Rapid and Reusable Text Visualization and Exploration Development with DELVE

Daniel R. Harris[1, 2], Ramakanth Kavuluru, Ph.D[2,3], Jerzy W. Jaromczyk, Ph.D[2], and Todd R. Johnson, Ph.D[4]

[1]Center for Clinical and Translational Sciences, University of Kentucky, Lexington, KY 40506.
[2]Department of Computer Science, University of Kentucky, Lexington, KY 40506.
[3]Institute of Biomedical Informatics, University of Kentucky, Lexington, KY 40506.
[4]School of Biomedical Informatics, University of Texas Health Science Center at Houston, Houston, TX 77030.

Abstract

We present DELVE (Document ExpLoration and Visualization Engine), a framework for developing interactive visualizations as modular Web-applications to assist researchers with exploratory literature search. The goal for web-applications driven by DELVE is to better satisfy the information needs of researchers and to help explore and understand the state of research in scientific literature by providing immersive visualizations that both contain facets and are driven by facets derived from the literature. We base our framework on principles from user-centered design and human-computer interaction (HCI). Preliminary evaluations demonstrate the usefulness of DELVE's techniques: (1) a clinical researcher immediately saw that her original query was inappropriate simply due to the frequencies displayed via generalized clouds and (2) a muscle biologist quickly learned of vocabulary differences found between two disciplines that were referencing the same idea, which we feel is critical for interdisciplinary work. We discuss the underlying category-theoretic model of our framework and show that it naturally encourages the development of reusable visualizations by emphasizing interoperability.

Introduction

The rapid pace of modern biomedical research has yielded a seemingly endless supply of peer-reviewed literature that is readily available in digital libraries. Despite general ease of access, the sheer quantity of material is a barrier for experts wishing to maintain an up-to-date understanding of their field, and the numbers suggest that is not feasible to read most or all of the material in a subspecialty [1]. Because there is an excess of information to manually review, computational tools play a pivotal role by allowing experts to ingest summarized or targeted subsections of the available literature [2, 3]. It is not only the depth of information that is problematic but also the breadth and reach of topics that makes it crucial to understand the needs of a diverse population of researchers and to create computational tools capable of assisting in such a large variety of aims and goals [4].

Online digital libraries such as Pubmed [5] have been greatly successful in creating an online source of information for biomedical researchers; these libraries also excel as a source of information for computational tools to attempt to enhance or augment the literature review and dissemination experience [6]. A review of Pubmed-based applications identified twenty-eight unique systems and placed them into four general categories: ranking and search results, clustering results into topics, extracting and displaying high level semantic entities and relations, and improving search engine and retrieval experience [6]. Since the completion of this specific application review, trends indicate that applications geared toward extracting and displaying high level semantic entities and relations [7–11] outweigh those that cluster [12] or generally improve the search engine and retrieval experience [13]. All of these research initiatives are successful in their own right but lack interoperability, making the novel ideas embedded into each system difficult to reuse. Information visualization has been shown to be useful in uncovering and communicating ideas [14], yet only three applications of the original twenty-eight reviewed contained some type of visualization component [6]. A review of text visualization publications identified over one hundred methods of visualizing text and created a taxonomy for categorizing visualization techniques [15]; each technique in some way contributes toward communicating raw data in a more effective manner. Outside of applications that leverage Pubmed, efforts exist to visualize large collections of data and provide analytical windows on top of raw data [16, 17].

We wish to provide a framework that does not depend solely on a single visualization technique but rather provides a suite of possible techniques that can work together in harmony. Existing online text visualization tools [17, 18] are difficult to extend, either because of their closed-source nature or because they lack a natural route of integrating other visualizations that might also

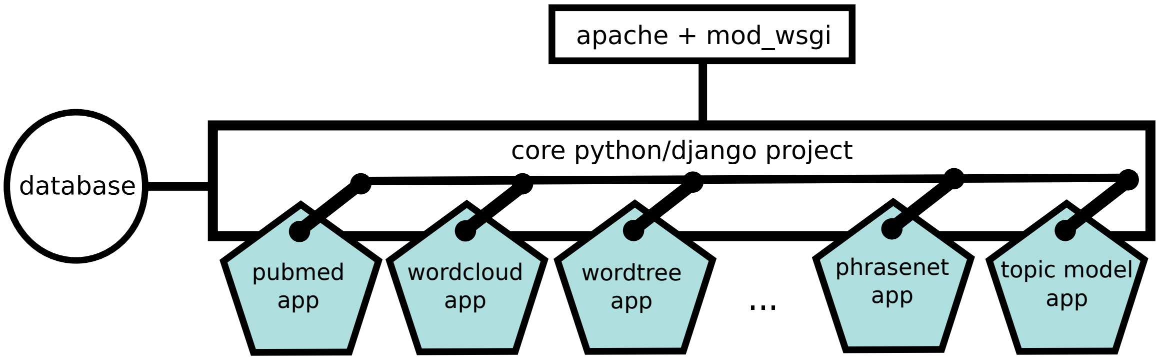

Figure 1: DELVE is a series of modular web applications, where each application maintains interoperability with the others via a common faceted data structure.

be helpful. We propose DELVE (Document ExpLoration and Visualization Engine) as a general framework for interoperable and modular development of light-weight web-based applications geared toward exploring and visualizing large collections of texts in a manner that strongly supports interoperability and reuse. We will also discuss how DELVE is a special case of the category-theoretic model of faceted browsing, which is known to enable the design of reusable and interoperable systems [19, 20].

Methods: The DELVE Framework

We created DELVE as a framework for developing visualizations of biomedical text rapidly; additionally, we also stress interoperability and reuse. In this section, we will first give an overview of DELVE and how its components are constructed; we then describe the theoretical underpinnings in its design and the practical consequences that follow.

We base our prototype upon fulfilling the needs of principles from user-centered design and human-computer interaction (HCI), in particular Shneiderman's information visualization mantra: overview first, filter and zoom, provide details on demand [21]. More specifically, DELVE's application programming interace (API) is capable of yielding both summarized information and the corresponding lower-level details of text documents. Filtering and zooming is supported by allowing a dynamic level of detail with each facet of information.

DELVE is implemented using the Python-based Django [22] web-development framework and is made available online as open-source software [23]. Given a Pubmed query, facets of the resulting publications are exposed via the DELVE API as JSON (Javascript Object Notation) [24]. These JSON files, which carry either aggregate or detailed information, are used to bootstrap visualizations created with d3 (Data Driven Documents) [25]. As seen in Figure 1, each web-application is a modular unit that ties into a common Django model that is responsible for communicating with the raw data residing in the database and exposing the necessary JSON. The seed that begins the DELVE workflow is a Pubmed query that is completely compatible with Pubmed's robust query engine, i.e. supporting tags such as *[Mesh Terms]*; the added value is that the abstracts and meta-data corresponding to these results are exposed via the DELVE API to feed directly into d3 visualizations. Exposing the results via JSON enables rapid development and prototyping visualizations and alternative or supplemental search experiences. Per query results, meta-data components such as publication details and authorship details are available; the text of the abstract is exposed as either (1) JSON lists of unigrams, bigrams, trigrams, and MeSH terms, or (2) partial or complete sentences depending on need.

Visualizing Text

As proof of concept, we implemented text visualizations known to be effective: word clouds [26], phrase nets [27], and word trees [28]; we are also experimenting with integrating topic model analysis. Each of these visualizations offers something different: clouds give frequency of words or phrases, word trees give the context surrounding a word or phrase, and phrase nets give relationships between words or phrases. Many more text visualization techniques exist [15], but the three we have selected provide a baseline for understanding the contents of publications matching a particular query.

Word clouds are visualizations where common words found within a collection of text are drawn with size relative to their frequency so that the most frequent words are drawn the largest. Clouds are not without controversy [29] but have been demonstrated to be useful in research settings [9, 10, 30, 31]. We generalized the concept of clouds so that they can show frequency of MeSH (Medical Subject Headings) terms [32], unigrams, bigrams, trigrams, and important phrases, which can be used to filter out or focus in on certain documents within the particular query being explored. Figure 2 shows an example of a MeSH cloud for a Pubmed query

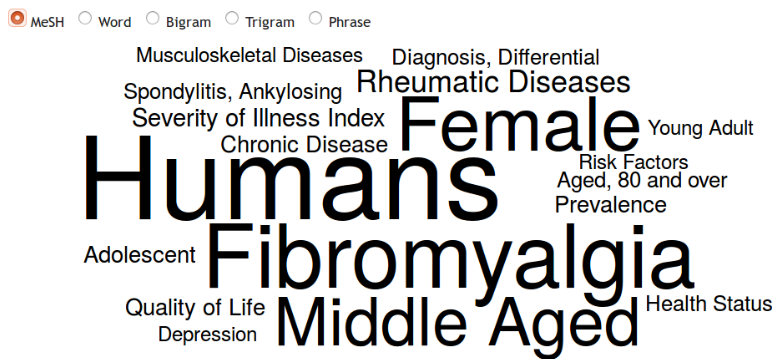

Figure 2: A MeSH cloud shows frequency of MeSH terms attached to a collection of articles. The above MeSH cloud was generated from documents returned from a Pubmed query regarding fibromyalgia.

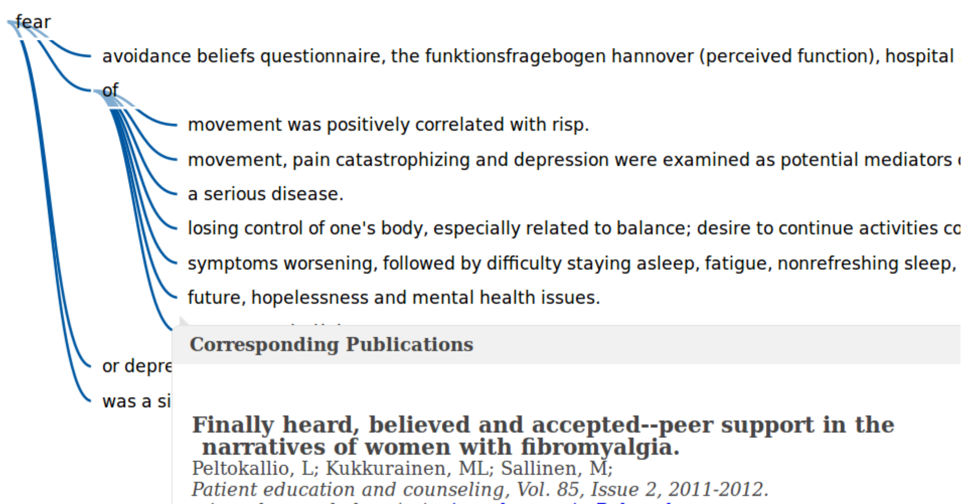

Figure 3: Word trees show where a chosen word or phrase appears in text. The root of this word tree is *fear* and the sentences shown correspond to documents returned from a Pubmed query regarding fibromyalgia.

for fibromyalgia; for example, clicking *ankylosing spondylitis* would show those articles that were also assigned the *ankylosing spondylitis* MeSH term. Common MeSH terms such as *humans* could be pre-filtered if desired.

Word trees are graphs that show where a chosen word or phrase appears in a body of a collection of texts [28]. Every occurrence is grouped together either by what precedes the word or by what follows it up to a configurable maximum height. For example, Figure 3 shows a word tree with a root word of *fear* drawn from the abstracts of documents returned from a Pubmed query on fibromyalgia. This example shows that the word *fear* is commonly followed by the word *of* which in turn is followed by a variety of fears. The document that a chosen sentence occurs in can be easily displayed after clicking a specific branch of the tree; if more than one document contains this sentence or sentence fragment, a list is presented. In this particular case, only a single article was found having *fear of* as a parent of *future, hopelessness, and mental health issues*. If clouds give frequency of certain words or phrases, word trees are a supplemental visualization that gives the context surrounding such words or phrases.

Additionally, phrase nets [27] could display local co-occurrences between words or concepts in order to give insight on relations embedded into the corresponding documents. We are looking at topic model analysis in order for topics to be visualized across sets of documents. Semantic knowledge bases also offer a source of additional information that could be visualized and presented as part of the suite, as later discussed in the results section.

Developing with the DELVE Framework

The DELVE API is easily extended if a data element is not already exposed via JSON for consumption. If a data element was missing, one would need only to initialize the application within the Django framework and add the view logic that produces the

JSON from the raw data so that it gets exposed per query at a chosen URL. For example, suppose publication date was not already available but we wish to produce a histogram of publication dates for articles corresponding to any given Pubmed query. The views in this new component would take the raw data and transform it into the JSON required by the application. Specifically, the JSON:


```
[{"total": 1, "journ_pub_year":2006},
 {"total": 1, "journ_pub_year":2007},
 {"total": 1, "journ_pub_year":2008},
 {"total": 2, "journ_pub_year":2010},
 {"total":27, "journ_pub_year":2011},
 {"total":55, "journ_pub_year":2012},
 {"total": 3, "journ_pub_year":2013}]
```


corresponds to the histogram in Figure 4 and is a great example of how basic histograms can imply much about trends within a research area. In other words, the presentation or visualization layer is strategically separated from the view layer that is responsible for deciding what data elements from the model should be exposed.

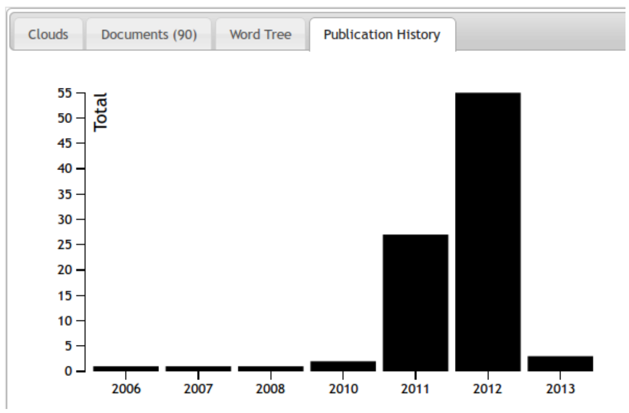

Figure 4: A basic histogram of publication quickly reveals temporal trends in topic popularity.

Interoperability and Reuse

Our DELVE search tool is a special case of a faceted browsing system [33] where facets control visualizations but also the visualizations contain facets that can be selected, such as any of the words or phrases displayed by the clouds, word trees, or phrase nets. When we first began working on DELVE, we were motivated by unifying visualizations for exploratory search for a common purpose and simply implemented a selection of visualizations for a specific group of publications that interested a clinical researcher in our collaboration. Although successful as a proof of concept that encouraged us to move forward, we had great difficultly in communicating abstractly how our visualizations were actually transforming the raw data into usable modules that contribute to the larger aim of exploring text collections. At the same time, we identified difficulties in reusing existing work and visualizations in the same area due to the variety of theoretical foundations used to create such systems [19].

We solved the issue of communication and the issue of reusability by creating an abstract model of faceted browsing based upon category theory that is able to consume and unify other theoretical models of faceted browsing [19]. Specifically in the biomedical domain, heterogeneous terminologies play an important role in exploring and presenting data [34] and the category-theoretic model of faceted browsing gives us precise language for describing how instances of terminologies can be used as facets. We briefly describe the category-theoretic model of faceted browsing and give examples on how it can help communicate the abstractions upon which DELVE depends.

A Category-Theoretic Model of Faceted Browsing

In this section, we present the underlying category-theoretic model important for our framework. However, we give only the minimum details needed to understand the category-theoretic model of faceted browsing and leave the additional details as related work [19, 20, 34]. Informally, a category C is a collection of objects and relationships (or morphisms) between the objects, where concepts of identity and composition are defined and where the relationships between objects obey identity and associativity laws [35].

Definition 1. *A category C consists of the following:*

1. *A collection of objects, $Ob(C)$.*
2. *A collection of morphisms (also called arrows). For every pair $x, y \in Ob(C)$, there exists a set $Hom_C(x, y)$ that contains morphisms from x to y; a morphism $f \in Hom_C(x, y)$ is of the form $f : x \to y$, where x is the domain and y is the codomain of f.*
3. *For every object $x \in Ob(C)$, the identity morphism, $id_x \in Hom_C(x, x)$, exists.*
4. *For $x, y, z \in Ob(C)$, the composition function is defined as follows: $\circ : Hom_C(y, z) \times Hom_C(x, y) \to Hom_C(x, z)$.*

Given 1-4, the following laws hold:

1. *identity: for every $x, y \in Ob(C)$ and every morphism $f : x \to y$, $f \circ id_x = f$ and $id_y \circ f = f$.*
2. *associativity: if $w, x, y, z \in Ob(C)$ and $f : w \to x$, $g : x \to y$, $h : y \to z$, then $(h \circ g) \circ f = h \circ (g \circ f) \in Hom_C(w, z)$.*

For example, **Set** is the category of sets as objects and functions between sets as morphisms. Another well-known category is **Rel**, the category of sets as objects and relations as morphisms, where we define relation arrows $f : X \to Y \in Hom_{\mathbf{Rel}}(X, Y)$ to be a subset of $X \times Y$. **Cat** is the category of categories. The objects of **Cat** are categories and the morphisms are functors (mappings between categories). We can define a slimmer version of **Rel**, called **Tax**, where we know exactly what binary relation is being used to order the objects. This represents a general taxonomic structure.

Definition 2. *Let **Tax** be a sub-category of **Rel**, the category of sets as objects and relations as morphisms where $Ob(\boldsymbol{Tax}) = Ob(\boldsymbol{Rel})$ and let the morphisms be the relations that correspond only to the $\subseteq$ relations. The identity and composition definitions are simply copied from **Rel**.*

We refer to facet types as the disjoint substructures within a large taxonomy. The actual objects within the category are sets of resources that are assigned to belong to this particular facet type. For example, genre would be a facet type for a book or age could be a facet type for a person. Within that facet type, selection is possible. For example, someone could select romance and comedy as a genre; we call this a focused selection.

Definition 3. *A facet type (a facet i and its related sub-facets) of a faceted taxonomy is a sub-category of **Tax**, the category of sets as objects and inclusion relations as morphisms. Let us call this sub-category $\boldsymbol{Facet}_i$ and let $Ob(\boldsymbol{Facet}_i) \subseteq Ob(\boldsymbol{Tax})$ with the morphisms being the corresponding $\subseteq$ relations for those objects. The relevant identity and composition definitions are also copied from **Tax**.*

Definition 4. *We can define a subcategory of $\boldsymbol{Facet}_i$, called $\boldsymbol{Focus}_i$, to represent a focused selection of objects from $\boldsymbol{Facet}_i$ having $Ob(\boldsymbol{Focus}_i) \subseteq Ob(\boldsymbol{Facet}_i)$ and the necessary corresponding morphisms, identity, and composition definitions for those objects.*

We can designate the collection of all facet types as a category that is very similar to **Cat**. Note that this structure is inherently different than **Tax** in that we have a larger structure that contains discrete substructures.

Definition 5. *Let **FacetTax** be a category that represents a faceted taxonomy, whose objects are the disjoint union of $\boldsymbol{Facet}_i$ categories. In other words, let $Ob(\boldsymbol{FacetTax}) = \bigsqcup_{i=1}^{n} \boldsymbol{Facet}_i$ and $n = |Ob(\boldsymbol{FacetTax})|$. The morphisms of **FacetTax** are functors (mappings between categories) of the form $Hom_{\boldsymbol{FacetTax}}(C, \mathcal{D}) = \{F : C \to \mathcal{D}\}$.*

The category-theoretic model of faceted browsing is useful because we can be specific in our communication. For example, we can give a precise definition for what it means to be a faceted query:

Definition 6. *A faceted query, Q, is the modified n-ary product [19] within the **FacetTax** category, defined as $\prod_{i=1}^{n} \boldsymbol{Focus}_i$, where $n = |Ob(\boldsymbol{FacetTax})|$. The n coordinates of Q are similarly defined as projection functors $P_j : \prod \boldsymbol{Focus}_i \to \boldsymbol{Focus}_j$.*

Impact on Implementation and Reuse

It is easy to convert other faceted browsing systems into the category theoretical model because there exists representations of sets, graphs, lattices, and other common data structures in category theory. In other words, category theory provides a common language for describing and modeling faceted systems. Because DELVE is a framework designed to enable development, reuse is encouraged in two ways: (1) consistent abstractions imply that novel ideas and features of applications formulated in the common language can be exchanged freely and (2) these ideas can be implemented in a common manner so that they are interoperable in practice.

Facet is providing the specification for what it means to be a facet; instances of **Facet** can be created for DELVE where $I_0, I_1, I_2, \ldots, I_N$ represent N collections of objects whose data are classified according to specific relationships, which is needed in systems where more than one faceted taxonomy can be leveraged [34]. An instance of MeSH and an instance of ICD9 could be utilized in visualizations requiring both MeSH terms and diagnostic codes.

In our related work, we demonstrated that the **Facet** category is structurally equivalent to **Schema**, the category-theoretic view of database schemas [34]. Both categories describe the conceptual layout that organizes information: rows/entities for databases and resources for facets. At the core of **Schema**, there exists primary key to foreign key relationships of which we can map facet and ancestor relationships so that we can easily implement faceted browsing in a relational database system. At a minimum, this requires two tables: one with the faceted structure and one with the relationships between facets and resources. The table for the faceted structure is at a minimum a two column table with a primary key representing a facet and a self-referential foreign key representing its ancestor.

Formalizing Visualization Computation

The model clearly provides structure for the taxonomy and the resources, but it also provides a way to formalize computations. As a simple example, consider MeSH clouds. Recall that each object of **Facet** is a set of resources that has been classified as belonging to that facet. If I_0 is an instance of the facet type for MeSH, then we can easily generate a sequence of frequencies for objects $x_1, x_2, \ldots \in Ob(I_0)$ by simply considering their cardinality:

Definition 7. *Given an instance of a facet I_0, for $x \in Ob(I_0)$, let the frequency of an object x be defined as freq(x) = $|x|$. Let freq(Ob(I_0)) be a n-tuple of frequencies: freq(Ob(I_0)) = $(freq(x_0), freq(x_1), \ldots, freq(x_n))$ where $n = |Ob(I_0)|$.*

Given that we can know all of the frequencies for facets in a facet type, we are free to use that information in other calculations. There are different strategies for computing font size within a cloud, taking either the raw frequency or the square root of the frequency and mapping that to a range of font sizes [26]. One possible solution is linear scaling with an offset:

$$s = \frac{x}{max(freq(Ob(I_0)))} + \varepsilon$$

where x is a given frequency from $Ob(I_0)$ and ε is a desired offset that could enforce a minimum font-size if desired; if not, ε can be set to 0. For example, if the scale of the font size ranges from 0 to 1 and the maximum frequency was 100, then our offset scale s given a frequency x would be calculated as $s = x/100 + \varepsilon$. Looking at boundary cases, words with the maximum frequency found would scale at $1 + \varepsilon$ and words with the smallest frequency possible would scale at $1/100 + \varepsilon$, assuming a minimum frequency of 1.

Results: Driving Exploratory Search with Visualizations

The visualizations discussed in the previous section are good examples of taking well-known text visualizations such as word clouds and word trees and turning them into modular applications. By themselves, these applications serve a very specific purpose: word clouds provide frequency and word trees provide context. Together, these applications can join cohesively into a single interface designed to take the Pubmed query results and immerse a researcher with a visual index of relevant publications. At this point, the visualizations generated with DELVE can either stand alone as they are or can be incorporated into a larger web application. Because we want to augment the researcher's exploratory search with additional information that would otherwise be difficult to see via traditional search methods, a centralized search portal that integrates all visualizations together is desirable.

We implemented a search tool using DELVE that offers configurable window panes of visualizations for a given Pubmed query; a sample session for a query for *fibromyalgia* and *rheumatoid arthritis* is shown in Figure 5-A. The general workflow consists of a Pubmed query returning associated documents that act as input for the visualization suite. Individual panes house each visualization in a separate tab. As seen in Figure 5, the layout of the panes is configurable so that the user can arrange the offered visualizations as they deem most effective. For example, in Figure 5-D, four different types of clouds are displayed at once: MeSH, unigram, bigram, and trigram; each cloud is distinct in the list of possible words or phrases that could be displayed and that could align with what the researcher is seeking. If the researcher is familiar with MeSH terms and a MeSH term exists for what they wish to explore, then MeSH clouds may be suitable. The other clouds attempt to fill in the information gaps; in the discussion section, we detail how one search strategy may impact results and how different clouds can assist the researcher.

The visualizations are linked together so that an interaction in one pane has consequences in another. The most useful interaction implemented is the ability to focus on a word, bigram, trigram, or MeSH term; focusing selects only those articles that either contain the chosen entry or in the case of MeSH terms, only those articles that were assigned that particular MeSH term. The visualizations operate on a set of documents; if the set of documents is manipulated by the interface with an action like focusing, the visualization is updated. Since the visualization is blind to the process, the interface is free to provide any type of interactivity that filters or zooms the data.

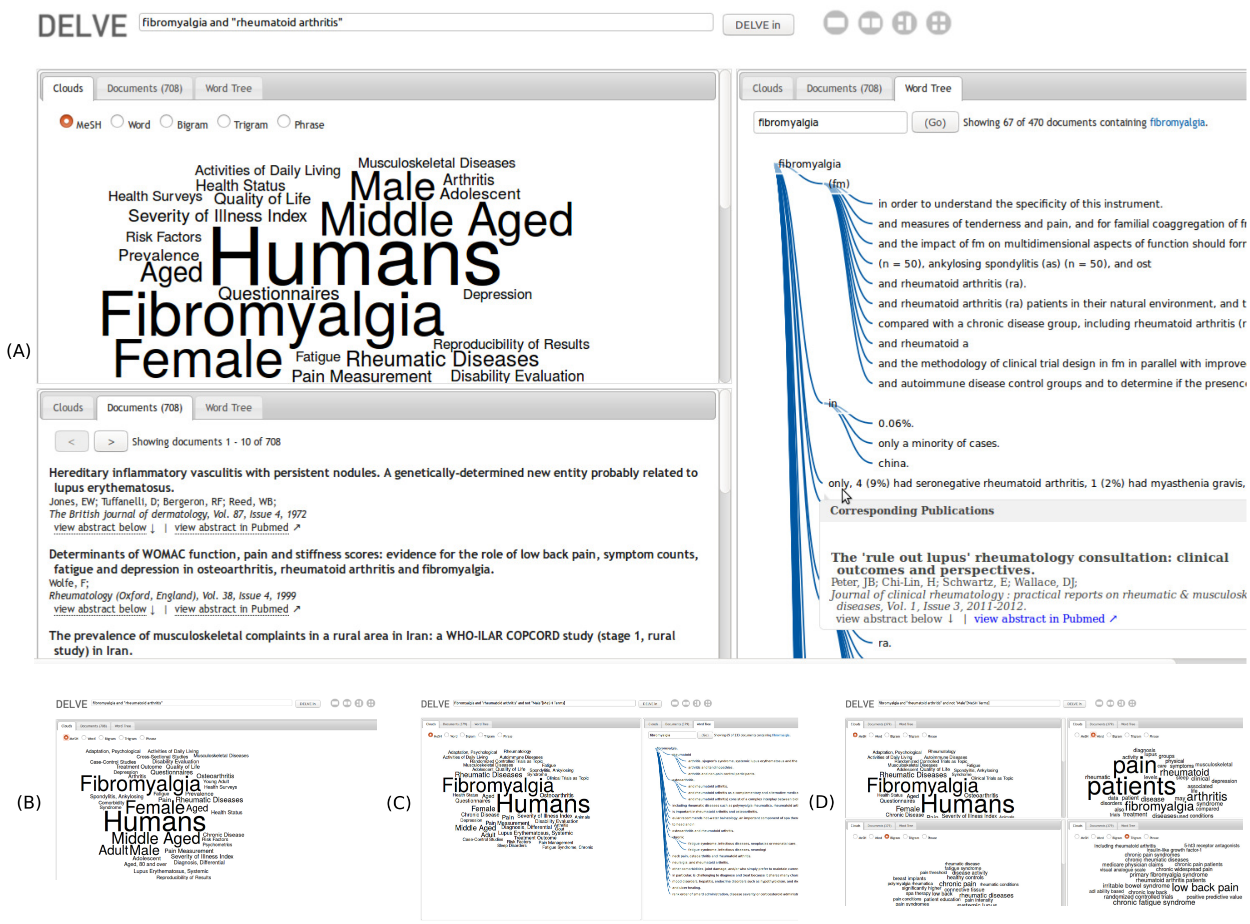

Figure 5: Our prototype combines different DELVE applications into a single user experience by providing integrated visualizations that collectively enable a researcher to explore collections of texts. Both the number of panes and the layout is configurable by the user in DELVE: (A) shows a sample three-pane view with clouds, word trees, and a document list, (B) shows a simple one-pane view that shows only a cloud, (C) adds a pane to show word trees, and (D) shows that different types of clouds can be viewed at once.

Discussion

It is known that users struggle to successfully refine queries in search-based systems simply from looking at the number of attempts per query [36]. Because most search systems return a limited list of top search results, there is routinely not enough information presented to determine if a query was incorrect or sub-optimal, outside of the extreme case where all results appear unrelated. DELVE attempts to correct this and make query refinement easier by providing clouds that clearly display frequency of terms or phrases for all documents being returned by this query. This gives the user feedback pertaining to the entire body of documents being returned rather than only the first *n*-publications for those systems that return a ranked top-*n* list.

Scenario: Sub-optimal Search Strategy

For example, suppose that a researcher is interested in *chronic fatigue syndrome* and its relationship with *functional somatic syndromes*. The phrase *functional somatic syndromes* is not available as a MeSH term and articles are usually assigned a more general *somatoform disorders* MeSH term, yet DELVE provides trigrams as an alternative to MeSH terms and *functional somatic syndromes* occurs in high frequency, as seen in the clouds A1 and A2 from Figure 6. Without multiple lenses to inspect summarized and aggregated data, it may be difficult for the researcher to reconcile what his or her needs are against what the machine reports and understands. Surprisingly, a search for only *functional somatic syndromes* yields a significant number of results for *chronic fatigue syndrome*, as seen in clouds B1 and B2 from Figure 6. This implies that the researcher's search strategy plays a large role in how successful they may be in finding relevant articles; to combat against a user's unintentionally sub-optimal search strategy,

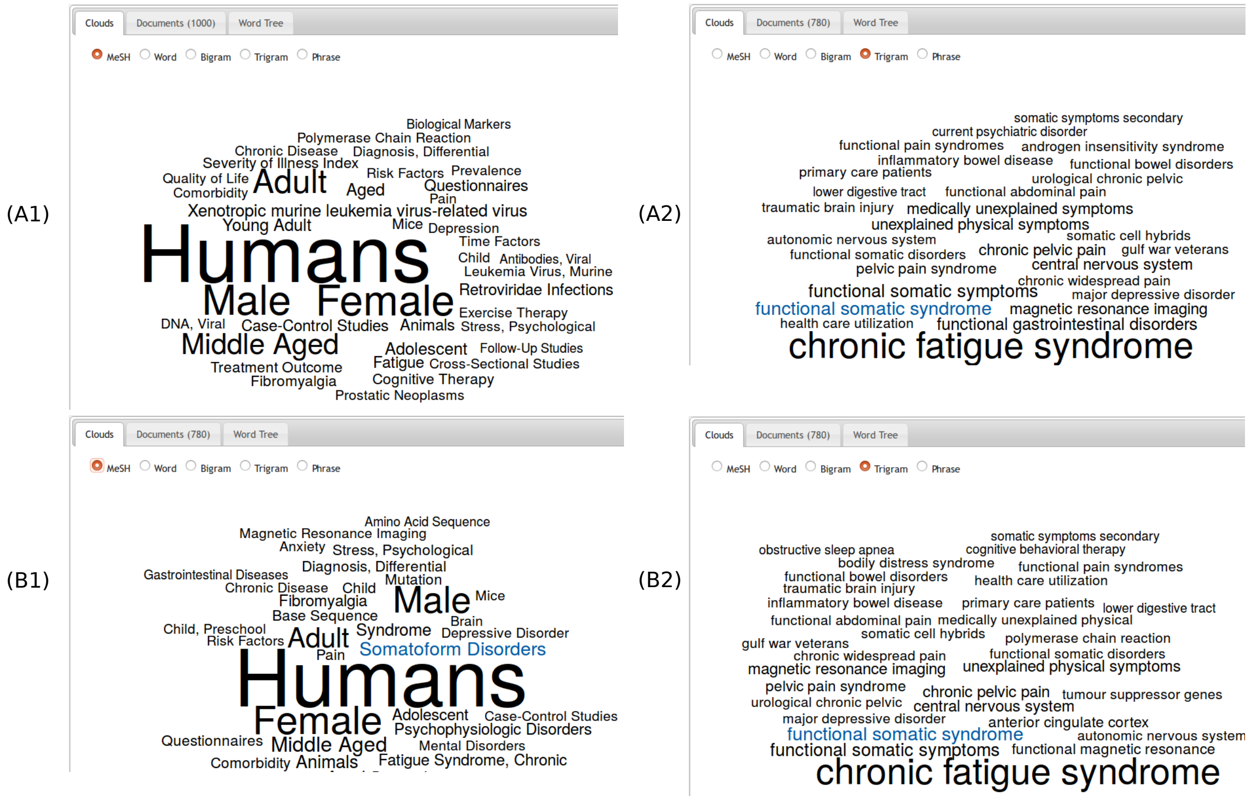

Figure 6: Differences between A1/A2 (MeSH and trigram clouds for *chronic fatigue syndrome*) and B1/B2 (MeSH and trigram clouds for *functional somatic syndromes*) illustrate the need for multiple lenses when searching; the MeSH cloud in A1 is missing the term *somatoform disorders* because it did not reach the minimum frequency required to be illustrated.

multiple lenses placed over the data potentially compensate for weaknesses in any single lens. For example, trigram clouds usually return phrases that do not overlap MeSH terms and may match what a researcher desired.

Scenario: Poisoned Queries

In our preliminary evaluation of our DELVE prototype, a clinical researcher immediately saw that her original query was inappropriate simply due to the frequencies displayed by the clouds; it was determined that, by including the word *inflammation* inside a conjunction, the query results were being poisoned with results related to the human liver for a disease that typically has very little to do with the liver. The refined and corrected query contained approximately twenty percent fewer documents and was ultimately deemed a more successful query by the researcher. In Figure 7, we give an example on how feedback is supplied back to the user. The goal is to help the researcher understand why this article was included in the results; if the article is not appropriate, the user can modify their query to attempt to calibrate the results.

Scenario: Term Unfamiliarity

Similarly, a researcher may struggle with unfamiliarity of the terms used by an alternative discipline within an interdisciplinary collaboration. A muscle biologist was successfully able to reconcile differences between his own terminology and that of a biomechanical engineer by being presented with both frequencies and the associated context given by word trees. We also wish to explore how researcher preference and technical background plays a role in seeking information; because the interface provides a variety of analytic windows on top of the raw data, user choice becomes a pivotal element in whether or not they are successful in finding the information they seek.

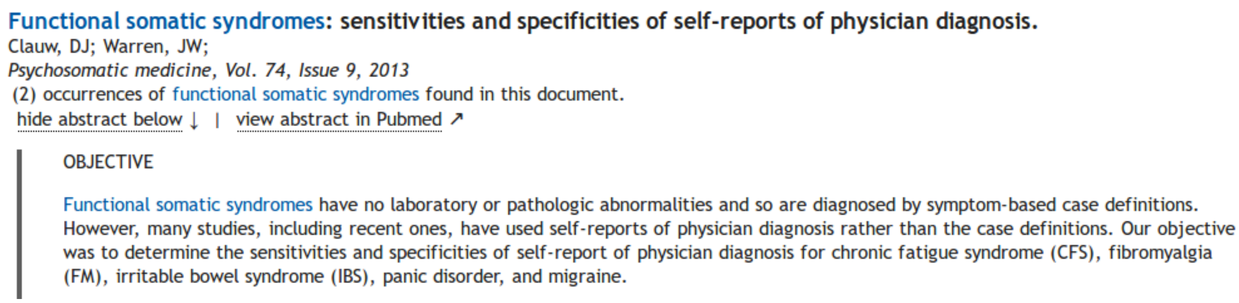

Figure 7: Terms and phrases can be selected as a point of focus; feedback such as highlighting the focused term or phrase must be given to the user so that they understand why this article was correctly or incorrectly included in the search results.

Conclusion

We presented DELVE, a framework for developing interactive visualizations as modular Web-applications; we targeted biomedical publications available via Pubmed to assist researchers with exploratory search for research. As future work, we would like to explore how transitioning from biomedical abstracts to the full texts of the articles would change DELVE's utility. This concept could work with any collection of texts, including those from outside of the biomedical domain. We demonstrated that DELVE is a special case of the category-theoretic model of faceted browsing; in this case, visualizations both contain facets and are driven by facets. This abstract framework enables the consistent design and implementation of reusable and interoperable DELVE applications. We also presented a publicly available prototype that demonstrates and integrates several DELVE-based visualizations. Preliminary evaluation indicated that DELVE was helpful in tuning a researcher's query for appropriateness and for helping cross barriers in interdisciplinary research by providing access to multiple lexicons understood by opposing fields. We are working to expand our library of DELVE applications in order to provide a complete suite of interoperable visualizations that immerse a researcher into an environment where research needs are easily serviced.

Acknowledgments

The project described was supported by the NIH National Center for Advancing Translational Sciences through grant numbers UL1TR001998 and UL1TR000117. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

References

[1] Fraser AG, Dunstan FD. On the impossibility of being expert. BMJ. 2010;341:c6815.

[2] Hunter L, Cohen KB. Biomedical language processing: what's beyond PubMed? Molecular cell. 2006;21(5):589–594.

[3] Kavuluru R, Thomas C, Sheth AP, Chan V, Wang W, Smith A, et al. An up-to-date knowledge-based literature search and exploration framework for focused bioscience domains. In: Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium. ACM; 2012. p. 275–284.

[4] Dogan RI, Murray GC, Névéol A, Lu Z. Understanding PubMed® user search behavior through log analysis. Database. 2009;2009:bap018.

[5] Using PubMed. U.S. National Library of Medicine; 2016. Available from: http://www.ncbi.nlm.nih.gov/pubmed.

[6] Lu Z. PubMed and beyond: a survey of web tools for searching biomedical literature. Database. 2011;2011:baq036.

[7] Song M. Exploring concept graphs for biomedical literature mining. In: 2015 International Conference on Big Data and Smart Computing (BIGCOMP). IEEE; 2015. p. 103–110.

[8] Lee K, Shin W, Kim B, Lee S, Choi Y, Kim S, et al. HiPub: translating PubMed and PMC texts to networks for knowledge discovery. Bioinformatics. 2016;Available from: http://bioinformatics.oxfordjournals.org/content/early/2016/08/20/bioinformatics.btw511.abstract.

[9] Zhang Y, Sarkar IN, Chen ES. PubMedMiner: mining and visualizing MeSH-based associations in PubMed. In: AMIA Annual Symposium Proceedings. vol. 2014. American Medical Informatics Association; 2014. p. 1990.

[10] Kim S, Yeganova L, Wilbur WJ. Meshable: searching PubMed abstracts by utilizing MeSH and MeSH-derived topical terms. Bioinformatics. 2016;p. btw331.

[11] Cameron D, Kavuluru R, Rindflesch TC, Sheth AP, Thirunarayan K, Bodenreider O. Context-Driven Automatic Subgraph Creation for Literature-Based Discovery. Journal of biomedical informatics. 2015;54:141–157.

[12] Médoc N, Ghoniem M, Nadif M. Exploratory Analysis of Text Collections Through Visualization and Hybrid Biclustering. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer; 2016. p. 59–62.

[13] Ernst P, Siu A, Milchevski D, Hoffart J, Weikum G. DeepLife: An Entity-aware Search, Analytics and Exploration Platform for Health and Life Sciences. ACL 2016. 2016;p. 19.

[14] Fekete JD, Van Wijk JJ, Stasko JT, North C. The value of information visualization. In: Information visualization. Springer; 2008. p. 1–18.

[15] Kucher K, Kerren A. Text visualization techniques: Taxonomy, visual survey, and community insights. In: 2015 IEEE Pacific Visualization Symposium (PacificVis). IEEE; 2015. p. 117–121.

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

[17] Koch S, John M, Wörner M, Müller A, Ertl T. VarifocalReaderâĂŤIn-Depth Visual Analysis of Large Text Documents. IEEE transactions on visualization and computer graphics. 2014;20(12):1723–1732.

[18] Viegas FB, Wattenberg M, Van Ham F, Kriss J, McKeon M. Manyeyes: a site for visualization at internet scale. IEEE transactions on visualization and computer graphics. 2007;13(6):1121–1128.

[19] Harris DR. Foundations of reusable and interoperable facet models using category theory. Information Systems Frontiers. 2016;18(5):953–965. Available from: `http://dx.doi.org/10.1007/s10796-016-9658-6`.

[20] Harris DR. Modeling Reusable and Interoperable Faceted Browsing Systems with Category Theory. In: Information Reuse and Integration (IRI), 2015 IEEE International Conference on. IEEE; 2015. p. 388–395.

[21] Shneiderman B. The eyes have it: A task by data type taxonomy for information visualizations. In: Visual Languages, 1996. Proceedings., IEEE Symposium on. IEEE; 1996. p. 336–343.

[22] Django. The Django Software Foundation; 2016. Available from: `https://www.djangoproject.com/`.

[23] DELVE. Bitbucket.org; 2016. Available from: `https://bitbucket.org/_harris/delve`.

[24] Crockford D. The application/json Media Type for JavaScript Object Notation (JSON); 2006.

[25] Bostock M, Ogievetsky V, Heer J. D^3 data-driven documents. IEEE transactions on visualization and computer graphics. 2011;17(12):2301–2309.

[26] Viegas FB, Wattenberg M, Feinberg J. Participatory visualization with Wordle. IEEE transactions on visualization and computer graphics. 2009;15(6):1137–1144.

[27] Van Ham F, Wattenberg M, Viégas FB. Mapping text with phrase nets. IEEE Trans Vis Comput Graph. 2009;15(6):1169–1176.

[28] Wattenberg M, Viégas FB. The word tree, an interactive visual concordance. IEEE transactions on visualization and computer graphics. 2008;14(6):1221–1228.

[29] Harris J. Word clouds considered harmful. Nieman Journalism Lab. 2011;.

[30] Sarkar IN, Schenk R, Miller H, Norton CN. LigerCat: using "MeSH clouds" from journal, article, or gene citations to facilitate the identification of relevant biomedical literature. American Medical Informatics Association; 2009. .

[31] Baroukh C, Jenkins SL, Dannenfelser R, Ma'ayan A. Genes2WordCloud: a quick way to identify biological themes from gene lists and free text. Source code for biology and medicine. 2011;6(1):1.

[32] Lowe HJ, Barnett GO. Understanding and using the medical subject headings (MeSH) vocabulary to perform literature searches. Jama. 1994;271(14):1103–1108.

[33] Wei B, Liu J, Zheng Q, Zhang W, Fu X, Feng B. A survey of faceted search. Journal of Web engineering. 2013;12(1-2):41–64.

[34] Harris DR. Modeling Integration and Reuse of Heterogeneous Terminologies in Faceted Browsing Systems. In: Information Reuse and Integration (IRI), 2016 IEEE 17th International Conference on. IEEE; 2016. p. 58–66.

[35] Spivak DI. Category Theory for the Sciences. MIT Press; 2014.

[36] Spink A, Wolfram D, Jansen MB, Saracevic T. Searching the web: The public and their queries. Journal of the American society for information science and technology. 2001;52(3):226–234.