory visualization with Wordle. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1137–1144, 2009.

[30] R. Vuillemot, T. Clement, C. Plaisant, and A. Kumar. What's being said near 'martha'? exploring name entities in literary text collections. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*, pages 107–114, 2009.

[31] M. Wattenberg. Arc diagrams: Visualizing structure in strings. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 110–116, 2002.

[32] M. Wattenberg and F. B. Viégas. The word tree, an interactive visual concordance. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1221–1228, Nov./Dec. 2008.

[33] J. A. Wise, J. J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow. Visualizing the non-visual: Spatial analysis and interaction with information from text documents. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 51–58, 1995.

[34] P. C. Wong, P. Mackey, K. Perrine, J. Eagan, H. Foote, and J. Thomas. Dynamic visualization of graphs with extended labels. In *Proceedings of IEEE Symposium on Information Visualization*, pages 73–80, 2005.