applied sciences

MDPI

Article

# Semantic Network Analysis Pipeline—Interactive Text Mining Framework for Exploration of Semantic Flows in Large Corpus of Text

**Martin Cenek [1,*,†], Rowan Bulkow [2], Eric Pak [3], Levi Oyster [3], Boyd Ching [3] and Ashika Mulagada [1]**

1 Computer Science, University of Portland, Portland, OR 90203, USA; mulagada21@up.edu
2 Resource Data, Inc., Anchorage, AK 99503, USA; rowan.bulkow@gmail.com
3 Computer Science, University of Alaska Anchorage, Anchorage, AK 99508, USA; epak3@alaska.edu (E.P.); loyster1@alaska.edu (L.O.); byching@alaska.edu (B.C.)
* Correspondence: cenek@up.edu; Tel.: +1-503-943-7524
† Current address: 5000 N Willamette Blvd., Portland, OR 97203, USA.

Received: 5 June 2019; Accepted: 19 November 2019; Published: 5 December 2019


**Abstract:** Historical topic modeling and semantic concepts exploration in a large corpus of unstructured text remains a hard, opened problem. Despite advancements in natural languages processing tools, statistical linguistics models, graph theory and visualization, there is no framework that combines these piece-wise tools under one roof. We designed and constructed a Semantic Network Analysis Pipeline (SNAP) that is available as an open-source web-service that implements work-flow needed by a data scientist to explore historical semantic concepts in a text corpus. We define a graph theoretic notion of a semantic concept as a flow of closely related tokens through the corpus of text. The modular work-flow pipeline processes text using natural language processing tools, statistical content narrowing, creates semantic networks from lexical token chaining, performs social network analysis of token networks and creates a 3D visualization of the semantic concept flows through corpus for interactive concept exploration. Finally, we illustrate the framework's utility to extract the information from a text corpus of Herman Melville's novel *Moby Dick*, the transcript of the 2015–2016 United States (U.S.) Senate Hearings on Environment and Public Works, and the Australian Broadcast Corporation's short news articles on rural and science topics.

**Keywords:** semantic concept; text mining; computational linguistics; language processing; natural language processing; interactive visualization

## 1. Introduction

Historical semantic concepts (HSC) modeling aims to understand what the key concepts discussed in a text corpus are, how concepts evolve over time, and what the context semantic concepts are used in is in relation to each other as well as their relation to the supporting sub-concepts. Although semantic networks can be used to capture the relationships among co-occurring words in a single document [1,2], interactive HSC exploration requires multi-step, computational linguistic work-flow to process the unstructured text to extract information from many documents in order to synthesize knowledge about the different concepts found in the corpus of text. This data science process relies on mature under-laying tools to process and simplify text using Natural Language Processing (NLP) tools, create the semantic networks and analyze them using Social Network Analysis (SNA) and generate an interactive visualization tool to explore the HSC.

Key phrases and textual memes are often equated to semantic concepts, which does not satisfy our notion of HSC to extract rich relationships within and across semantic concepts. In particular,

existing text analysis techniques often extract a set of discrete textual memes from a text corpus which does not preserve the meme's context, relationship to other meme(s), nor how these relationships change throughout a corpus of text. To illustrate these shortcomings, let us consider a toy example of three newspaper articles that were published sequentially on the topic of "salmon" and a set of key textual memes extracted from each article—environment, cost, salmon, economy, harvest, ecology, economy, investment, and global, economy, salmon, environment, cost. It is not clear analyzing the set of textual memes from the first article if the article's main thesis is on the "environmental cost" of the salmon harvest economy or on the "salmon ecology" and how it is effected by economy, or many other possibilities. Although the second article is categorized under the "salmon" topic, it is described by a limited number of key textual memes which does not include many of the memes from the first article. That said, salmon could be a textual meme with minor importance with respect to the overall article and is used as a supportive sub-topic when discussing the "investment". Finally, the textual meme "investment" could be in a supportive role of "environment" in the last article.

To address the lack of context of the key textual memes and how they relate to each other, we model the HSC using sets of related tokens as well as the relationships among sets in favor of using sets of discrete memes. For the above example, capturing additional relationships between salmon-cost, environment-ecology, salmon-harvest and harvest-ecology would disambiguate the first article's thesis. Also exploiting the role of the sub-topics or the textual memes with minor importance across the three newspaper articles would relate the key textual memes of "salmon" to "investment" and "investment" to "environment".

Our scientific contribution is the conceptualization of the *semantic concept flows* as a set-based construct that allows for a semantically rich exploration of a text corpus. The engineering contributions include a construction of an open-source, web service enabled, interactive platform that integrates existing text processing tools, proposed semantic concept flows, and immersive visualization.

At this point, we informally define a **semantic concept** as a group of high frequency co-occurring tokens (words) in a document. The **semantic concept flow** then refers to the dynamics of how these semantic concepts propagate through subsequent documents over time. The dynamics of semantic concept flows then include the splitting of a semantic concept into multiple sub-concepts in the subsequent document, or merging multiple sub-concepts from the previous document into a single semantic concept in a current document. This definition of the semantic concepts and the flows allows for much richer representation and exploration of text corpus that is well positioned between building full Markov models of natural language and simple term frequency-inverse document frequency (TF-IDF) statistics [2–4].

The research goal was to construct a linguistic processing pipeline that implements HSC work-flow from processing raw text to interactive 3D visualization in an effort to understand the HSC and dynamics of concept flows through the corpus. The resulting Semantic Networks Analysis Project (SNAP) is a modular architecture that interfaces with existing computational linguistic tools and allows a user to extend the framework by swapping or adding text-processing modules to augment the current work-flow [5]. SNAP's web-services implementation makes the framework a widely accessible computational linguistics, data science tool and an interactive visualization platform that requires no programming knowledge. The framework's project management allows for the inspection and validation of the intermediate text-processing steps, management of large data sets and provides data security.

## 2. Background

The field of Natural Language Processing (NLP) studies processing natural language (text) by machines to achieve tasks such as language understanding, machine cognition and perception, and dialogue systems to name a few. As a result, a variety of mature text-processing tools are available as open-source frameworks [6–9]. SNAP uses the NLP as a low-level, text-processing tool to analyze the words from the original text to produce the normalized tokens.

In 1958, Richard Richens introduced *Interlingual Machine Translation* as a semantic network-generating architecture that implements a computational linguistic representation for language translation [10]. Richens' project models the text's tokens as graph nodes and the relationships between tokens as the graph's edges. This concept of semantic networks [2] was used for text summarization [11–13], knowledge graph representation for Google's search engine optimization [14], and text understanding by evolving ontologies [15–17], to name a few.

The language ambiguity makes the word normalization into tokens using NLP tools difficult but feasible. For example, removing past tense from verbs or extracting a singular form of a noun. The language cognition, on the other hand, makes the automation of edge generation to model the meaning as the relationships between tokes very difficult. Another example, should the phrase "dusty sugar" be disambiguated as sugar with dust on it or a sugar that is in a dust form. In recent years, the availability of machine learning algorithms and large text corpora resulted in high accuracy models on low level linguistic processing tasks such as adjective disambiguation, parts-of-speech tagging, and sentence chunking [18–20]. That said, the machine learning approach relies on a large amount of text that is likely not available when exploring the semantic concepts in a set of documents. Instead, SNAP uses simple lexical chaining, first introduced by Barzilay, to represent spatio-temporal organization of unstructured text [21]. *Lexical chaining* defines an edge between any two tokens if the tokens occur next to each other in a sentence within a fixed number of neighbors. Lexical chaining has been successfully used for keyword extraction and text summarizing [21,22], semantic word disambiguation [16,23], and document tagging [24].

Graph theoretic algorithms are often used to analyze the semantic networks produced by lexical chaining for network structures that reveal additional semantic content. Steyvers et al. linked the emergence of network structures to the semantic content [25], while Ensan et al. used the semantic linking for document retrieval from a large corpus [26]. Graph representation of semantic content and a dictionary-based sense tagging was used to validate and disambiguate sense annotations [27].

Several open-source projects address the exploration of semantic concepts in a corpus of unstructured text. Overview Project uses a document's term frequency–inverse document frequency (TF-IDF) to create document trees, document tagging, coding and 2D visualizations of key words in documents [28]. AlchemyAPI, Document Cloud and Unstructured Information Management applications (UIMA) allow for document annotation and some content analysis [29–31].

Several projects rely on and visualize the aggregates of keyword analysis [32–35]. Stanford's Nifty system tracks and visualizes the flow of textual *memes* on the web in near real-time analysis [33]. An application of analogous methodology was used to analyze the news coverage of the United States' arts funding [36]. Dou et al. explored topic hierarchies using interactive visualization and clustering of a large text corpus with equally large number of topics [34]. Chaney et al. proposed a machine-learning-based document aggregation and thematic visualization based on the keyword probability distribution [35].

TextFlow is the only other project that analyzes how semantic topics evolve over time by selecting, ranking, and keyword-based tracking across multiple documents over time [37]. Although an aggregate, key-word based system, the project identifies when new flows originate, intersect with other topic flows, or cease to exist.

The visualization of the text analysis results is commonly presented as a 2D plot for a frequency-based approached and a directed acyclic graph for a hierarchical topic exploration. Chuang et al. proposed a two-fold, model-driven visualization of text analysis that captures the topical interpretation (data inference) and trust (accuracy) [38]. The Jigsaw project implements a hybrid visualization approach that combines graph-based topical clusters to capture content and term frequency tables to explore the topical hierarchies. Altaweel et al. models semantic flows across multiple documents as 3D semantic networks that combined TF-IDF and 3D key word flow visualization [39].

SNAP, in comparison to the previously proposed aggregate approaches, defined semantic concepts as high co-occurrences of content tokens rather then keywords or key phrases which allows for the semantic concepts to split and flow through corpus in time (note: the concept of time is loosely defined and can be inferred from the document's publication time stamp or generated artificially using the document's chapter number). The framework also allows the exploration of any given document in multiple dimensions—first, analyzing the semantic concepts in a single document, and second, using the semantic concept flows to analyze the context of the document's semantic concept in relation to the semantic concepts in temporally adjacent documents. In particular, how semantic concepts are related to each other is analyzed in each document, as well as how individual tokens within a single semantic concept are related to each other. Finally, the 3D visualization mutually interconnects the extracted semantic relationships that would otherwise be hard to analyze using 2D aggregate visualization.

## 3. Work-Flow

Semantic concepts emerge from a corpus of unstructured text as multiple computational linguistic tools are applied to the corpus. After a user authenticates into the SNAP framework and uploads the raw, time-stamped text files, the exploration of semantic concepts is an iterative process of adjusting system parameters used by the underlying linguistic tools to narrow down and extract the underlying semantic concepts in the text corpus. In addition to the final semantic flows, the work-flow can be verified by manual inspection of intermediate result files. Figure 1 illustrates the methodology used to implement the framework's work-flow. The headers denote each modular text-processing step that produces intermediate output files (not shown).

### 3.1. From Unstructured Text to Semantic Flows

Figure 1, step 2, shows the first text-processing step of NLP tools to normalize input words into tokens. The SNAP framework relies on the linguistic models that are native to the NLP framework used, and SNAP does not evolve or augment these models. SNAP currently uses Natural Language Toolkit (NLTK) for NLP processing, but it can be swapped for any other NLP framework [40]. For a detailed description and implementation of various NLP algorithms that process the raw text, please see the NLP framework documentation [40]. In this section, we will not cover the algorithmic details implemented by different NLP engines since the same NLP task can be implemented in multiple ways from dictionary look-up, stochastic search algorithms, or models built using machine learning. It is important to note, though, that the framework is designed to accommodate various NLP tool kits with varying algorithm implementations which will result in slightly different tokens.

The input text is first tokenized, which splits the stream of characters in the raw text file into groups that roughly delineate individual words. The tokenized words are stemmed to remove various inflected forms of the same word so all word variations can be analyzed as a single token. For example, the original word "identified" is stemmed into the present tense *identify*, noun plurals are removed "whales" → *whale*, and so forth. Each processed document is split into sentences that are then tagged for part-of-speech (POS). The words tagged by the same POS tag that span more than one word are combined into a single token. For example, "16 year old" is concatenated to *16yearold*, geographic and organization entities are also merged to a single token "Wildlife Service" → *wildlifeservice*, "human society" → *humansciety* and so forth. The words that were not combined by a POS tagging are lemmatized to produce word roots. Examples of this processing step include converting words such as "observations," "gloomy," "determination" to *observ*, *gloom* and *determin* respectively. The lemmatization is an essential step that reduces the many word-forms used to a single token and seeds the notion of a semantic concept; for example "domination," "dominate," "dominator," and "domineering" are all lemmatized to a single token *dominat*. The final step of the word-to-token conversion is the named entity recognition (NER) that will also combine multiple words into a single token. These examples include "Bird Ridge" → *BirdRidge*, "Senator Merkeley" → *senatormerkeley*, "United States Of America" → *unitedstatesofamerica*.

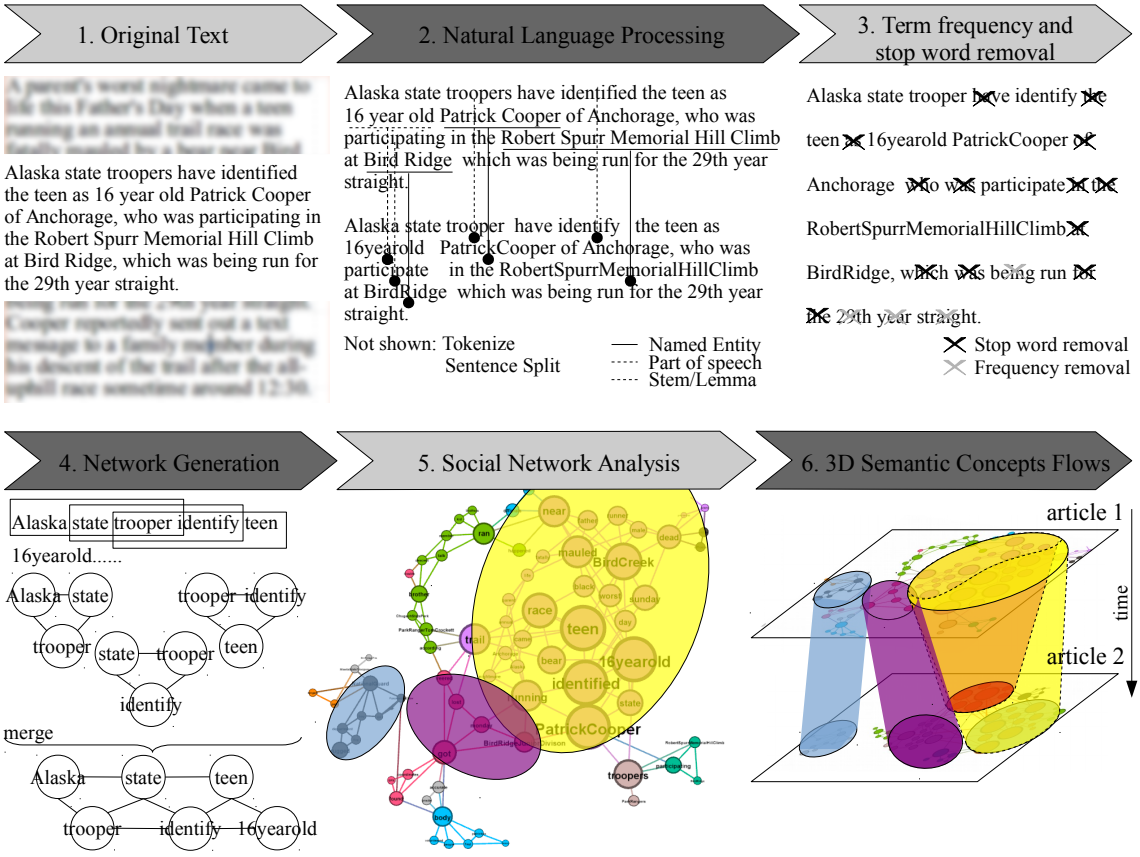


**Figure 1.** Illustration of SNAP's work-flow. (SNAP: Semantic Networks Analysis Project) (1) Original text of a single document uploaded for processing, (2) Natural Language Processing (NLP) step processes text by stemming, tokenizing, sentence splitting, part-of-speech tagging, lemmatizing and named entity recognition, (3) content narrowing by stop-word and frequency threshold token removal (TF-IDF—term frequency-inverse document frequency), (4) lexical chaining generation of semantic networks using sliding window (3 token window shown) and the merger of sub-networks into a document wide network, (5) Social Network Analysis of the document network. Detected communities of nodes (nodes with the same color) represent semantic concepts. Node's size proportional to its Eigenvector centrality, (6) 3D network of semantic flows is constructed by connecting matching semantic concepts across multiple documents. The illustration shows breakup of a yellow semantic concept into two sub-concepts in the subsequent document. The other two semantic concepts are propagated with minimal change across two sample documents.

Figure 1, step 3, illustrates the dictionary-based stop-word removal that removes tokens from the document that do not have a semantic meaning (e.g., "the," "from," "on" etc.). Additional tokens are removed based on the user-defined preference using lower and upper percentiles thresholds for the document's token frequencies (TF). The resulting set of tokens $T_d$ in a document $d$ is used to construct a graph $G_d$ representing the document's semantic network of the tokens and their co-occurrences in a sentence.

The semantic network generation using lexical chaining with sliding window size of three tokens is shown in Figure 1, step 4. Each unique token is represented as a network node. The undirected network edges are generated by creating connections between all pairs of tokens in a sentence within the fixed window size. The illustration shows sub-graphs generated for consecutive positions of the sliding windows at the top of the pane, while the bottom of the pane shows the composite graph that merged the constituent sub-graphs into a single graph. A semantic network is generated for each document by applying sub-graph generation for every sentence with the sliding window shifted by one token and creating a composite graph.

### 3.2. Semantic Concept

The semantic concepts are generated using Social Network Analysis's (SNA) community detection algorithm (Figure 1, step 5) [41–44]. The nodes $n$ that have high mutual inter-connectivity are colored the same color to mark a group of nodes belonging to the same structural network feature—a community C

$$C_{k,d} = \{n_i \mid n_i = Q_k, \ \forall \ i \ \epsilon \ T_d\} \tag{1}$$

is defined by the node membership in a community $k$ is $C_{k,d}$ that consists of all nodes (representing tokens in a document $d$) with the same modularity class $Q_k$ [43].

We equate the communities to semantic concepts as these structures identify nodes with common co-occurrences. Previously published research has shown that the semantic concepts emerge as the network's community structures and are identified as the semantic concept carrying structures [25]. Note that each node can be assigned to exactly one community in a document with a unique time stamp. The node size is proportional to the token's role within the network and can be selected to correspond to the node's connectivity (degree), Eigenvector centrality (centrality) [42,45] or betweenness centrality (bridges) [42,46].

The same way the SNAP's raw text processing relies on mature NLP tools, the semantic concept processing uses mature SNA tools to discover semantic concepts in the lexical chains of tokens. The SNAP framework uses Gephi open source software (OSS) for the SNA processing through its headless application programming interface (API) engine [42] which implements the above cited network analysis algorithms.

The above definition of a semantic concept extends the basic definition of key concepts that are commonly identified by the TF-IDF or manual coding by giving contextual information for each token. For example, Figure 1 shows the POS *16YearOld* being commonly used with *teen*, *mauled*, and *bear* within the same semantic concept, and *troopers* and *trail* (through *teen*) in adjacent semantic concepts. Modeling the semantic concepts using lexical chaining and subsequent SNA of the semantic networks retains the full richness of how semantic concepts relate to each other and the supporting concepts.

### 3.3. Semantic Flows

SNAP's final step is to create semantic flows by tracking and connecting the semantic concepts in temporally adjacent documents. In addition to the creation of the integrated text-mining SNAP framework, the notion of semantic concept flows is our main, novel contribution (in addition to the design and implementation of SNAP as an OSS framework). The semantic concept flows refer to the dynamics of how the semantic concepts behave in space and time by being split, merged, annihilated or originated. A semantic flow is created between two documents if the documents are adjacent to each other in time; implemented as a difference between two document time-stamps being less then a user defined threshold.

A semantic flow $F_{d_p,d_q}$ between two adjacent documents $d_p$ and $d_q$ is defined as a relationship between any two concepts $C$ that share nodes $n$

$$F_{d_p,d_q} = \{\exists n_i \mid n_i \ \epsilon \ C_{d_p,k_p} \wedge n_i \ \epsilon \ C_{d_q,l_q}, \ \forall \ k_p = 1 \ldots Q_p, \ l_q = 1 \ldots Q_q\}. \tag{2}$$

A semantic concept flow is implemented as a 3D mesh that connects two semantic communities. The algorithmic implementation of a semantic flow detection is by a simple token identity search in two consecutive document layers. If a token in a document $d_p$ is found in the subsequent document $d_q$, a semantic flow is added as an edge connecting the two communities $C_{d_p,k_p}$ and $C_{d_q,k_q}$ that the node $n_i$ belongs to. The dynamics of a semantic flow is not explicitly quantified; it is left for a user to explore using the framework's interactive 3D visualization.

Figure 1, step 6, shows two sets of identical tokens that appear in both temporally adjacent documents and belong to the same blue semantic concept, thus defining a perimeter of a new 3D mesh that connects these nodes and propagates this semantic concept through time.

On the other hand, the yellow semantic concept has the same tokens in two temporally adjacent documents 1 and 2, but not all tokens in the document 2 belong to the same community. The original semantic concept is therefore mapped to two semantic concepts in the subsequent document by generating two 3D meshes that track tokens that belong to a yellow community in document 1 but are assigned to two different semantic concepts in document 2. This mechanism of one-to-many mapping allows for semantic concept splitting. Formally, we define the semantic concept splitting as

$$F^{+}_{d_p,d_q} = \{\exists n_i,\ n_j \mid n_i,\ n_j\ \epsilon\ C_{d_p,k_p} \wedge n_i\ \epsilon\ C_{d_q,l_q} \wedge n_j\ \epsilon\ C_{d_q,m_q} \wedge l_q \neq m_q, \\ \forall\ k_p = 1 \ldots Q_p,\ m_q\ ,\ l_q = 1 \ldots Q_q\}, \tag{3}$$

which requires two nodes from the same semantic concept $C_{d_p,k_p}$ to belong to two different semantic concepts $C_{d_q,l_q}$ and $C_{d_q,m_q}$ in the subsequent document $d_q$. The semantic concept flow merge is defined symmetrically as

$$F^{-}_{d_p,d_q} = \{\exists n_i,\ n_j \mid n_i \epsilon\ C_{d_p,k_p} \wedge\ n_j \epsilon\ C_{d_p,o_p} \wedge n_i,\ n_j\ \epsilon\ C_{d_q,l_q} \wedge k_p \neq o_p, \\ \forall\ k_p,\ o_p = 1 \ldots Q_p,\ l_q = 1 \ldots Q_q\}, \tag{4}$$

where two nodes within the proceeding document $d_p$ that belong to two different semantic concepts will belong to the same concept flow $C_{d_q,l_q}$ in the document $d_q$. Visualization of the merging of two semantic flows can be seen when two or more distinct 3D meshes that track two or more semantic concepts tokens from the previous document are combined to one semantic concept flow in the current document (many-to-one mapping of semantic concepts).

A semantic flow is originated if tokens exist in the current document and the subsequent document(s) but do not appear in the previous document (formal definition is identical to Equation (2)). Finally, the annihilation of a semantic concept flow is the inverse of a semantic flow origination

$$F^{\emptyset}_{d_p,d_q} = \{\forall n_i |\ n_i \epsilon\ C_{d_p,k_p} \wedge n_i \epsilon T_{d_q} = \emptyset\}, \tag{5}$$

when none of the token nodes from a semantic concept $C_{d_p,k_p}$ can be found in the subsequent document. The 3D mesh representing such a semantic concept flow is not propagated to the subsequent document. Figure 1 illustrates the above-defined dynamics of the semantic concept flows.

After the semantic flows are generated, the 3D semantic networks have two types of edges within each document and a 3D mesh in between articles in time. The inter- semantic concept edges show how tokens within an individual semantic concept are related to each other in support of the central (key) tokens. The intra- semantic concept edges show how the key concepts relate to each other in the same document. Finally, the 3D meshes show the dynamics of concept flows over time.

### *3.4. 3D Network Visualization*

With 3D concept semantic concept flow representation, a user can view the overall concept landscape for the entire corpus and identify concept flow patterns with desired dynamics that would not be apparent when viewing one single document's semantic network at a time or when tracking individual key tokens through multiple documents. The interactive visualization enables an ad-hoc exploration of the semantic concepts by viewing (or flying through) the 3D semantic concept network, concealing or visualizing various network structures (such as nodes, inter- edges, intra-edges, 3D meshes, labels etc.), or visualizing the networks in a visualization theater or a cave, such as a planetarium [47].

### *3.5. Implementation Notes*

SNAP is implemented as a full-stack application and runs on Linux or MacOS Apache MySQL PhP (LAMP/MAMP) stack [5,48]. The framework's back-end database uses MySQL for user account, project, and configuration management. User accounts are created using two step-account creation and the MD5 hashed passwords are stored with user account information in the database. Some of the tested NLP tool kits also use the MySQL database back-end for meta-data and book-keeping management. With exception of the interactive visualization, the front-end web interface is implemented in PhP using CodeIgniter libraries, and the visualization layer uses JavaScript libraries to implement the interactive visualization as an in-browser applet [49,50]. The NLTK libraries implement the NLP functionality and is accessed as a web service from the web interface using Python2.7 [40]. The semantic concepts are processed using Gephi's headless API that is also accessed as a web-service using Java interface [42]. The rest of the SNAP's software is implemented in Java that includes the frequency filtering, semantic flow processing, and file I/O. In case the resulting semantic network graph is too large and cannot be visualized using the in-browser applet, the data structures can be exported into a visualization using an external partiview application [51]. For the exact list of packages, binaries, and libraries, please see the project portal [5].

Figure 1 not only illustrates the document processing work-flow, it also closely parallels the design and implementation of the modular work-flow as the framework's software components. The document processing parameters for each software component are stored and queried from the project configuration before each processing step and are stored in the user's profile in the database. As the software components are stand-alone and do not rely on each other, the overview figure also captures the software component design schema.

## 4. Sample Corpus Analysis

We tested SNAP's ability to discover and track semantic concepts in three different corpora of text: a well-known and understood novel that contains the main plot and supporting sub-plots, a committee hearings with only a broadly defined focus, and a news media reports on a wide range of topics for a large geographic region. Each corpus illustrates differently structured text with different semantic concept flows. As each corpus can be analyzed in light of many hypotheses, we will not provide a detailed semantic concept flow analysis of each corpus beyond the annotation of the general observations of semantic flow structures.

The interactive and semantic concept rich exploration of a text corpus is the main benefit of using the SNAP framework over the existing language processing tools that often (1) implement only one step of SNAP's framework, (2) rely on a manual text coding to identify topics of interest to be explored, (3) lack the interactive visualization layer, and (4) provide limited and pre-defined semantic concept exploration. SNAP allows a user to concurrently explore multiple hypotheses in the text or freely explore the final 3D network of the semantic flows for 'unexpected' synapses. This benefit is best illustrated by exploring the second corpus of a committee hearings (please see Section 4.2 for more details).

The figures in this section show the semantic concept of an individual document on a x/y plane, while the temporally adjacent documents are stacked on the z-axis. For the visualization clarity the result figures in the top and bottom right panes do not show the inter- nor intra- document edges and only the 3D semantic concept flow. A detail view these edges in a single document is provided in the bottom left pane of the figures.

### *4.1. Moby Dick*

Figure 2 shows the network of semantic flows for the 1851 novel *Moby Dick* by Herman Melville [52]. To engineer the time-stamped documents for analysis, we split the novel's chapters such that each

document contains exactly one chapter and the chapter's number corresponds to a day offset in time. The resulting corpus had 135 documents with an average word document count of 1554 words.

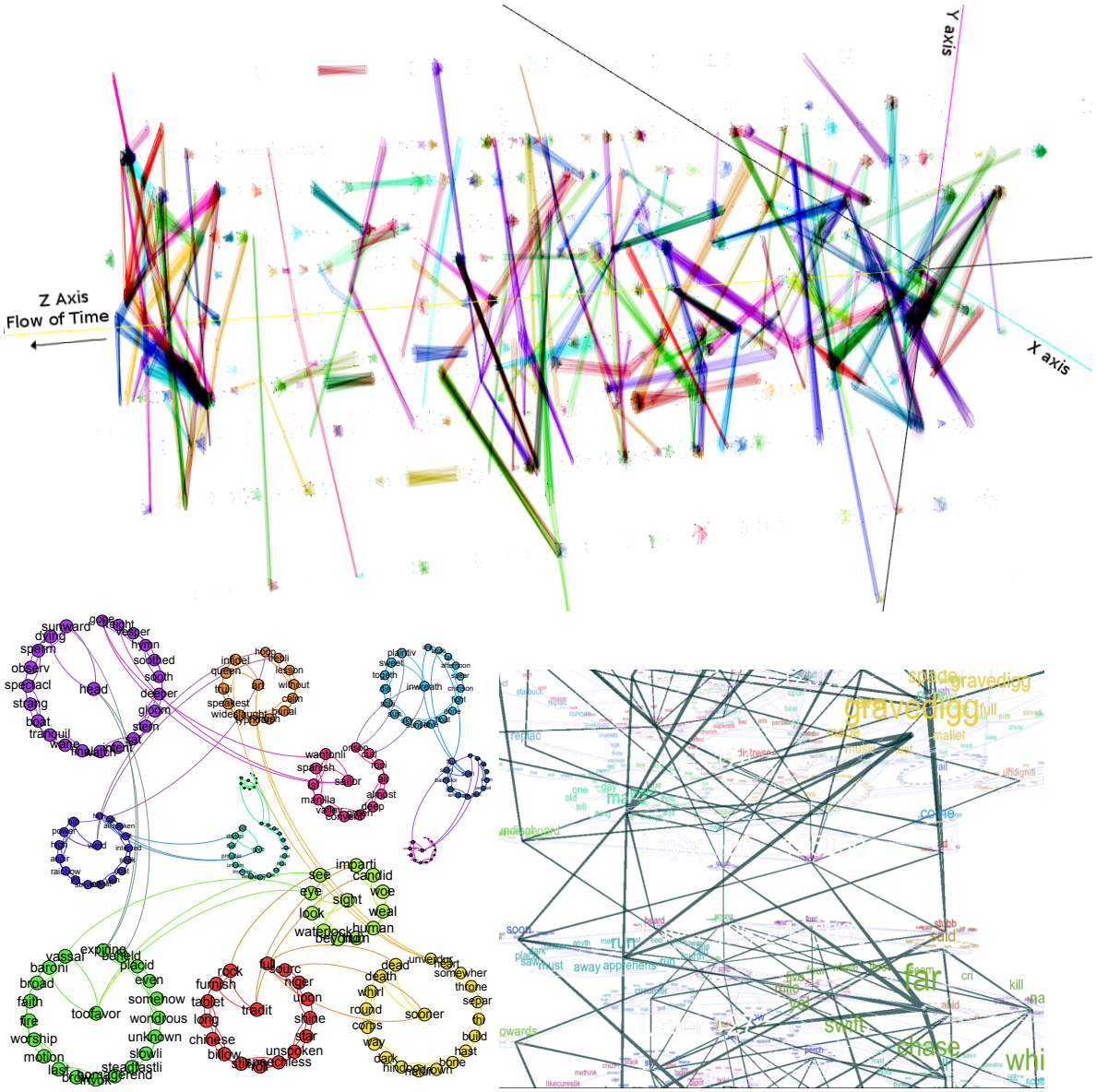


**Figure 2.** SNAP's output visualization of the 1851 novel Moby Dick by Herman Melville. Top pane: A global view of the novel's semantic concepts with the 88.0 percentile of high frequency and the 5 percentile low frequency tokens filtered out. The circles in x/y plane are the semantic concepts delineated by the SNA. The inter- and intra- edges within each document are hidden for visualization clarity. The lines flowing in the z-axis are the semantic concept flow meshes that track the dynamics of the semantic concepts in space and time. Bottom left pane: A semantic network of a single chapter "The Dying Whale". The node with highest Eigenvector Centrality value is in the center of each semantic concept. Bottom right pane: An alternative in-browser visualization showing a closeup of the novel's chapters 123 "The Ship"–128 "The Tail". Please note the view's temporal z-axis is oriented vertically with the chapters in the ascending order.

SNAP's output shows cohesive, well-interconnected semantic concept flows which summarize the novel's main plot on the right hand side of the figure. Building of a plot and the conflict is visible at the very beginning of the semantic flow with the main semantic concepts *savage*, *sail*, and *miles* outlined. The novel's conclusion is visible at the end of the semantic flows, as a convergence of the

flows to the final semantic concepts of *heavens*, *mate*, *souls*, *helmsman*, and *harpoon*. The supporting subplots can be seen as disconnected concepts with short duration e.g., *Constantinople* and *ladies*.

As a literary piece, the text corpus cannot be characterized by a list of semantic concepts that would describe the novel. Instead, we validated SNAP's ability to identify the main semantic concepts qualitatively by comparing chapter's semantic concepts and flows using SNAP's outputs against the novel's plot. We observed strong correlation between the detected semantic concepts and flows and the novel's plot. For example, chapters 125–127 set the stage for the novel's conclusion and the topics of death and re-birth are renewed, among other topics. The SNAP framework identifies common semantic concept flows that support this plot-line as *distress* → *gravedigg* → *soundingboard*. An example of a semantic concept that supports this theme is *far* that contains supporting tokens *white*, *head*, *kill*, *whale*, *chase*, *swift*. The concept flow's dynamics also support the storyline by splitting *far* into the subsequent semantic concepts of *doubt*, *transform* and *endure*.

As far as the mis-identified concepts, the resulting 3D network of semantic flows does contain semantic concepts and flows with low information content that were included by the token frequency filtering having low precision to identify the undesired tokens for removal. Although the semantic concepts such as *let* and *make* remain in the 3D semantic flow network, the supporting tokens of each concept do carry content meaning such as *let* → *instant*, *forward*, *stand*, *alone*, *air* and *make* → *one*, *old*, *day*, *tell*, *aye* respectively.

### *4.2. U.S. Senate Hearings*

Figure 3 shows the analysis of the US Senate Hearings Committee on Environment and Public Works [53]. The corpus has 73 documents with an average of 15,605 tokens each. Each document is a verbatim transcript of a senate hearing session and the time stamp corresponds to the day the hearing. The corpus analysis of traditional term-frequency-based tools would be difficult because the corpus includes a variety of presentation styles, the documents use inconsistent lexical sets, and the legislative session covered a variety of topics of discussion.

The transcript documents contain a variety of presentation formats from a dialogue between two or more U.S. Senate representatives, a testimonial from a single speaker, a piece of evidence submitted to the committee, and errata documents, to name a few. Unlike a single piece of writing that assumes a consistent lexical set used throughout the document, the hearing transcripts consist of sources from various individuals, and the consistency of lexical set used throughout the document cannot be assumed with the exception of the key topic tokens. For example one of the topics of discussion was the economic and ecological impact on Alaska's salmon fisheries in the context of oil production. The dialogue interchangeably used *fish*, *salmon*, and *sockeye* words to reference the same noun for the fish species of concern that should be represented by a single token. As a result, the analysis of the 2015–2016 U.S. Senate Hearings on Environment and Public Works meeting transcripts shows a much larger range of semantic concepts than the analysis of the Moby Dick novel. The connectivity and temporal dynamics of the semantic flows is much richer with inter- and intra-edges and numerous local semantic flows that propagate concepts across a small number of adjacent documents.

Unlike the Moby Dick novel, the US Senate Hearings do not have just one common topic or a plot, instead the hearings discuss a large number of semantic concepts that fall into the *environment* and *public works* topics, how they relate to each other, and which topics are carried over the legislative holidays. The topic catalysts, where multiple topics merge only to break apart in the subsequent document include *financial*, *environment*, *congress*, *ResourcesDistricts*, *pollutant* semantic concepts. The semantic concepts surrounding the catalysts are contents and issue-based with associate flows such as *oil*, *resources*, *dangerous*, *mineral*. The semantic flows also include semantic concepts expressing sentiment and connotation which include *doubtful*, *opportunity* and *easier*.

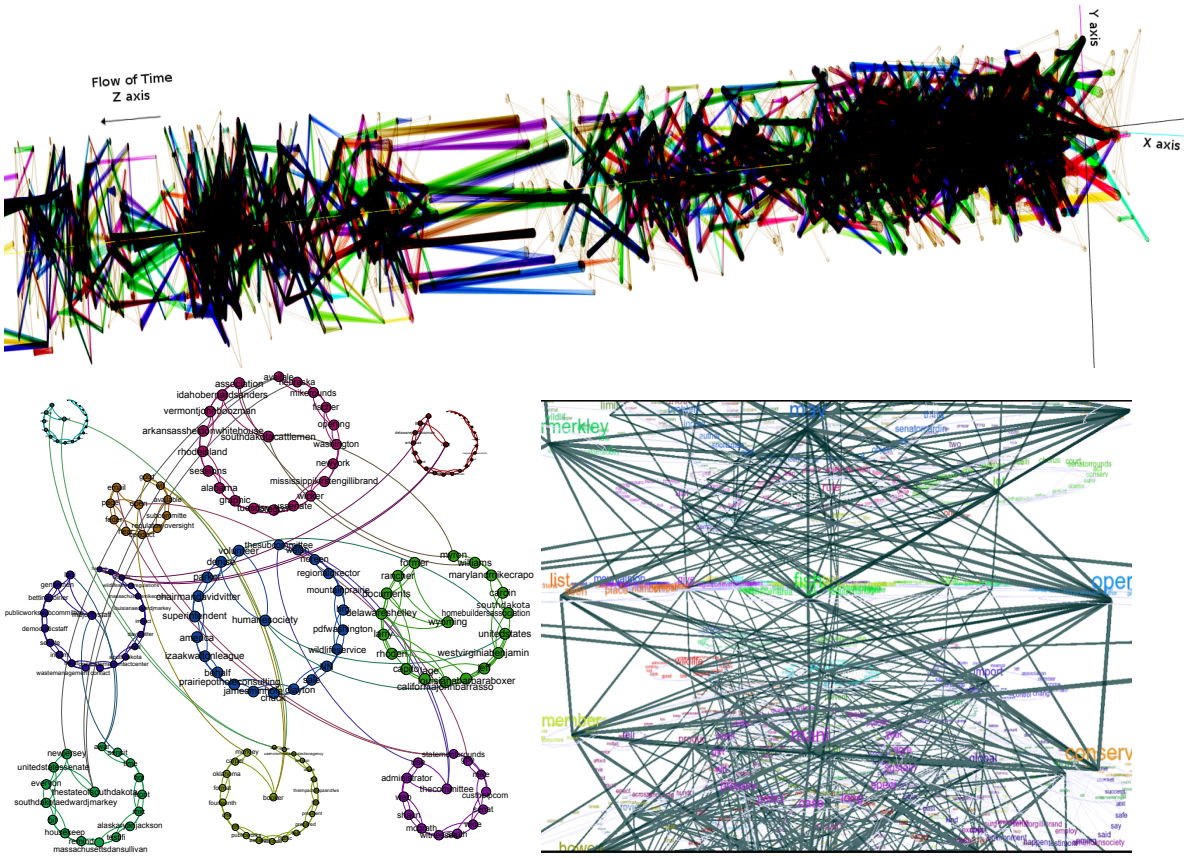


**Figure 3.** SNAP's output visualization of US Senate Hearings Committee on Environment and Public Works corpus for 2016–2017 legislative session. The sub figures in each pane are formatted the same way as in Figure 2. Top pane: The zoomed-out view of the entire corpus. Bottom left pane: A transcript from the 30 August 2016 of the U.S. Senate hearings. Bottom right pane: The in browser closeup of semantic concept flows in the hearing transcripts between 6 May 2015 and 11 May 2015.

In a true exploratory fashion, we were interested in the areas of concerns and topics associated with fisheries. The two main semantic concepts that merged into the *fish* semantic concept were *exempt* with associated tokens *senatorbooker*, *private*, *build*, *continue* and *senatormerkely* with associated tokens *time*, *use*, *public*, *water*. Although the preceding semantic concept merged into the semantic concept *fish* with supporting tokens *streamprotectionrule*, *health*, *require*, they also split, and were related, to other semantic concepts that appeared and were inter-connected to the *fish* semantic concept, namely *protect* with associated tokens *stream*, *mine*, *water*, *coal*, *mountaintop* with associated tokens *senatorcapito*, *environ*, *response*, *surfac*, *pollut*, *meeet*, *develop*, *industristandard* and *last* with associated tokens *senatormarkely*, *rulemake*, *last*, *alaska*, *study*, *compani*, *condit*, *establish*. The semantic concepts *fish*, *protect*, *mountaintop* and *last* fell into the following semantic concepts in the subsequent document: *propos*, *public*, *wildlife*, *right*, *conserv*, and *import*.

The observations above confirm SNAP's ability to "sieve" the documents in the corpus of interest and distill the main concepts and their relations to the preceding and subsequent documents. Furthermore, the corpus analysis was done in a fraction of time it would take to manually code the documents. Some mis-identified (false positive) semantic concepts did occur in the resulting 3D semantic network as artifacts of the raw documents formatted as online (.html) documents with the markup tags still present, legal disclaimers and headers associated with the official government documents, or references to external sources. These concepts include: *web*, *pdf*, *fax*, *day*, *sector*.

### *4.3. Australian Broadcast Commission*

The corpus of the Australian Broadcasting Commission's (ABC) news documents from science and rural sections has 765 documents averaging 473 words per document [54]. Figure 4 shows SNAP's analysis of the corpus with a relatively small number of tokens in each temporal layer (document) and the large number of unique general topics across the entire corpus resulting in a sparse semantic flow connectivity.

Unlike previously analyzed corpora of the Moby Dick novel or the U.S. Senate Hearings, the ABC results illustrate that each news report is discrete, stand alone and only connected to the temporally adjacent news briefs with global semantic concepts that include *environment*, *data*, *health*, *research*, *ocean*. These corpus characteristics are confirmed by the results that show a lack of semantic concept flows with a minimal connectivity between adjacent documents since the corpus mixes two news feeds (the science topic and the rural news articles) as well as the news briefs explicitly assigned a time-stamp which staggers the semantic concepts away from temporally adjacent documents resulting in a breaking the flows of semantic concepts. The SNAP's historical semantic concept analysis confirms the lack of common semantic threads that flow through the corpus.

Figure 4 bottom left pane shows the semantic network analysis of a single document. Although the overall corpus shows the lack of semantic concept flows, this sub-figure shows that each short news document is well written with cohesive semantic concepts, showed as 12 groups of the same color tokens which are well inter- and intra-connected by the lexical chaining edges.

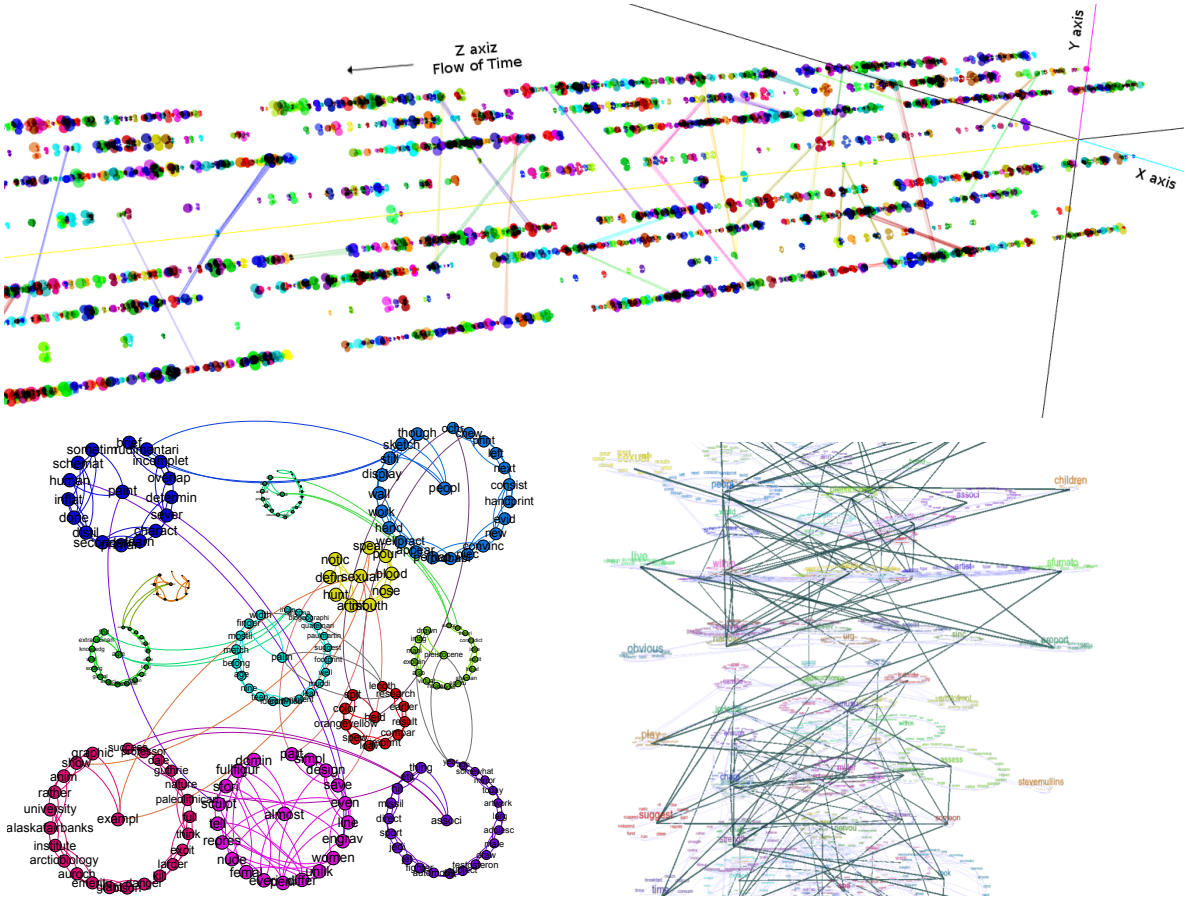


**Figure 4.** SNAP's output visualization for Australian Broadcasting Commission's corpus on rural science articles. The sub figures in each pane are formatted the same way as in Figure 2. Top pane: The zoomed-out view of the entire corpus. Bottom left pane: A semantic network analysis of the news article from October 20th, 2015. Bottom right pane: A close-up view of news articles from 20 October 2015 to 27 October 2015.

## 5. Discussion and Conclusions

We constructed an interactive text mining framework for historical semantic concept exploration that allows much richer text analysis well beyond TF/IDF or identification of the key words by manual coding. The modular framework relies on mature linguistic tools that can be easily swapped to customize the mechanics of the computational linguistics processing. One such customization might include the implementation of a workflow to analyze the sentiment concept flows, where a sentiment concept flow would track and connect tokens coded with a sentiment label. SNAP framework does not require any knowledge of programming, and is easily accessible through web-services. While taking advantage of computational linguistic tools, graph theory analysis, and immerse 3D visualization.

Although we presented an analysis of the semantic concept flows by listing discrete concepts, the interactive exploration of the semantic flow dynamics can be used to analyze the corpus in detail. As currently implemented, SNAP and its modules can be used for corpus summarization, content analysis, or as a writer's aid. To support a writer, SNAP can be used to check if all concepts flow through the corpus without interruption (semantic concept flow continuity), and are included in the document's summary (merger of multiple concepts into a single closing section). New concepts are only introduced in support of the main theses in a multi-section document (a concept flow splitting). Swapping or adding different linguistic modules can augment the work-flow for sentiment tracing, influence tagging, concept disambiguation, and actor identification to name a few.

We used SNAP to analyze three different corpora that ranged in size, construction and structure. The analysis of the Moby Dick novel shows the framework's ability to explore a corpus with a strong story-line with supporting sub-plots and an examination of the plot dynamics. The analysis of the U.S. Senate hearings corpus identified the key semantic concepts that unite diverse semantic concepts as well as the role of supporting semantic concepts that cause the subsequent breakup of a semantic flow. Finally, the ability to identify a corpus without historical semantic flows is illustrated using the news briefs from different topics and a time-stamp that disrupts the flow of semantic concepts.

**Author Contributions:** Project conceptualization, M.C.; methodology, all authors; software, all authors; validation, all authors; formalism, M.C.; data curation, M.C., E.P., L.O.; writing–original draft preparation, M.C., E.P., L.O., A.M.; writing–review and editing, M.C.; visualization, all authors; supervision, M.C.; funding acquisition, M.C.

**Funding:** This project was in part funded by Alaska EPSCoR NSF award #OIA-1208927.

**Acknowledgments:** The authors would also like to thank all project collaborators and Mackenzie Bartlett, Ben Gurganious, and Neal Logan research assistants.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Smith, A.E. Automatic extraction of semantic networks from text using Leximancer. In *Companion Volume of the Proceedings of HLT-NAACL 2003-Demonstrations*; ACL: Edmonton, AB, Canada, 2003.
2. Sowa, J.F. *Principles of Semantic Networks: Explorations in the Representation of knowledge*; Morgan Kaufmann: Burlington, MA, USA, 2014.
3. Donovan, R.E.; Woodland, P.C. A hidden Markov-model-based trainable speech synthesizer. *Comput. Speech Lang.* **1999**, *13*, 223–241. [CrossRef]
4. Nadkarni, P.M.; Ohno-Machado, L.; Chapman, W.W. Natural language processing: An introduction. *J. Am. Med. Inform. Assoc.* **2011**, *18*, 544–551. [CrossRef] [PubMed]
5. Cenek, M. Semantic Network Analysis Project (SNAP), 2006. Available online: https://github.com/mcenek/SNAP (accessed on 10 May 2019).
6. Bird, S. NLTK: The natural language toolkit. In Proceedings of the COLING/ACL on Interactive Presentation Sessions, Sydney, Australia, 17–21 July 2006; Association for Computational Linguistics: Stroudsburg, PA, USA, 2006; pp. 69–72.
7. spaCy-Industrial-Strength Natural Language Processing in Python. Available online: https://spacy.io/ (accessed on 30 June 2018).

8. Stanford Natural Language Processing Group. Available online: https://nlp.stanford.edu/software/ (accessed on 10 May 2019).
9. Manning, C.D.; Surdeanu, M.; Bauer, J.; Finkel, J.R.; Bethard, S.; McClosky, D. The stanford corenlp natural language processing toolkit. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Baltimore, MD, USA, 22–27 June 2014; pp. 55–60.
10. Richens, R.H. Interlingual machine translation. *Comput. J.* **1958**, *1*, 144–147. [CrossRef]
11. Fatima, Q.; Cenek, M.; Cenek, M. New graph-based text summarization method. In Proceedings of the 2015 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM), Victoria, BC, Canada, 24–26 August 2015; pp. 396–401.
12. Jarmasz, M.; Szpakowicz, S. Not as easy as it seems: Automating the construction of lexical chains using roget's thesaurus. In Proceedings of the Conference of the Canadian Society for Computational Studies of Intelligence, Halifax, NS, Canada, 11–13 June 2003; pp. 544–549.
13. Patel, S.M.; Dabhi, V.K.; Prajapati, H.B. Extractive Based Automatic Text Summarization. *JCP* **2017**, *12*, 550–563. [CrossRef]
14. Singhal, A. *Introducing the Knowledge Graph: Things, Not Strings*; Official Google Blog. Available online: www.blog.google (accessed on 4 December 2019).
15. Miller, G.A. WordNet: A lexical database for English. *Commun. ACM* **1995**, *38*, 39–41. [CrossRef]
16. Fellbaum, C. WordNet. In *Theory and Applications of Ontology: Computer Applications*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 231–243.
17. Vossen, P. *A multilingual Database with Lexical Semantic Networks*; Springer: Berlin/Heidelberg, Germany, 1998.
18. Tur, G.; Celikyilmaz, A.; He, X.; Hakkani-T'ur, D.; Deng, L. Deep Learning in Conversational Language Understanding. In *Deep Learning in Natural Language Processing*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 23–48.
19. Young, T.; Hazarika, D.; Poria, S.; Cambria, E. Recent trends in deep learning based natural language processing. *IEEE Comput. Intell. Mag.* **2018**, *13*, 55–75. [CrossRef]
20. Zheng, R.; Chen, J.; Qiu, X. Same representation, different attentions: Shareable sentence representation learning from multiple tasks. *arXiv* **2018**, arXiv:1804.08139.
21. Barzilay, R.; Elhadad, M. Using lexical chains for text summarization. In *Advances in Automatic Text Summarization*; MIT Press: Cambridge, MA, USA, 1999; pp. 111–121.
22. Barzilay, R. Lexical Chains for Summarization. Ph.D. Thesis, Ben-Gurion University of the Negev, Beersheba, Israel, 1997.
23. Galley, M.; McKeown, K. Improving word sense disambiguation in lexical chaining. In Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, 9–15 August 2003; Volume 3, pp. 1486–1488.
24. Dang, J.; Kalender, M.; Toklu, C.; Hampel, K. Semantic Search Tool for Document Tagging, Indexing and Search. U.S. Patent 9,684,683, 20 June 2017.
25. Steyvers, M.; Tenenbaum, J.B. The Large-Scale Structure of Semantic Networks: Statistical Analyses and a Model of Semantic Growth. *Cogn. Sci.* **2005**, *29*, 41–78._3. [CrossRef] [PubMed]
26. Ensan, F.; Bagheri, E. Document Retrieval Model Through Semantic Linking. In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, Cambridge, UK, 6–10 February 2017; pp. 181–190.
27. Navigli, R. Consistent validation of manual and automatic sense annotations with the aid of semantic graphs. *Comput. Linguist.* **2006**, *32*, 273–281. [CrossRef]
28. Overview Project: Completed News Stories, 2017. Available online: https://github.com/overview/overviewserver/wiki/News-stories (accessed on 10 May 2019).
29. Document Cloud: Analyze, Annotate, Publish. Turn Documents into Data. 2017. Available online: https://www.documentcloud.org/ (accessed on 10 May 2019).
30. Apache UIMA—Apache UIMA. Available online: http://incubator.apache.org/uima/ (accessed on 10 May 2019).
31. IBM Watson: AlchemyAPI. Available online: https://www.ibm.com/watson/alchemy-api.html (accessed on 10 May 2019).

32. Newman, D.; Noh, Y.; Talley, E.; Karimi, S.; Baldwin, T. Evaluating topic models for digital libraries. In Proceedings of the 10th Annual Joint Conference on Digital Libraries, Queensland, Australia, 21–25 June 2010; pp. 215–224.
33. Suen, C.; Huang, S.; Eksombatchai, C.; Sosic, R.; Leskovec, J. Nifty: A system for large scale information flow tracking and clustering. In Proceedings of the 22nd International Conference on World Wide Web, Rio de Janeiro, Brazil, 13–17 May 2013; pp. 1237–1248.
34. Dou, W.; Yu, L.; Wang, X.; Ma, Z.; Ribarsky, W. Hierarchicaltopics: Visually exploring large text collections using topic hierarchies. *IEEE Trans. Vis. Comput. Graph.* **2013**, *19*, 2002–2011. [PubMed]
35. Chaney, A.J.B.; Blei, D.M. Visualizing Topic Models. In Proceedings of the Sixth International AAAI Conference on Weblogs and Social Media, Dublin, Ireland, 4–7 June 2012.
36. DiMaggio, P.; Nag, M.; Blei, D. Exploiting affinities between topic modeling and the sociological perspective on culture: Application to newspaper coverage of US government arts funding. *Poetics* **2013**, *41*, 570–606. [CrossRef]
37. Cui, W.; Liu, S.; Tan, L.; Shi, C.; Song, Y.; Gao, Z.; Qu, H.; Tong, X. Textflow: Towards better understanding of evolving topics in text. *IEEE Trans. Vis. Comput. Graph.* **2011**, *17*, 2412–2421. [CrossRef] [PubMed]
38. Chuang, J.; Ramage, D.; Manning, C.; Heer, J. Interpretation and trust: Designing model-driven visualizations for text analysis. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Austin, TX, USA, 5–10 May 2012; pp. 443–452.
39. Altaweel, M.R.; Alessa, L.N.; Kliskey, A.D.; Bone, C.E. Monitoring land use: Capturing change through an information fusion approach. *Sustainability* **2010**, *2*, 1182–1203. [CrossRef]
40. Bird, S.; Klein, E.; Loper, E. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2009.
41. Bastian, M.; Heymann, S.; Jacomy, M. Gephi: An Open Source Software for Exploring and Manipulating Networks. In Proceedings of the Third International AAAI Conference on Weblogs and Social Media, San Jose, CA, USA, 17–20 May 2009.
42. Gephi—The Open Graph. Available online: http://gephi.org (accessed on 30 June 2017).
43. Blondel, V.D.; Guillaume, J.L.; Lambiotte, R.; Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**, *2008*, P10008. [CrossRef]
44. Lambiotte, R.; Delvenne, J.C.; Barahona, M. Laplacian dynamics and multiscale modular structure in networks. *arXiv* **2008**, arXiv:0812.1770.
45. Ruhnau, B. Eigenvector-centrality—A node-centrality? *Soc. Netw.* **2000**, *22*, 357–365. [CrossRef]
46. Brandes, U. A faster algorithm for betweenness centrality. *J. Math. Sociol.* **2001**, *25*, 163–177. [CrossRef]
47. Abbott, B. The Digital Universe Guide for Partiview, 2006. Available online: http://haydenplanetarium.org/universe/duguide (accessed on 10 May 2019).
48. WAMP, LAMP and MAMP Stacks: Softwaculous AAMPS, 2019. Available online: http://www.ampps.com/ (accessed on 1 June 2019).
49. CodeIgniter Web Framework, 2019. Available online: https://www.codeigniter.com/ (accessed on 1 June 2019).
50. 3D JavaScript Libraries, 2019. Available online: https://threejs.org (accessed on 1 June 2019).
51. Partiview. 2019. Available online: http://virdir.ncsa.illinois.edu/partiview/ (accessed on 1 June 2019).
52. Melville, H. *Moby-Dick*; Courier Corporation: Garden City, NY, USA, 2003.
53. U.S. Government Publishing Office, W.D. U.S. Senate, Committee on Environment and Public Works, 2016. Available online: http://www.gpo.gov/fdsys (accessed on 10 May 2019).
54. Australian Broadcasting Commission 2006, 2006. Available online: https://github.com/nltk (accessed on 10 May 2019).