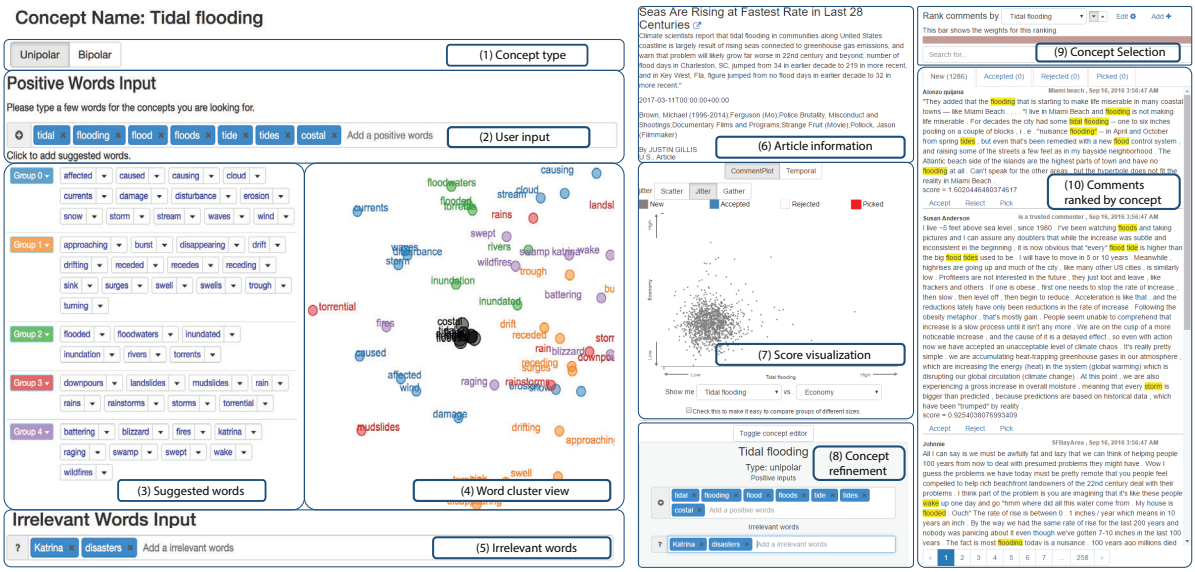
ConceptVector: Text Visual Analytics via Interactive Lexicon Building using Word Embedding

Deokgun Park, Seungyeon Kim, Jurim Lee, Jaegul Choo,
Nicholas Diakopoulos, and Niklas Elmqvist, *Senior Member, IEEE*



(a) Concept-Building View (b) Document Analysis View

Fig. 1. ConceptVector supports interactive construction of lexicon-based concepts. Here the user creates a new unipolar concept (1) by adding initial keywords related to 'tidal flooding' (2). The system recommends related words along with their semantic groupings (3), also shown in a scatterplot (4), revealing word- and cluster-level relationships. Irrelevant words can be specified to improve recommendation quality (5). Concepts (9) can then be used to rank document corpora (10). Document scores can be visualized in a scatterplot based on concepts such as 'tidal flooding' and 'money' (7). Users can further refine concepts based on results (8).

Abstract—Central to many text analysis methods is the notion of a *concept*: a set of semantically related keywords characterizing a specific object, phenomenon, or theme. Advances in word embedding allow building a concept from a small set of seed terms. However, naive application of such techniques may result in false positive errors because of the polysemy of natural language. To mitigate this problem, we present a visual analytics system called ConceptVector that guides a user in building such concepts and then using them to analyze documents. Document-analysis case studies with real-world datasets demonstrate the fine-grained analysis provided by ConceptVector. To support the elaborate modeling of concepts, we introduce a bipolar concept model and support for specifying irrelevant words. We validate the interactive lexicon building interface by a user study and expert reviews. Quantitative evaluation shows that the bipolar lexicon generated with our methods is comparable to human-generated ones.

Index Terms—Text analytics, visual analytics, word embedding, text summarization, text classification, concepts

1 Introduction

We live in a world that routinely produces more textual data on a daily basis than can be comfortably viewed—let alone analyzed—by a single person in virtually any given domain: finance, journalism, medicine, politics, and business, to name just a few. As a result, automatic text analysis methods, such as sentiment analysis [34], document summarization [4], and probabilistic topic modeling [3] are becoming increasingly important. Central in most of these methods is the focus on textual *concepts*, defined as a set of semantically related keywords describing a particular object, phenomenon, or theme. For example, sentiment analysis can be viewed as analyzing documents according to two concepts: positive and negative sentiment. Similarly, the topics derived in topic modeling can be thought of as document-driven concepts. The benefit of this unified view is that concepts, once created, can then be shared and reused many times, similarly to widely applicable lexicon sets such as Linguistic Inquiry and Word Count (LIWC) [37] or General Inquirer (GI) [39].

Generally, building a lexicon for a particular concept requires significant human effort, and thus only a limited number of human-generated concepts have been available, usually with a small number of keywords

- *D. Park and N. Elmqvist are with University of Maryland in College Park, MD, USA. E-mail: {intuinno, elm}@umd.edu.*
- *S. Kim is with Google Inc. in Mountain View, CA, USA. E-mail: seungyeonk@google.com.*
- *J. Lee and J. Choo, the corresponding author, are with Korea University in Seoul, Republic of Korea. E-mail: {jurim0301, jchoo}@korea.ac.kr.*
- *N. Diakopoulos is with Northwestern University in Evanston, IL, USA. E-mail: nicholas.diakopoulos@gmail.com.*

Manuscript received 31 Mar. 2017; accepted 1 Aug. 2017.
Date of publication 28 Aug. 2017; date of current version 1 Oct. 2017.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TVCG.2017.2744478

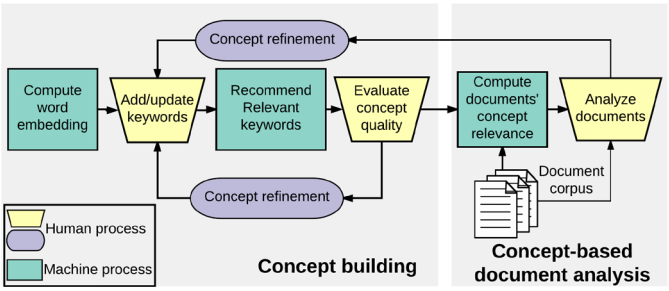


Fig. 2. Workflow of ConceptVector, involving human- and machine-side tasks in a collaborative manner. See Section 5 for details.

contained in each. Recently, Fast et al. [14] proposed a technique called Empath that uses state-of-the-art word embedding [31] to efficiently build a semantically meaningful lexicon for a concept. Given user-provided keywords, such as 'bleed' and 'punch,' Empath automatically generates semantically related keywords (e.g., 'violence'). This enables user-driven document analysis from diverse aspects. For example, they found that deceptive languages in fake reviews tend to use stronger and exaggerated words while real reviews often use spatial words to describe their experiences in concrete detail.

However, we claim that without considering the document context and keyword usage patterns in it, blindly applying a pre-built lexicon for document analysis can easily lead to a misunderstanding of the content. For instance, when we compared Twitter messages from the U.S. 2016 presidential candidates by using a built-in lexicon provided by Empath, we found that Donald Trump used twice as many keywords in an 'alcohol'-related lexicon than the other candidate. A close inspection of the usage pattern of this lexicon revealed the single word *lightweight* to be a dominant keyword. *Lightweight* is colloquially used for a person who cannot withstand an alcoholic drink, and hence had found its way into the lexicon for 'alcohol.' However, Trump used this keyword to mock people as less influential or important; therefore, in this corpus, this keyword is not related to the concept 'alcohol.' This example shows the difficulty in applying a lexicon to document analysis in a custom domain because of different usages of keywords in their context.

Motivated by this challenge, we present a visual analytics system called CONCEPTVECTOR[1] that seamlessly integrates a user-driven lexicon-building process with customized document analysis in a highly efficient and flexible manner. As shown in Figure 1, users can easily create a lexicon for a particular concept by selecting system-recommended keywords and by adding new keywords of their choice. A user can also explicitly tag recommended words that are unrelated to the concept under consideration as irrelevant, clarifying the meaning by weakening the overall relevance of those words. In addition, ConceptVector supports the definition and construction of bipolar concepts (e.g., positive vs. negative sentiments, liberal vs. conservative political orientation, and Trekkie vs. Star Wars fans) that can be modeled by providing two sets of seed words corresponding to the two polarities. ConceptVector also allows users to analyze any document corpus with respect to any desired concept, such as product reviews based on sentiment, blog posts based on political orientation, or trade articles based on business sectors. As shown in Figure 2, the document corpus analysis process is tightly integrated with the concept-building process described above so that users can customize concepts during document analysis.

Our quantitative evaluation validates the proposed bipolar concept-building model by comparing automatically generated rankings with a small number of seed words to the human-labeled rankings of words associated with the concept 'happiness' [11]. We also present a user study to evaluate the interactive concept-building process, where we compared the performance of the lexicon-building process against using an online thesaurus (`Thesaurus.com`) and the WordNet [33] lexical database. We also provide usage scenarios demonstrating the concept-based document analysis process.

[1]http://www.conceptvector.org/

In summary, the contributions of our work include the following:

- A visual analytics system called CONCEPTVECTOR where users can interactively build and refine a lexicon for custom concepts and analyze a document corpus using them in a seamless manner;
- Models for user-steerable word-to-concept similarities to handle irrelevant keywords as well as bipolar concepts; and
- Quantitative results comparing the capabilities of our word-to-concept similarities to human-labeled ones; and
- Results from a user study comparing concept generation performance using ConceptVector to Thesaurus.com and WordNet.

2 RELATED WORK

Numerous previous studies have attempted to scale up human capability to make sense of a text corpora. ConceptVector is a visual analytics system that uses word-level semantics using a lexicon for concepts. In this section, we discuss current research related to our work from three perspectives: (1) manual approaches for constructing word relationships and hierarchies, (2) automatic word-embedding approaches, and (3) visual analytics approaches for word-level content analysis.

2.1 Building Word Relationships and Hierarchies

Manually building a lexicon with coherent semantics has long been an active area of research. LIWC [37] is an example of a manually built lexicon that characterizes various concepts. The General Inquirer[2] is a comparable line of research that builds lexica in diverse concepts. Beyond building a lexicon for a particular purpose, researchers have also developed sophisticated structures that store relationships and hierarchies of words.

Unlike these methods, which rely on a small number of experts to compose a lexicon, the Hedonometer project [11] employed crowd-sourcing to build a lexicon for sentiment ranking. One benefit of this approach is its large-sized lexicon, containing the ranked list of 7,000 words in terms of the degree of happiness.

Although these manually built databases, which store relationships and hierarchies of words, provide high-quality information for various natural language understanding and text analysis tasks, the main problem is the significant human effort needed to create and validate them. This makes it difficult for users to efficiently create a lexicon for their own purpose. Because of this high cost, only a limited number of widely applicable concepts can be built, and building a domain-specific custom lexicon has not been well-supported. This has motivated a slew of automatic methods to craft a lexicon of custom concepts.

2.2 Word Embedding

Word embedding computes semantically meaningful vector representations of words in a high-dimensional space. Compared to traditional methods of representing a word as a vector, such as the bag-of-words representation [29] or latent semantic indexing [9], recent word embedding methods such as word2vec [31] and GloVe [38] have two noteworthy advantages in terms of high-level semantics: meaningful nearest neighbors and linear substructures [38]. Regarding the first, these techniques satisfactorily capture semantically related words as the nearest neighbors of a particular word in a vector space. As for linear substructures, the vector obtained by subtracting two words in a vector space often yields semantics that contrast the words. For instance, if we subtract a word vector 'queen' from 'king' and then add 'girl,' the resulting vector corresponds to 'boy.' This stems from the fact that the vector from 'king' to 'queen' and from 'boy' to 'girl' are similar, commonly representing the notion of gender (from male to female).

Since such word embedding techniques have shown their advantages in numerous tasks in natural language processing and information retrieval, advanced word embedding techniques have recently been actively studied. Ling et al. proposed the use of multidimensional transformation matrices to flexibly capture different semantics of a single

[2]http://www.wjh.harvard.edu/ inquirer/

word [27] leading to better representations for part-of-speech tagging tasks. Similarly, assigning more weight to a particular word than other words in a sentence produced better word embeddings by extending the continuous bag-of-words model [28]. The weights are computed by an attention model, yielding better performance than neural network models [1]. Tian et al. integrated an expectation-maximization (EM) algorithm with the continuous skip-gram model to handle the polysemy problem [42]. For example, the word 'bank' can have multiple vector representations corresponding to 'a place related to money' and 'a place where water runs,' respectively. Besides transforming word-level embeddings, several efforts extended this technique to document-level embeddings that yielded good performance in information retrieval tasks [20, 24]. Other notable recent studies applied the technique to machine translation [30, 32]. Additionally, the skip-gram idea of word2vec has been applied in generating the embeddings of entities in other domains, e.g., bibliographic items in scientific literature [2] and nodes in network analysis [15]. Finally, and most relevant to this work, Fast et al. [14] showed that word embedding can be used to expedite lexicon-building so that users can easily create their own concepts.

2.3 Word-Level Content Analysis

The use of a coherent set of keywords for characterizing a particular concept has wide applicability in various document analysis tasks. For instance, the problem of sentiment analysis has been tackled by identifying a set of keywords expressing the positive (or the negative) sentiment, possibly with different degree values, and this is also known as a lexicon-based sentiment analysis [34, 40]. In topic modeling, such as latent Dirichlet allocation (LDA) [3], a topic represents a set of semantically related keywords found in a document corpus, e.g., sports- or science-related topics, generated from a large amount of news articles. Recent studies by Kim et. al [19, 21] are particularly notable because they introduced a continuous embedding space similar to *concepts* as considered in this paper, although they only covered emotion-related concepts.

Topic modeling has also been actively employed in visual analytics approaches for document analysis. TIARA [44] is one of the first systems that integrated LDA with interactive visualization. This system visualizes the topical changes of documents over time in a streamgraph view reminiscent of ThemeRiver [16]. Other studies, such as ParallelTopics [12] and TextFlow [8], also focused on visualizing topical changes over time in document data by using different visualization techniques, such as parallel coordinates and custom glyphs, respectively. In most of these studies, the key information for understanding the visualized topics is a set of dominant keywords associated with each topic. However, the number of topics can be as large as several hundreds or thousands [41]. This makes manual interpretation of topic characterization or topic labeling a main bottleneck for the topic modeling. To facilitate this task, Termite [7] provides an interactive visualization with which a user can explore topics in terms of their dominant keywords, as well as the overlapping patterns of keywords among different topics. In addition, various interactive capabilities that can steer the topic modeling process in a user-driven manner have been studied. iVisClustering [26] allows a user to perform a user-driven topic modeling process by interactively constructing topic hierarchies and changing keyword weights of a topic. Chang et al. introduced a interactive clustering system based on knowledge-graph embeddings [5]. More recently, non-negative matrix factorization [25] has been proposed as an alternative topic modeling method that can flexibly support user needs such as splitting and merging topics, creating a new topic via particular keywords, and supporting user-driven topic discovery [6].

Our ConceptVector work in this study has much in common with topic modeling: both try to summarize documents, and both express words and documents as high-dimensional vectors. However, they differ in whether humans or the document corpus itself drive the latent semantics behind each dimension. Topic modeling, therefore, is better-suited for finding hidden underlying topic clusters, while ConceptVector provides better interpretability and transferability. In this sense, topic modeling and ConceptVector are complementary.

Lexicon-based document analysis has also been applied in various application domains. For instance, Kwon et al. [23] used a manually built lexicon to identify online health community postings that share personal medical experiences. In most of these previous studies, document analysis relied on lexicons of properly chosen words that were created for a specific purpose. The ConceptVector system aims to help users easily create such lexicons.

3 Motivation: Concept-Based Document Analysis

Here we describe two real-world examples where concept-based document analysis was performed by using Empath and Jupyter Notebook.[3] First, we show how concepts can reveal the underlying differences in two document sets, such as tweets from Hillary Clinton and from Donald Trump, highlighting the importance of integrating the lexicon-building process with its refinement during the document analysis. Second, we demonstrate how NASDAQ 100 companies can be clustered using the differences in concepts and how each cluster can be interpreted using tweets mentioning them.

3.1 Tweets by U.S. 2016 Presidential Candidates

Empath [14] provides prebuilt lexica of various concepts that can be used to compare two document groups. Using these 194 prebuilt concepts provided by Empath, we analyzed two sets of tweets composed by Hillary Clinton and Donald Trump[4] respectively, each of which contains about 3,000 tweets. Figure 3(a) shows the top ten categories statistically significantly different from each other ($p < .01$). For example, Trump mentioned more terms in the 'ugliness' (13.9 odds), 'swearing terms' (6.7 odds), and 'surprise' (5.8 odds) categories, whereas Hillary used more in the 'sexual' (4.97 odds), 'eating' (4.6 odds), and 'home' (4.2 odds) categories. Interestingly, Trump used more casual language while Hillary's tweets contained words related to 'anger' and 'disgust.'[5]

However, further examination reveals numerous false positives. Figure 3(b) shows the most dominant keywords corresponding to each concept. While some keywords make sense, e.g., 'wow' in the 'surprise' category, less meaningful words exist in other categories. For example, Trump was shown to talk more about the 'plant' concept because of the term 'bush,' which in fact indicates Jeff Bush. 'crooked' in the 'ugliness' concept means 'deformed', whereas Trump is using it in his catchphrase 'Crooked Hilary' to mean 'not straightforward; dishonest.' Besides, another strong concept 'hipster' emerged because of the use of the term 'looking,' while 'swearing terms' emerged because of the use of 'bad.' In Hillary's case, the 'sexual' concept appeared owing to the use of 'violence,' which did not make much sense. After removing these words from the corresponding concepts, these concepts no longer show significant differences between the two.

3.2 Tweets from NASDAQ 100 Companies

Concepts can be also used to extract meaningful features from documents. Given tweets about NASDAQ 100 companies,[6] our goal in this work was to find meaningful clusters and their distinct characteristics by using concepts as features. That is, for a set of tweets belonging to each company, we obtained its 194-dimensional feature vector by computing the occurrence count of words contained in each of the 194 prebuilt concepts. Afterwards, we performed k-means clustering and 2D embedding via principal component analysis (PCA) [18].

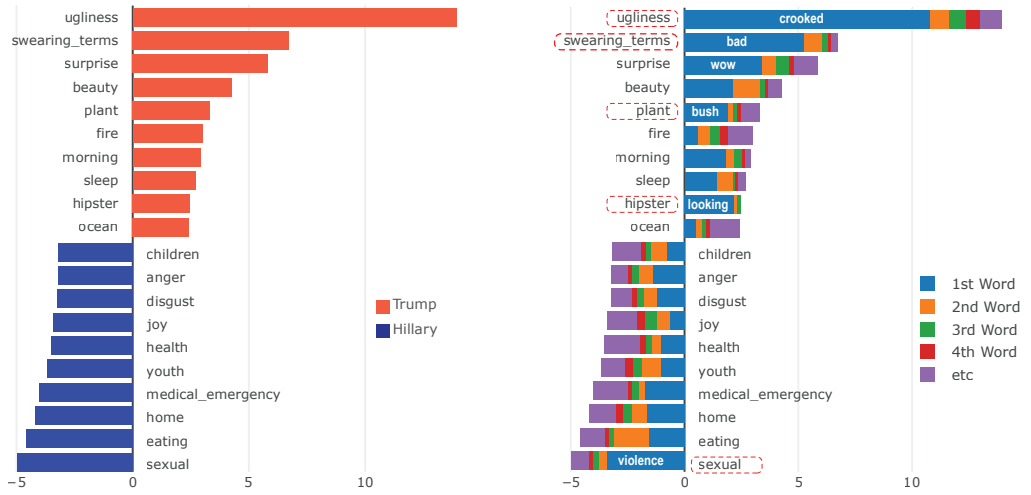
The results (Figure 4(a)) reveal that words from the company name affect the results, e.g., 'cooking' and 'restaurant' categories for Dish Network Corporation. Companies containing 'technology' in their names form a cluster because of similar reasons. After removing these words from the lexicon of the corresponding concept and recomputing feature vectors, the clustering results are shown to be more reasonable (Figure 4(b)). For example, Marriott and TripAdvisor form a single cluster owing to the high frequency of words in 'tourism,' 'warmth,' 'sleep,' and 'vacation' mainly because of the use of the words 'hotel'

[3] http://conceptvector.org/#/twitter

[4] https://www.kaggle.com/benhamner/clinton-trump-tweets

[5] http://graphics.wsj.com/clinton-trump-twitter/

[6] http://www.followthehashtag.com/datasets/nasdaq-100-companies-free-twitter-dataset/



(a) Top 10 categories between Trump and Hillary

(b) Detailed analysis of top words in each category

Fig. 3. Comparison of tweet messages from Hillary Clinton and from Donald Trump during the U.S. 2016 presidential election. The odd ratios of the top 10 categories show differences between the two candidates in (a). The analysis on actual keywords contributing to their corresponding category scores reveals limitations of using the prebuilt lexicon in (b). Red dotted categories do not make sense, because an irrelevant top word is counted dominantly. For example, keywords such as 'bush' in the 'plant' category and 'looking' in the 'hipster' category are not relevant to their categories.

and 'hot.' Companies with their tweets containing negative sentiments such as 'ridicules,' 'neglect,' 'kill,' or 'hate' are clustered together.

This example shows that document analysis using concepts as a feature extractor is useful, but that existing systems such as Empath lack the integrated support for concept construction and refinement, as well as interactive concept-based analysis itself.

4 ConceptVector in Action

To address the limitations of using prebuilt lexica, ConceptVector aims at facilitating user-driven concept building as well as the subsequent concept-based document analysis in a seamless manner.

While the previous examples started with prebuilt lexica, we now present how ConceptVector can be used to build custom concepts in the task of journalistic curation of user comments on online news. Moderation of online comments can follow various approaches, and often includes mechanisms to remove uncivil, profane, or otherwise inflammatory comments. That is, however, not our focus here; instead we consider the approach championed by the New York Times, in which editorially interesting and insightful comments are selected and highlighted on the site as "NYT Picks" comments. Below we present a scenario showing how an expert community moderator from an organization such as the New York Times could leverage the capabilities of ConceptVector to define and deploy those concepts useful for finding and selecting "NYT Picks" comments.

It is helpful to understand the general editorial attitude and approach—the persona—of an online news moderator. Prior research has enumerated several dimensions of editorial interest for finding high-quality comments including factors such as comment relevance, argument quality, novelty, and personal experience [10]. Importantly, different articles or subcommunities on a site demand different approaches to moderation and the application of different editorial criteria [35]. Diversity is a dimension of utmost importance to comment moderators; it is a difficult task to select high-quality comments that also reflect the diversity of voices available in a comment stream. ConceptVector is well-suited to enabling such diverse selection because of its capabilities to allow moderators to develop content-specific or even article-specific concepts to apply to different contexts, and to see how comments are scored when applying that concept.

Let us follow Laurie, a hypothetical comment moderator at the New York Times who is trying to moderate comments on several different articles. Her task is to pinpoint diverse but representative comments to highlight on the site as "NYT Picks."

The article she is examining is entitled "Seas Are Rising at Fastest Rate in Last 28 Centuries," which has over 1,200 comments when she logs on.[7] She is really not looking forward to moderating the comments for this article, because an article like this always brings out the global warming skeptics who can cause quite a ruckus. The article is specifically about the idea of 'tidal flooding,' i.e., the notion that coastal areas will be flooded more often as sea levels rise. Using ConceptVector, she first wants to develop a tightly defined concept on this specific idea of 'tidal flooding' so that she can find comments maximally relevant to the article.

Laurie creates a unipolar concept for 'tidal flooding' by typing in its relevant keywords, starting with the words 'tidal' and 'flooding.' She then sees related words as recommendations in the scatterplot that help her flesh out the concept by adding related terms such as 'flood,' 'floods,' 'tide,' and 'tides,' as shown in Figure 1. She examines the clusters of other terms generated, and decides to avoid words related to specific instances of tidal flooding, such as 'katrina,' or those associated with storms and hurricanes, such as 'storm,' 'raging,' or 'swell.' She wants to keep this a general-purpose concept. Moving on to the second phase, she applies the concept to the comments on the article and immediately notices other key terms, e.g., 'storm,' highlighted as yellow in the retrieved comments. She then adds them to the relevant keyword set of the concept using the integrated concept editor.

Based on her understanding of media framing, Laurie knows that people often discuss complex issues in terms of specific frames relating to definitions, causal interpretations, moral evaluations, and solutions [13], as well as using topical perspectives like economic, political, or scientific. She decides to find a comment to highlight that deals with tidal flooding from the perspective of economic implications. Similar to how she developed the unipolar concept for 'tidal flooding,' she develops another unipolar concept relating to economic implications. She starts with 'economic,' and the scatterplot of recommended words leads her to add related terms such as 'economy' and 'economies,' as well as some of the negative implications that she wants to include, such as 'crisis,' 'impact,' 'turmoil,' and 'instability.' Her economic concept is thus tuned towards negative economic impacts that could arise.

To apply a combinations of these two concepts, Laurie checks the distribution showing all comments plotted against the relevance scores to each of the two concepts (Figure 5). Here she maps the 'tidal

[7] http://www.nytimes.com/2016/02/23/science/sea-level-rise-global-warming-climate-change.html

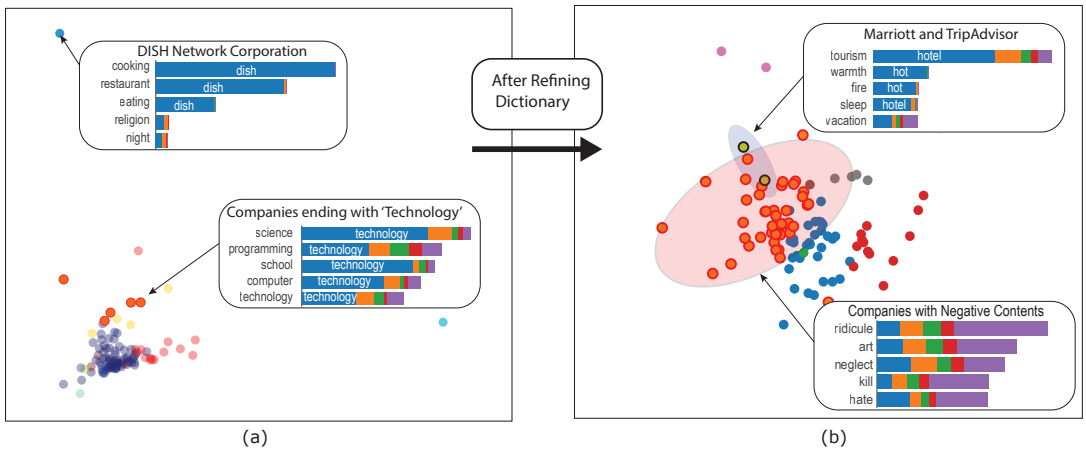


Fig. 4. PCA 2D projection of NASDAQ 100 companies with their k-means clustering labels color-coded, where the feature vector of each company is computed from its tweets' word count in each of 194 concepts. The clustering using the prebuilt lexica shows some outliers (a), where further investigation of contributing words shows that the company name itself acts trivially as strong signals, such as 'dish' in Dish Network Corporation. Another cluster is shown to be formed because of the common word 'technology' in their names. After excluding them in the initial lexicon, more meaningful clusters are revealed. For example, Marriott and TripAdvisor form a cluster because of words in 'tourism,' 'vacation,' and 'sleep' concepts (olive green with a black border). Companies with negative sentiments such as 'ridicules,' 'neglect,' 'kill,' and 'hate' were also clustered together (bright red dots with red border).

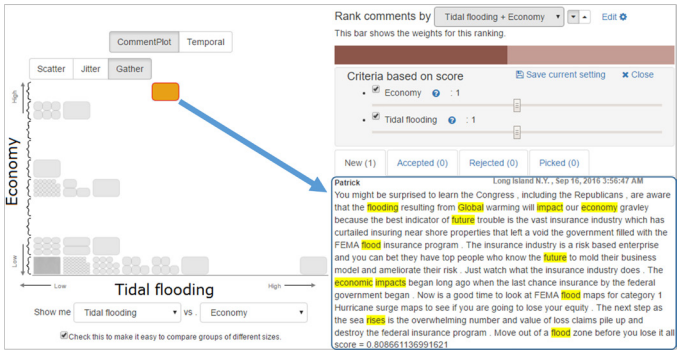


Fig. 5. Distribution of comments across the 'tidal flooding' (X-axis) and the 'economy' (Y-axis) concepts. A comment that has scored relatively high on both concepts is selected (orange box). The content of the corresponding comment within this dataset is shown.

flooding' concept on the x-axis and the 'economy' concept on the y-axis. She then brushes on the scatterplot to find comments containing both concepts, and these comments are filtered into the ranked list. She finds an insightful comment she likes that perfectly combines the two concepts, discussing coastal flooding in terms of impacts to the economy as exposed through the insurance industry. She marks the comment as a "NYT Pick" and it gets highlighted on the site.

She then begins to read those comments with high scores from the top of the list and quickly finds an insightful one indicating that some of the coastal flooding in Virginia has actually been shown to be a result of subsidence of land. Laurie thinks that highlighting this will deepen the discussion online by pointing out the diverse factors that society needs to grapple with as it confronts global warming. Therefore, she marks this comment as an "NYT Pick" as well.

5 The ConceptVector System

Motivated by the limitations of using prebuilt lexica for concept-based document analysis, we designed ConceptVector as a visual analytics system that tightly integrates concept building and refinement with direct support for concept-based document analysis. In detail, our design rationale behind ConceptVector is as follows:

D1 **Supporting diverse user needs in concept building.** Users may have diverse meanings in mind for defining their concepts. Thus, users should be able to construct the lexicon of a concept from scratch and/or refine a prebuilt one to suit to their exact requirements.

D2 **Supporting integrated analysis of iterative lexicon refinement and concept-based document analysis.** As seen from our motivational examples (Section 3), even carefully curated lexica need to be adjusted depending on a document corpus. Thus, the concept-based document analytics system should provide interactive refinement capabilities of a lexicon as well as dynamic document analysis based on the updated lexicon.

D3 **Revealing lexicon word context in documents.** The system should allow users to understand how the words in a lexicon are used in documents in terms of their context.

In this section, we explain how our front-end interfaces and the back-end computational modules support these tasks, and associate each component with design guidelines.

5.1 Front-end Visual Interface

Based on our design rationale, the text analytics process in ConceptVector is composed of two iterative processes: concept building and document analysis (Figure 2). We introduce the two views that allow the user to interactively build concepts and analyze documents.

5.1.1 Concept Building View

As shown in the left pane of Figure 2, the *concept building* process allows a user to interactively build the keyword sets describing a user's intended concept. Figure 1 shows a screenshot of our front-end interface that was taken during this process when the user was building the 'tidal flooding' concept.

We define two types of concepts: bipolar and unipolar. Bipolar concepts have two nontrivial polarities, e.g., positive vs. negative sentiments, happiness vs. unhappiness, etc., while unipolar concepts have a single polarity, e.g., work-related (or not), biology-related (or not), etc. To support both concept types, ConceptVector models a particular concept using three different sets of keywords: positive, negative, and irrelevant (**D1**). In the case of unipolar concepts, the positive keyword set contains those keywords relevant to a concept of interest, while the negative set is an empty set. For both types, the irrelevant keyword set includes the words marked explicitly as irrelevant by the user.

The user starts building a concept by adding seed keywords to describe the concept. ConceptVector then recommends keywords that are potentially relevant to the seed keywords for each positive and negative keyword set, and performs k-means clustering, where we set k as 5, based on their word embeddings. Keyword clusters are presented to the user (Figure 1(3)), along with their 2D embedding view, computed by t-distributed stochastic neighbor embedding (t-SNE) [43] (Figure 1(4)). Checking these recommendation results, the user can either expand the initial keyword set by (1) adding individual words, (2) adding a keyword cluster of them, or (3) move words to the irrelevant set by marking them as irrelevant (**D1**). This iterative *concept building* continues until the user is satisfied with the constructed keyword set.

As relevant (or irrelevant) keywords often appear together in a single cluster, processing words at the cluster level makes the concept building process much more efficient than without clustering (**D1**). For example, if a user enters 'happy' as the only keyword for a concept, irrelevant words such as 'everyone,' 'anyway,' 'yes,' and 'anymore' are recommended as a single cluster, while semantically relevant words such as 'glad,' 'good,' and 'thrilled' form another cluster. When the semantic distinction among words is not clear, users can tag individual words in the cluster. The t-SNE embedding space has very strong neighboring effects [31, 38], placing similar words closely to each other, and hence the 2D embedding view shows the distribution among user-initiated keywords and recommended ones. Users can enter/remove keywords in the t-SNE view as well (**D1**).

5.1.2 Concept-Based Document Analysis View

The concept-based document analysis view, shown in the right pane of Figure 1, allows the user to analyze a document corpus with respect to the constructed concepts. See Section 4 for a detailed description.

Given a single or multiple user-selected concepts, ConceptVector computes the relevance scores of documents for each concept and retrieves/ranks those documents with high score values (Figure 1(10)), which would be meaningful to the user who created/selected the corresponding concept. To help the user understand why these documents have high scores, the significantly contributing keywords are highlighted in yellow color (**D3**). Please note that our relevance scoring algorithm is not limited to the keywords registered in the positive/negative/irrelevant sets, but that other keywords potentially relevant to the concepts are considered as well. We will describe the algorithm further in the following section.

Additionally, ConceptVector provides two different views: a temporal view showing the concept strength over time, and a scatterplot showing the distribution of documents according to the relevance scores for the two different concepts, e.g., 'tidal flooding' vs. 'economy' concepts (Figure 5). According to the Jänicke et. al., extraction, evolution, and clustering are the three main tasks in visual text analysis [17]. The temporal view supports the temporal tracking of the topic signal evolution, while the scatterplot allows mapping/clustering documents in semantic space. Users can assign user-defined concepts as axes of the scatterplot to explore the distribution of the semantic meaning of documents (**D2**). Note here that we use a modified version of a scatterplot, where both dimensions are binned and dots are scaled to fill the assigned space [36]. This improves the visibility of outliers and densely overplotted areas. In these views, the user can brush over a time axis or data items to filter data in the ranked retrieval results.

During the process, the user may add additional words to the relevant and the irrelevant keyword sets of the concept (**D2**). For example, when applying the 'tidal flooding' concept shown in Figure 1 to a document corpus, the word 'disaster' was highlighted owing to its high relevance score to the concept. Since this word is not related to the 'tidal flooding' concept, the user can add it to the irrelevant keyword set to revise the concept and update the ranking of documents accordingly. This interaction allows in-situ concept refinement.

Note that the two analysis tasks of concept building and document analysis are not separate but tightly connected in ConceptVector, so that the user can fluidly switch between concept building/refinement and document analysis based on concepts.

5.2 Back-end Relevance Scoring Model

ConceptVector is built upon the vector representations of words generated by word embedding techniques such as word2vec [31] or GloVe [38]. In this step, the training corpus for word embeddings could be a generic one such as Wikipedia articles or a corpus within a particular domain, so that the trained vectors can better reflect the semantics of the domain. ConceptVector currently adopts pretrained vector embedding using Wikipedia articles by GloVe.[8] ConceptVector represents a concept C as the three set of keywords: the positive, the negative, and the irrelevant ones—L_p, L_n, and L_i, respectively. Given a word or a document, ConceptVector computes its relevance scores to the concept, based on the probability of a given word belonging to each of L_p, L_n, and L_i using a kernel density estimation (KDE) method.

In detail, let us denote q as the vector representation of a query word, l as that of the keyword contained in the keyword set L, where L can be one of L_p, L_n, or L_i. We define the probability of q belonging to L as

$$p(q|L) = \frac{1}{|L|} \sum_{x \in L} k(q, l), \tag{1}$$

where $k(q, l)$ represents a kernel function computing the similarity value between the two word vectors q and l. That is, Eq. (1) computes the average similarity values between q and each word l contained in a particular keyword set L. The reason for using a kernel function instead of a simple similarity measure such as cosine similarity is because this provides not only a user-controllable, flexible similarity measure but also a principled probabilistic framework of incorporating multiple similarities of q with L_p, L_n, and L_i, as will be described later.

The choice of the kernel function $k(q, l)$ can vary, but in ConceptVector, we adopted a Gaussian kernel defined as

$$k(q, l) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\|q - l\|_2^2}{\sigma^2}\right),$$

where σ^2 is the bandwidth parameter that determines how quickly the similarity decreases as the L^2 distance increases. A small bandwidth value gives a high similarity only on the words exactly contained in L, which is suitable when L contains many words and a user does not want to consider other words outside L as relevant to the concept. A large bandwidth, on the other hand, will consider many of the outside words as relevant to L, which is useful when a user wants to define the concept in a broad and flexible manner, not just limited to those words contained in L.

Viewing $p(q|L)$, which is computed by Eq. (1), as the likelihood in a Bayesian context, we can define the prior probability $p(L)$ and the posterior probability $p(L|q)$, respectively, as

$$p(L) = \frac{|L|}{|L_p| + |L_n| + |L_i|}, \text{ and}$$

$$p(L|q) = \frac{p(L) \cdot p(q|L)}{p(L_p) \cdot p(q|L_p) + p(L_n) \cdot p(q|L_n) + p(L_i) \cdot p(q|L_i)}.$$

Using these, the final relevance score $r(q, C)$ of a query word q to the concept C is computed as

$$r(q, C) = (1 - p(L = L_i|q)) \cdot \big(p(L_p) \cdot p(q|L_p) - p(L_n) \cdot p(q|L_n)\big)$$

Basically, $r(q, C)$ computes the differences between the joint probabilities $p(q, L_p)$ and $p(q, L_n)$, ranging between -1 and $+1$, and furthermore, as $p(L = L_i|q)$ increases, $r(q, C)$ becomes close to zero, indicating irrelevance to the concept.

In the case of a unipolar concept, the relevance score is computed in the exact same manner by setting $L_n = \emptyset$. These bipolar scores and unipolar scores are used for recommendation of relevant words.

Finally, the relevance score of a document to a particular concept is computed by simply taking the average relevance score among all the words contained in a document.

[8] http://nlp.stanford.edu/projects/glove/

5.3 Implementation Details

ConceptVector was implemented as a web-based application using D3 and AngularJS. We employed the New York Times online article comments as our corpus; naturally, the approach can be applied to any document corpus. We selected articles with more than 300 comments from the most popular articles during the period August to September 2016. Articles and comments were collected using the NYT API.[9]

The back-end computational modules were implemented using Python with the Flask framework.[10] The key computation shown in Eq. (1) for recommending relevant words requires computing the one-to-all distances for all words in the current keyword set (either positive, negative, or irrelevant). Computing a single one-to-all distance repeatedly due to frequent user interaction may slow down the overall process. We instead compute the one-to-all distance incrementally with a cache that contains recently computed pairs. This is possible because the user incrementally adds a single word at a time to the keyword set. To this end, a least recently used cache of size 10,000 word pairs was employed, resulting in a speed-up of efficient user interactions.

6 Evaluation

Visual analytics systems comprise many interconnected components, and this complicates their overall evaluation. Here we separate the visual interface and the back-end computation and evaluate them individually with a user study and a quantitative evaluation, respectively. For the front-end, we focus on the effectiveness of the concept-building view because document analysis requires analysts with domain knowledge and is subjective to inter-analyst differences. For the back-end, we validate the effectiveness of supporting the process of building bipolar concepts. Although we did not evaluate a unipolar case, we generally expect the same level of effectiveness since the process of building unipolar concepts is similar yet simpler than the process of building bipolar concepts. Finally, we also include results from an expert review comparing ConceptVector to Empath [14] to show ConceptVector's performance in relation to the state of the art.

6.1 Evaluation of Concept Building

We conducted a user study to evaluate how users generate lexica with ConceptVector compared to WordNet [33][11] and Thesaurus.com[12] as baselines. WordNet is known for its large-scale lexical database, and Thesaurus.com is an online thesaurus containing exhaustive synonyms and antonyms for the English language. We employed the following performance metrics: (i) the completion time for building concepts, and (ii) the quality of the resulting concepts.

6.1.1 Methodology

We recruited 15 graduate students (1 female and 14 males) majoring in computer science to participate in the study. All participants reported high computer skills.

Each study session lasted 15–25 minutes and involved three systems: ConceptVector, WordNet, and Thesaurus.com. Before starting the session, a test administrator briefly explained how to use the systems and allowed the participant to spend enough time to familiarize themselves. Participants were then asked to build a lexicon for three concepts: 'family,' 'body,' and 'money,' which we selected as relatively neutral and easily comprehensible by all participants. Each participant was randomly assigned to a system for each concept so that at the end of the study they had used all three conditions. Each concept-building task was capped at three minutes. All three systems, including ConceptVector, were accessed by their official websites. We recorded both the lexicon each participant created as well as the number of keywords in it as a dependent variable.

As the ground truth lexicon for each concept, we selected three dictionaries from Linguistic Inquiry and Word Count (LIWC) 2007 [37]. The ground truth lexicon sizes of the three concepts are 65 words for 'family,' 180 for 'body,' and 173 for'money.' We adopted widely used information retrieval evaluation metrics, precision and recall, where precision is the fraction of correct answers over the total number of answers given, and recall is the fraction of retrieved correct answers out of all correct ones. The null hypothesis assumes that the difference of methods does not affect the precision, recall, or average number of words in the resulting lexicon.

6.1.2 Results

Table 1 shows precision, recall, and average total words generated for the three methods. ConceptVector achieved the highest scores in all three metrics, indicating that the user-created lexicon using ConceptVector is the most accurate and most time-efficient. We further analyzed the effect of employing ConceptVector using mixed linear model analysis, where the fixed effect is the choice of methods (ConceptVector, WordNet, and Thesaurus.com) and the random effect is the choice of specific concepts ('family,' 'money,' and 'body').

Figure 6 shows boxplots for precision, recall, and total words generated. We used a pairwise Tukey HSD method to test statistical significance between different methods. There was a significant performance boost of employing ConceptVector on recall ($F(2,40) = 5.25, p = .0094$). Pairwise Tukey HSD between ConceptVector and the other methods showed significant differences ($p < .05$). There was also a significant main effect for technique T on precision ($F(2,40) = 5.22, p = .0096$). Pairwise comparisons with a Tukey HSD showed significant differences ($p < .05$) between ConceptVector and Thesaurus.com. Finally, there was a significant main effect for technique T on the number of total words generated ($F(2,40) = 5.40, p = .0084$). Pairwise comparisons with a Tukey HSD showed significant differences ($p < .05$) for ConceptVector and WordNet.

Recall rates in all three systems are relatively low compared to the high precision rates. This is mainly because the size of the ground truth lexicon is much larger than the average size of the lexicon a person can create within a short period of time (three minutes in our case). As seen in Table 1, the average size of the created lexicon was around 8 to 15 depending on the system.

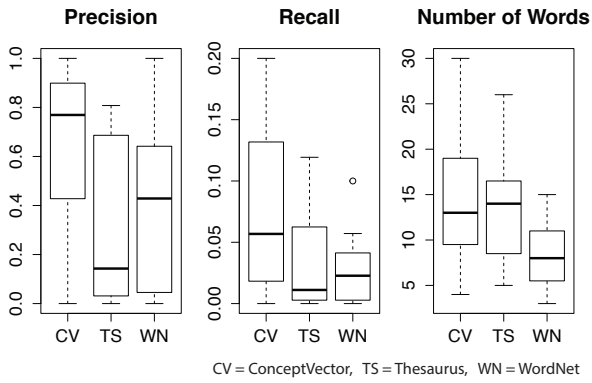


Fig. 6. Boxplots comparing the three methods in terms of precision, recall, and the size of resulting lexicon. ConceptVector shows the best result.

Since the current experimental design does not consider polysemy or subtle nuance differences, this experiment could be improved further by employing more sophisticated ground truth data instead of the current ones obtained from LIWC. For example, the 'family' concept may diverge in terms of its subtler meanings to different people. On the one hand, it may correspond mainly to the members of a family such as 'mother,' 'grandfather,' and 'son.' On the other hand, it may correspond to emotional words such as 'love,' 'rest,' and 'nursing.' Since our ground truth lexicon from LIWC was mostly composed of the keywords from the first case, the user-generated keywords from the second case were treated as false positive words. We expect that ConceptVector will perform better if we find ground truth data that enable us to measure

[9] http://developer.nytimes.com/

[10] http://flask.pocoo.org/

[11] http://wordnetweb.princeton.edu/perl/webwn

[12] http://www.thesaurus.com/

Table 1. Precision, recall, and average number of keywords per concept for three methods constructing user-defined concepts. The values in parentheses indicate the standard deviation. See Section 6.1.2 for details.

Metrics	ConceptVector	Thesaurus	WordNet [33]	F value	Pr > F
Precision	**0.6363** (0.1701)	0.3099 (0.3773)	0.3794 (0.3637)	5.22 [2, 40]	0.0096
Recall	**0.0789** (0.0308)	0.0333 (0.0385)	0.0275 (0.0242)	5.25 [2, 40]	0.0094
Average word count	**15.6667** (7.4536)	13.8000 (6.0685)	8.2667 (3.3360)	5.40 [2, 40]	0.0084

these richer relations, and this will be one of our future directions. Furthermore, regardless of which type of concept a user had in mind, ConceptVector properly supported the concept-building process by recommending suitable keywords for different cases. This indicates the flexibility and the affordance that ConceptVector offers compared to other, more rigid, systems.

6.2 Quantitative Evaluation of Bipolar Concepts

We validate the bipolar concept model supported by ConceptVector to address the following two questions:(1) Does our proposed approach generate relevance scores comparable to human judgments? and (2) How many input words are required to properly model concepts? To answer these questions, we conducted a quantitative analysis.

6.2.1 Experiment Setup

Validation of a lexicon requires ground truth. For unipolar concepts, the prior work from Fast et al. compared the result with "golden standard dictionaries" such as LIWC and GI [14]. While many lexica for unipolar concepts have been developed, bipolar lexica are rare. In this study, we adopted a keyword database available from the Hedonometer project[13] [11]. This database contains a ranked list of 10,200 keywords in terms of their relevance to the concept of 'happiness,' where the ranking was determined by crowdsourcing. The word ranking begins with the happiest word and ends with the saddest word. From this database, we selected 9,600 words from the intersection of the Hedonometer ranking and the vocabulary set from the Wikipedia corpus[14] used to train our word embedding model. From the Wikipedia corpus, we removed 71,697 documents that no longer exist, and used the resulting 171,729 articles. We then removed the words containing nonalphanumerical characters as well as those appearing less than ten times in the entire document corpus, resulting in 142,275 keywords in total.

The goal of our experiments was basically to evaluate how well the ranking of words computed by our back-end algorithm matches with the ground truth ranking, given a subset of top and bottom k words as positive and negative sets, respectively, to form a concept. As the methods to generate word vector representations, we used two different word embedding techniques—word2vec [31] and GloVe [38]—as well as a baseline method, latent semantic indexing (LSI) [9]. Additionally, in each vector space, we compared our KDE-based algorithm against logistic regression for computing the word-to-concept relevance score and the associated word ranking. As an evaluation measure, we computed Spearman's rank correlation coefficient between the ranking of the ground truth and that from each different case.

6.2.2 Comparison Results

Figure 7 shows the Spearman's rank correlation coefficients obtained for various word embedding and relevance scoring methods by varying the value of k in the top k and the bottom k keywords used to train each model. In general, given a small number of input keywords less than 200, the algorithm was shown to generate a reasonably good rank correlation of more than 0.4. In addition, as we increase k, the rank correlation increases in all cases, indicating that more information helps the model learn the intended concept (happiness in this case). Between the two word embedding methods and LSI, the former showed a rapidly increasing performance even with a small number, e.g., around 100, of keywords necessary for training. Between our KDE-based scoring method and logistic regression, the former outperformed the latter

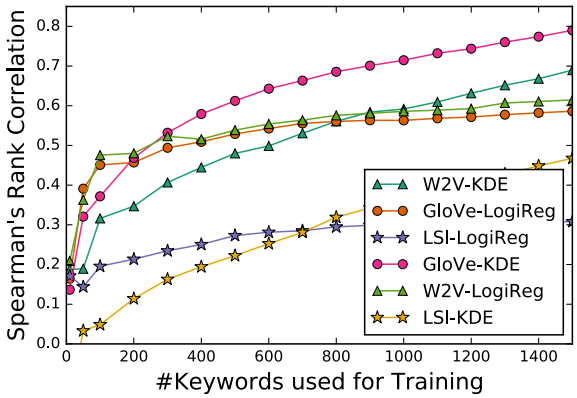


Fig. 7. Spearman's rank correlation coefficient results with respect to the number of keywords used for training. `KDE` stands for kernel density estimation, `LogiReg` for logistic regression, `W2V` for word2vec [31], `LSI` for latent semantic indexing, and `GloVe` for GloVe [38].

method when the size of the keyword set is sufficient, e.g., more than 300. Furthermore, the performance of our KDE-based method consistently increases by a large margin compared to competing methods.

Among different word embedding techniques, the GloVe model followed by the KDE-based method achieved the best rank correlation performance of around 0.8. Word2vec performed relatively well, but it was inferior to GloVe in our task. On the other hand, traditional methods such as LSI do not perform well in this task, showing a rank correlation of 0.45 even with large k values. Finally, the overall performance gain due to the increase of the embedding dimensions was not significant.

Our experiment involves only bipolar concepts (no unipolar ones), and we did not examine the effect of an irrelevant keyword set. In this case, logistic regression may not be applicable at all. In addition, we found that the ground truth ranking is not always correct, especially among the mid-ranked unclear words. However, the results presented here highlight the potential superiority of our proposed KDE-based scoring approach combined with GloVe, and in the next section, we present results from an expert review to show the effectiveness of the system when used in practice.

6.3 Expert Review

To evaluate the visual interface of ConceptVector in depth, we engaged two experts, one in text analytics (P1) and one in visual analytics (P2), to provide qualitative feedback on ConceptVector compared to the Empath system. Both were postdoctoral researchers, the former of which conducts research on social networks, crisis analytics, and credibility in social media, while the latter studies the healthcare domain that frequently involves text mining and visualization (e.g., electronic medical records). We began with a 15-minute tutorial introducing our system and Empath. Afterwards, the experts built their own custom concepts using both systems. Those concepts were used in the analysis of New York Times comments. During the process, we gathered the feedback on both the model and the visual interface of the system. The online version of Empath was used as a reference.[15]

Both P1 and P2 agreed that the recommended keywords given by

[13] http://hedonometer.org/

[14] https://cs.fit.edu/ mmahoney/compression/textdata.html

[15] http://empath.stanford.edu/

ConceptVector, which are shown in a semantically meaningful grouping in a scatterplot, were easier to grasp than the simple list given by Empath. The scatterplot helped them digest the generated words by providing a high-level overview (P2) or chunking the words into semantically homogeneous groups (P1). It was especially useful in the early stage of concept building, because irrelevant words formed a separate group in many cases, allowing the user to spot them easily and mark them as irrelevant. The word clusters (Figure 1(3)) were good for reading words quickly (P2), and was used during most of the concept-building process (P1). Furthermore, the t-SNE view (Figure 1(4)) provided an additional benefit of showing the similarities between words (P1) and the relationships between the input terms and the recommended terms (P2). For example, whether the input words form tight clusters or not gives a visual clue as to whether the generated concept is consistent (P1). At the same time, P1 noticed that an input term was actually an outlier compared to other input terms forming a packed cluster. After examining this word, he removed it because it had a very broad meaning and thus dilutes the clarity of the concept.

Both experts noted that the difference between the corpora used to train word embedding affects the concept quality. Empath used modern amateur fiction data, but ConceptVector used the Wikipedia dataset. For example, when P1 used 'politics,' 'voting,' and 'elections' as seed terms in Empath, the generated words contained several words such as 'shipping' and 'readers' which did not really make sense. According to P1, Empath also generated more 'high school'-related words. This does not necessarily mean that one system is better but rather that using word embedding trained by a corpus suitable for target corpus to analyze is important. After building a concept about 'grievance,' P1 noted *"The recommended words for the grievance concept is different from what I saw on social media. That is, many legalese and lengthy words related to grievance were recommended, but very unlikely to show up on social media."* P2 suggested using ConceptVector as a tool to evaluate multiple versions of word embedding models during iterative model development.

P1 and P2 both agreed that comparing Empath and ConceptVector is challenging, because the main focus of Empath is not its user interface. P1 thought the visual interface in ConceptVector was useful to explore the semantic space. Being able to look around and select words that are not originally shown to him helped to expand the lexicon. P2 pointed out that the document analysis feature of Empath is more of a blackbox and felt uncomfortable with trusting the result. For example, when analyzing the Wikipedia page about 'Ramen,' the 'friends' category was ranked as the 6th, but it is not clear which words in the friends category were counted.

P1 noted that the word-highlighting feature of ConceptVector allows for the easy spotting of false positives, but detecting false negatives is not currently supported. P2 appreciated the concept score scatterplot (Figure 1(7)) that showed the distribution of comments with respect to custom concepts as axes. It revealed outliers and enabled filtering of comments based on semantic contents. After using ConceptVector, P2 said that it could be useful to build a concept for drugs by adding related symptoms and using a positive/negative sentiment as another axis to visualize the sentiment for a particular drug. P2 also liked that the concept dictionary can be refined by trial and error.

P1 expressed concern about fundamental limitations of both systems. Both systems use word embedding based on the assumption that word co-occurrence statistics reflect semantic similarities, which might not be always true in real-world text analysis. P1 pointed out that while the color coding of words to highlight newly recommended words is an improvement over Empath, it was still difficult to follow the word changes according to the input terms. P2 liked the bipolar concepts feature because it helps in building more sophisticated concepts. As an alternative design, P2 suggested showing the words interpolating positive and negative terms. Those interpolated words will reveal the validity of a concept, as suggested in Axisketcher [22].

7 Discussion

ConceptVector is a novel approach for text analysis that falls somewhere between sentiment analysis performed using manually constructed dictionaries, and topic modeling performed by automatic algorithms. This unique position brings new benefits as well as limitations.

In general, when achieving a particular analytic goal, an interesting tradeoff between quality and efficiency can be considered. That is, human efforts secure the quality of the outcome, while automated approaches can significantly boost the efficiency of our efforts. For concept building, purely manual approaches such as LIWC and Hedonometer can be viewed as extreme cases, where the task relies completely on human effort. Thus, the resulting dictionary is of high quality, but it is achieved by an inefficient, costly process without automation. On the other hand, purely automated approaches such as topic modeling, which generate multiple sets of semantically coherent words, maximize the efficiency of the task, but the quality of the outcome cannot be controlled by the user. Human labor is still needed to interpret the results that such fully automatic approaches generate.

In this sense, our approach in the ConceptVector system can be viewed as a balanced—or hybrid—case, where both efficiency and interpretability are achieved via a synergetic blending of both human efforts and automated machine computations. That is, our main steps of adding and removing keywords to construct a particular concept are all confirmed by humans, and in this manner, a high quality outcome is maintained. However, our system significantly accelerates these human-guided processes by crucial automated approaches, including word recommendation based on word embedding, followed by word grouping and visual presentation. Also, after users build a specification, this specification is used to build the concept model, which calculates the relevance scores of all words with this particular concept. In this respect, our system represents an illustrative example for properly achieving human-machine collaborations. As it happens, this is also precisely in line with the visual analytics philosophy, where automatic algorithms and visual interfaces create synergies .

8 Conclusion and Future Work

Current text analytics methods are either based on manually crafted human-generated dictionaries or require the user to interpret a complex, confusing, and sometimes nonsensical topic model generated by the computer. In this paper we proposed ConceptVector, a novel text analytics system that takes an visual analytics approach to document analysis by allowing the user to iteratively define concepts with the aid of automatic recommendations provided using word embeddings. The resulting concepts can be used for concept-based document analysis, where each document is scored depending on how many words related to these concepts it contains. We crystallized the generalizable lessons as design guidelines about how visual analytics can help concept-based document analysis. We compared our interface for generating lexica with existing databases and found that ConceptVector enabled users to generate concepts more effectively using the new system than when using existing databases. We proposed an advanced model for concept generation that can incorporate irrelevant words input and negative words input for bipolar concepts. We also evaluated our model by comparing its performance with a crowdsourced dictionary for validity. Finally, we compared ConceptVector to Empath in an expert review.

The text analysis provided by ConceptVector enables several novel concept-based document analysis, such as richer sentiment analysis than previous approaches, and such capabilities can be useful for data journalism or social media analysis. There are many limitations that ConceptVector does not solve. Among these, the selection/integration of multiple heterogeneous training data according to the target corpus and the automatic disambiguation of multiple meanings of words according to the context are promising avenues of future research.

Acknowledgments

Research reported in this publication was partially supported by NIH grant R01GM114267 and the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIP) (No. NRF-2016R1C1B2015924). Any opinions, findings, and conclusions or recommendations expressed in this article are those of the authors and do not necessarily reflect the views of the funding agencies.

REFERENCES

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[2] M. Berger, K. McDonough, and L. M. Seversky. cite2vec: Citationdriven document exploration via word embeddings. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):691–700, Jan 2017.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[4] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the ACM Conference on Research and Development in Information Retrieval*, pp. 335–336, 1998.

[5] S. Chang, P. Dai, L. Hong, C. Sheng, T. Zhang, and E. H. Chi. AppGrouper: Knowledge-based interactive clustering tool for app search results. In *Proceedings of the International Conference on Intelligent User Interfaces*, pp. 348–358, 2016.

[6] J. Choo, C. Lee, C. K. Reddy, and H. Park. UTOPIAN: User-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):1992–2001, 2013.

[7] J. Chuang, C. D. Manning, and J. Heer. Termite: Visualization techniques for assessing textual topic models. In *Proceedings of the ACM Conference on Advanced Visual Interfaces*, pp. 74–77, 2012.

[8] W. Cui, S. Liu, L. Tan, C. Shi, Y. Song, Z. Gao, H. Qu, and X. Tong. TextFlow: Towards better understanding of evolving topics in text. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2412–2421, 2011.

[9] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the Society for Information Science*, 41:391–407, 1990.

[10] N. Diakopoulos. Picking the NYT Picks: Editorial criteria and automation in the curation of online news comments. *ISOJ Journal*, 2015.

[11] P. S. Dodds, K. D. Harris, I. M. Kloumann, C. A. Bliss, and C. M. Danforth. Temporal patterns of happiness and information in a global social network: Hedonometrics and twitter. *PLOS ONE*, 6(12):e26752, 2011.

[12] W. Dou, X. Wang, R. Chang, and W. Ribarsky. ParallelTopics: A probabilistic approach to exploring document collections. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology*, pp. 231–240, 2011.

[13] R. M. Entman. Framing: Toward clarification of a fractured paradigm. *Journal of Communication*, 43(4):51–58, 1993.

[14] E. Fast, B. Chen, and M. S. Bernstein. Empath: Understanding topic signals in large-scale text. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pp. 4647–4657, 2016.

[15] A. Grover and J. Leskovec. Node2vec: Scalable feature learning for networks. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining*, pp. 855–864, 2016.

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

[17] S. Jänicke, G. Franzini, M. F. Cheema, and G. Scheuermann. Visual text analysis in digital humanities. *Computer Graphics Forum*, pp. n/a–n/a, 2016. doi: 10.1111/cgf.12873

[18] I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 1986.

[19] S. Kim, J. Lee, G. Lebanon, and H. Park. Estimating temporal dynamics of human emotions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 168–174, 2015.

[20] S. Kim, J. Lee, G. Lebanon, and H. Park. Local context sparse coding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 2260–2266, 2015.

[21] S. Kim, F. Li, G. Lebanon, and I. Essa. Beyond sentiment: The manifold of human emotions. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pp. 360–369, 2013.

[22] B. C. Kwon, H. Kim, E. Wall, J. Choo, H. Park, and A. Endert. Axisketcher: Interactive nonlinear axis mapping of visualizations through user drawings. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):221–230, 2017.

[23] B. C. Kwon, S.-H. Kim, S. Lee, J. Choo, J. Huh, and J. S. Yi. VisOHC: Designing visual analytics for online health communities. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):71–80, 2016.

[24] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *Proceedings of the International Conference on Machine Learning*, pp. 1188–1196, 2014.

[25] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

[26] H. Lee, J. Kihm, J. Choo, J. Stasko, and H. Park. iVisClustering: An interactive visual document clustering via topic modeling. *Computer Graphics Forum*, 31(3pt3):1155–1164, 2012.

[27] W. Ling, C. Dyer, A. Black, and I. Trancoso. Two/too simple adaptations of Word2vec for syntax problems. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1299–1304, 2015.

[28] W. Ling, Y. Tsvetkov, S. Amir, R. Fermandez, C. Dyer, A. W. Black, I. Trancoso, and C.-C. Lin. Not all contexts are created equal: Better word representations with variable attention. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1367–1372, 2015.

[29] C. D. Manning, P. Raghavan, and H. Schutze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[30] T. Mikolov, Q. V. Le, and I. Sutskever. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168, 2013.

[31] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*, pp. 3111–3119, 2013.

[32] T. Mikolov, W.-t. Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 746–751, 2013.

[33] G. A. Miller. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.

[34] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2008.

[35] D. Park, S. Sachar, N. Diakopoulos, and N. Elmqvist. Supporting comment moderators in identifying high quality online news comments. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pp. 1114–1125, 2016.

[36] D. G. Park, S.-H. Kim, and N. Elmqvist. Gatherplots: Extended scatterplots for categorical data. Technical Report HCIL-2016-10, University of Maryland, College Park, 2016.

[37] J. W. Pennebaker, M. E. Francis, and R. J. Booth. *Linguistic inquiry and word count: LIWC 2001*. Lawrence Erlbaum Associates, 2001.

[38] J. Pennington, R. Socher, and C. D. Manning. GloVe: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1532–1543, 2014.

[39] P. J. Stone, D. C. Dunphy, M. S. Smith, and D. M. Ogilvie. *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press, 1966.

[40] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede. Lexicon-based methods for sentiment analysis. *Journal Computational Linguistics*, 37(2):267–307, 2011.

[41] E. M. Talley, D. Newman, D. Mimno, B. W. Herr II, H. M. Wallach, G. A. Burns, A. M. Leenders, and A. McCallum. Database of NIH grants using machine-learned categories and graphical clustering. *Nature Methods*, 8(6):443–444, 2011.

[42] F. Tian, H. Dai, J. Bian, B. Gao, R. Zhang, E. Chen, and T.-Y. Liu. A probabilistic model for learning multi-prototype word embeddings. In *Proceedings of the International Conference on Computational Linguistics*, pp. 151–160, 2014.

[43] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[44] F. Wei, S. Liu, Y. Song, S. Pan, M. X. Zhou, W. Qian, L. Shi, L. Tan, and Q. Zhang. TIARA: a visual exploratory text analytic system. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining*, pp. 153–162, 2010.