DycomDetector: Discover topics using automatic community detections in dynamic networks

Tommy Dang*
Texas Tech University
P.O. Box 43104
Lubbock, Texas 79409-3104
tommy.dang@ttu.edu

Vinh Nguyen†
Texas Tech University
P.O. Box 43104
Lubbock, Texas 79409-3104
vinh.nguyen@ttu.edu

Md. Yasin Kabir‡
Texas Tech University
P.O. Box 43104
Lubbock, Texas 79409-3104
yasin.kabir@ttu.edu

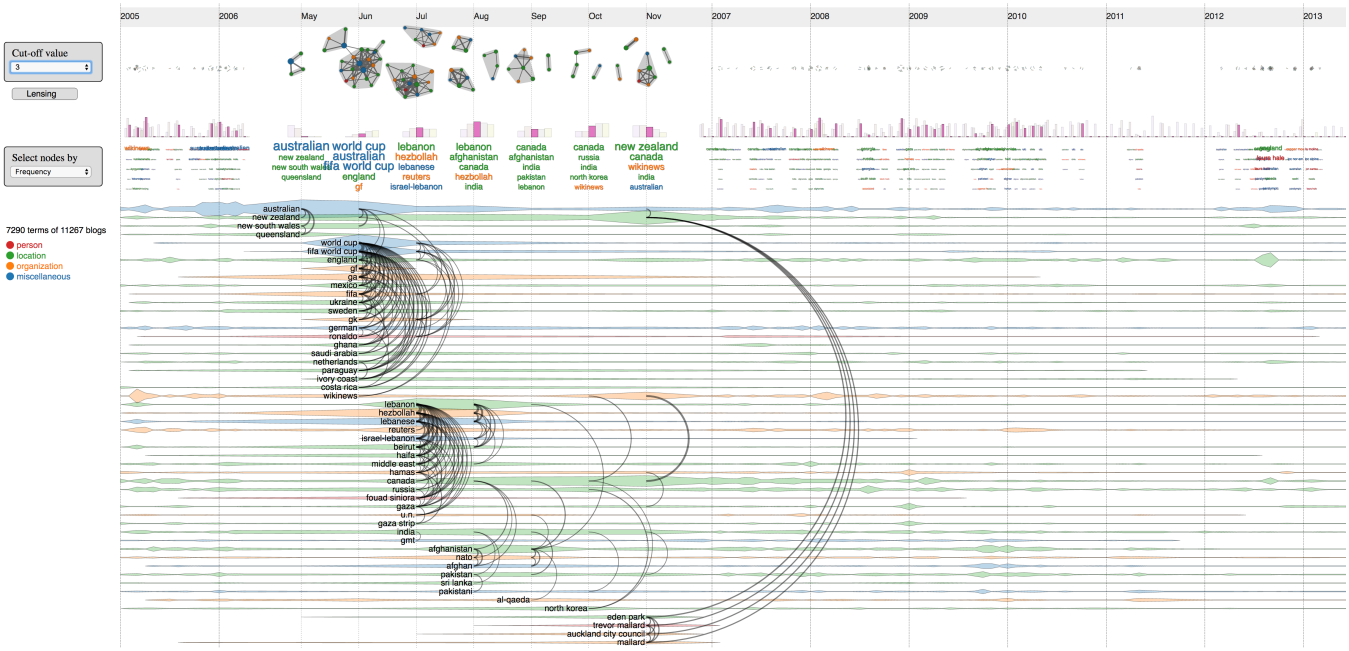


Figure 1: *DycomDetector* visualization: nodes are popular terms from Wikinews which are color-coded by categories, and links indicate the co-allocation of terms in news. The term networks for each month are displayed on the top with their features and details at the bottom.

ABSTRACT

Due to the rapid expansion and heterogeneity of the data, it is a challenging task to discover the trends/patterns and relationships in the data, especially from a corpus of texts from published documents, news, and social media. In this paper, we introduce *DycomDetector*, a novel approach for topic modeling using community detections in dynamic networks. Our algorithm extracts the important terms/phrases, formulates a network of collocated terms, and then automatically refines the network on various features (such as *term/relationship frequency*, *sudden changes* in their time series, or vertex *betweenness centralities*) to reveal the structure/communities in the given network. These communities are corresponded to different hidden topics in the input texts. *DycomDetector* provides an intuitive interface and supports a range of interactive features, such as lensing or filtering, allowing users to quickly narrow down events of interest. We also demonstrate the applications of *DycomDetector* on several real world datasets to evaluate its capabilities.

Paper type: Novel research paper

KEYWORDS

Topic Modeling, Latent Dirichlet Allocation, community detections, dynamic networks, Storyline Visualization

ACM Reference format:
Tommy Dang, Vinh Nguyen, and Md. Yasin Kabir. 2017. *DycomDetector*: Discover topics using automatic community detections in dynamic networks. In *Proceedings of , Halifax, Nova Scotia, Canada, August 14th, 2017 (KDD 2017 Workshop on Interactive Data Exploration and Analytics (IDEA'17))*, 6 pages.

*Dr. Tommy Dang is with the Department of Computer Science at Texas Tech University.
†Vinh Nguyen is with the Department of Computer Science at Texas Tech University.
‡Md. Yasin Kabir is with the Department of Computer Science at Texas Tech University.

KDD 2017 Workshop on Interactive Data Exploration and Analytics (IDEA'17), Halifax, Nova Scotia, Canada
2017. .
DOI:

DOI:

1 INTRODUCTION

We are living in the age of big data in which a vast amount of digitalized information being collected grows exponentially. The expansion of IT infrastructure along with broadband uses helps us access information instantly. Although more data becomes available, accessing to what information we are looking for is still a challenging task due to the level of the details we are trying to achieve.For example in 2016, Twitter had approximately 500 million tweets per day, while Facebook had 216 million posts within the same time scale [21]. One might be interested in various specific questions such as: what were the hottest topics on Twitter last month? Are there any commonalities among different demographics of people? Or how did interests about a particular topic change over time? These intriguing questions have been motivating researchers to look for answers over the years. Thus the need to have automated tools and techniques to filter, organize, and explore vast quantities of information is highly desirable.

Our contributions in this paper thus are:

- We present a new approach for discovering topics based on graph theory, specifically community detection in networks.
- We provide an interactive data analytics prototype, called *DycomDetector*, for visualizing and analyzing topic abstractions and how they change overtime.
- We provide use cases on other application domains and make comparisons to a classic topic modeling algorithm.

The rest of this paper is organized as follows. Section 2 reviews related work and existing methodologies and Section 2.3 discusses our approach for community detection in networks. Section 3 introduces the *DycomDetector* prototype and illustrates it on real datasets. In Section 4, we present our experiments on real-world topic discoveries in dynamic networks. Finally, we conclude our paper with future work.

2 RELATED WORK

2.1 Topic Modeling

Latent Dirichlet Allocation (LDA) [3] is a flexible generative probabilistic model for text mining. LDA aims to find potential topics in the text corpora. In this mining approach, documents are treated as random mixtures of certain number of topics, and LDA categorizes the topics based on the distribution over the words through a three-level hierarchical Bayesian model. Doyle et al. [13] implement basic LDA for financial topics modeling for stock market to detect the companies which tend to move together. Wang et al. [25] extend LDA into *Spatial Latent Dirichlet Allocation* (SLDA), which encodes spatial structures among visual words into the same topic. *LDAAnalyzer* [30], a tool designed for software engineering researchers, used for source code modeling and numerical data visualization was developed based on LDA.

Topic modeling enables us to organize, re-order, and summarize large text corpora in an effective way. Hence, it is used highly for the visualization of large data in a time efficient manner. Many good visualization tools and techniques have been developed for the visualization of topics. Wei et al. [27] present TIARA, which determines time-sensitive keywords to portray the content evolution of each topic over time using stack graph metaphor. ParallelTopics [12] was also developed to represent the temporal changes of topics using Parallel Coordinates.

2.2 Dynamic Network Visualization

Improved technologies and devices enable us to gather data which are changing in every moment. The large temporal data from various fields motivate the creation of novel visualization techniques. Beck et al. [2] present state of the art in visualizing dynamic data which provide an overview of the novel techniques for representing relational data. This survey provides a hierarchical taxonomy of dynamic graph visualization and classifies the existing techniques into the taxonomy based on a systematic literature review. The survey shows that time-line based techniques (time-to-space mapping) are becoming more popular in dynamic visualization.

For visualizing the temporal changes in dynamic networks [6, 23, 29] matrices are very useful. When visualizing dense graphs [15] adjacency matrices are particularly effective as they avoid edge-crossing problem in node-link diagrams [10, 14, 18]. *TimeMatrix* [29] displays a modest temporal bar chart inside each cell of the matrix which allow comparing the changes of edge weights for the two corresponding vertices. Alternatively, *gestaltmatrix* [7] uses *gestaltlines*, intra-cell lines that encode various metrics utilizing the angle and length. *Matrix Cubes* [1] stacks adjacency matrices in chronological order to represent the dynamic networks based on the space-time cube metaphor.

2.3 Community detection in networks

One of the most widely used algorithms for community detection in practice is the Louvain method [4] because of its speed and desired modularity value. This is a greedy optimization method, that is, it first looks for "small" communities by optimizing modularity locally, then each small community is grouped into one node and the first step is repeated iteratively until a maximum of modularity is attained and a hierarchy of communities is produced.

3 *DYCOMDETECTOR* VISUALIZATION

To enable users to explore the vast temporal text corpora in an interactive and efficient way, *DycomDetector* visualization introduces several components and following steps:

- **Compute and extract the terms**: Our method extracts terms from text data based on frequency and standardized net frequency of the entities at each time point. We then rank them and filter only the top 200 terms. We describe this step in Section 3.1.
- **Construct relationships**: This step constructs the relationships between terms/phrases based on the contexts that they are situated together in Section 3.2.
- **Redefine and reorder the vertices**: This process allows reordering the vertices based on a user selected parameter in Section 3.3.
- **Generate visualizations**: The visualization is generated in this step. Due to the limited screen display, each network (at each time point) is represented as a thumbnail which summarizes the structure of the network in Section 3.4.

- **Interations**: Users can explore popular terms and dynamic relationships between them via various interactions and selections supported within *DycomDetector*. (Section 3.5).

The *DycomDetector* implements four low-level analysis tasks:

- **T1**: Provide a summary view of text corpus over time [17]. *DycomDetector* provides a quick overview of important topics using the network thumbnails. Moreover, we also display a summary histogram of network modularity on different settings as well as the top 5 popular terms in these communities (see Section 3.2).
- **T2**: Mouse over timeline to expand several consecutive snapshots of the network and the relationships of collocated terms (see Section 3.3).
- **T3**: Filter terms/ topics on user request (see Section 3.3). For example, users may want to see political events at a specific geographic area (see Section 3.5).
- **T4**: Sort terms based on a selected measure: *term frequency*, *sudden increase in frequency*, *vertex degree*, or *betweenness centrality* (see Section 3.4).

3.1 Extract terms

The input text documents are preprocessed into entities and further classified into different categories: people, locations, organizations, and others. We use colors to encode these categories: red for person, green for location, blue for organization, and orange for miscellaneous. This color-encoding is used consistently for all figures in the paper.

3.2 Construct networks for each time stamp

There are several methods to generate relationships among two given text element entities. For instance, a link can be generated by the similarity between two documents, or the frequency with which two entities are mentioned together [16], or defined logical forms[19]. The weight for each link is calculated accordingly. Since *DycomDetector* works directly on individual terms to obtain community-based coherence, the relationship is determined based on the collocation of the terms in the same articles/blogs. A link with high weight indicates that two terms are frequently situated together in a given period of time, such as within one day or within one month.

3.3 Redefine the vertices

The *DycomDetector* allows users to narrow down the text network using different parameters (visualization task **T3**). In particular, users can redefine vertices in the relationship network using several properties which are divided into non-network properties (including *term frequency* and *sudden increase in frequency*) and network-related properties (including *vertex degree* and *betweenness centrality*). Depending on the user selection, *DycomDetector* recalculates the networks and their communities' formulations accordingly.

3.4 Generate visualizations

In *DycomDetector* visualization, we align network snapshots horizontally from left to right. This design is widely adopted in many time series visualizations [8, 9, 26]. A histogram below each network displays modularity Q on different filtering settings (called *relationship cut*). *Relationship cut* can extend from 1 to the highest weight of vetices in all terms networks.

The term timelines are shown at the bottom. In particular, *CloudLines* style visualization [20, 22] can be overlaid to highlight the evolution of terms over time. Arcs are used to connect collocated terms which are ordered by month. Terms in the same month are ordered by communities to reduce edge-crossings (visualization task **T4**) since the community detection algorithm groups highly connected nodes (frequently collocated terms) and loosely connected nodes in other clusters. The quality of produced cluster formation is reflected in modularity Q presented in histograms above.

3.5 Interactions

DycomDetector supports a range of interactions, such as lensing or filtering (based on the four features above), allowing users to quickly drill down events of interest. Moreover, users can input into a search box to provide a topic of interest (visualization task **T3**). Figure 2 shows an example of topics related to the inputted geographic location "Tucson". Data is political blogs from the *Huffington Post* which contain 75,293 blogs, and we extracted 33,528 terms in total. Notice that the networks and their cluster formations (number of clusters as well as members in each cluster) are very different for different months. Consequently, their modularity histograms vary significantly. In the example, only the second bar in each histogram is highlighted since users have set the *relationship cut* to 2.

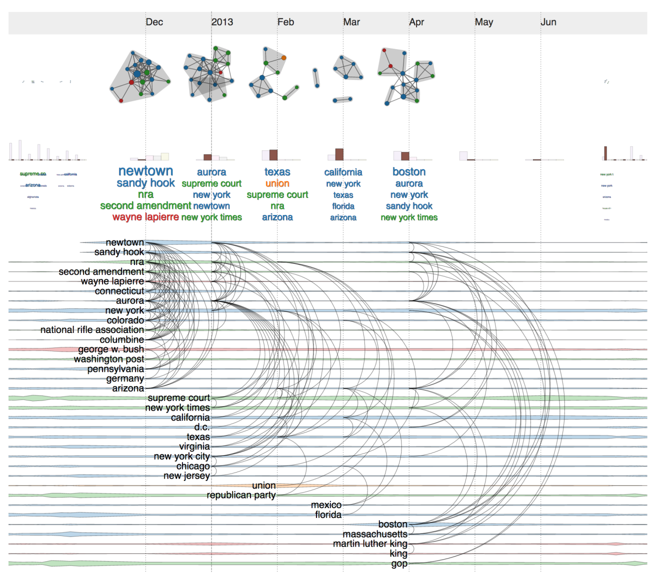


Figure 2: Related topics in the *Huffington Post* for an input term "Tucson". Terms in the list are ordered by *betweenness centrality*. For example, *Newtown* and *Sandy Hook* are the two central terms appearing on many political blogs due to the shooting event at the Sandy Hook Elementary School shooting which occurred on December 14, 2012, in Newtown, Connecticut [28].

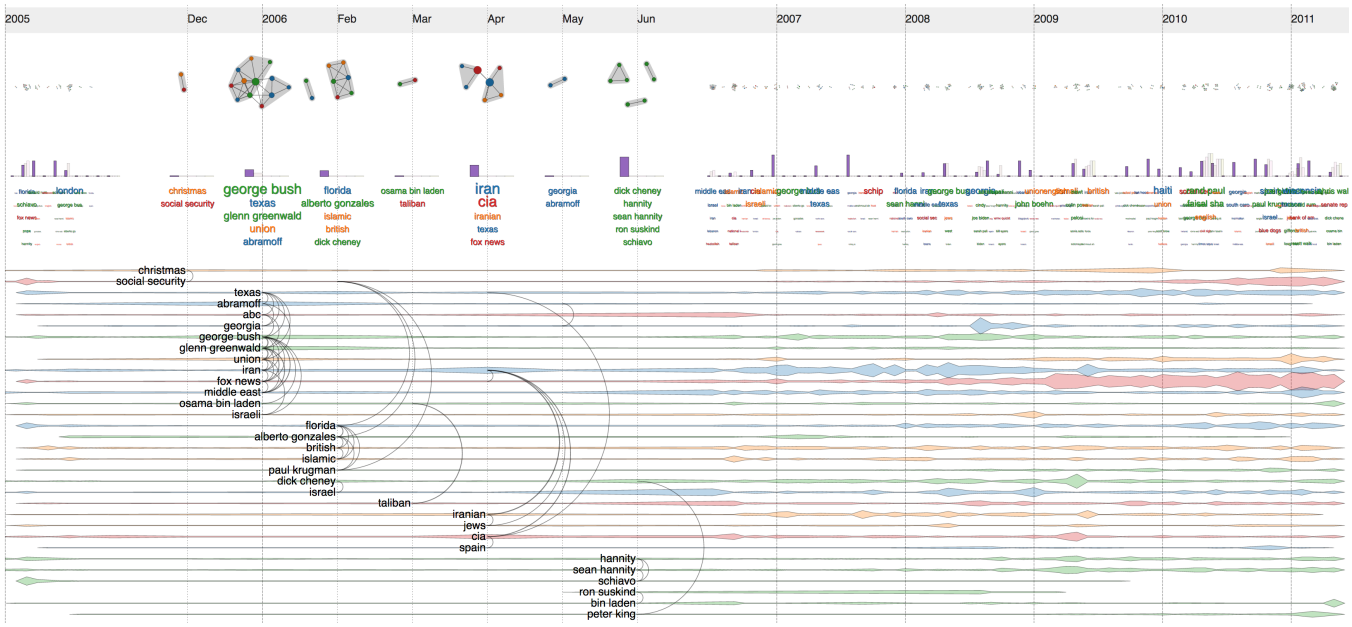


Figure 3: *DycomDetector* on Crooks And Liars blogs with lensing into 2006. The top 5 terms under modularity histograms are ordered by *betweenness centrality*. For example, *iran* and *cia* are the hottest terms in April 2006.

Since it is difficult to explain the interactions with *DycomDetector* from static images, we advise the viewers to conduct the demo video on our *DycomDetector* project page.

4 EXPERIMENTS

4.1 Datasets and use cases

We will illustrate the features of *DycomDetector* mainly through examples. We use datasets retrieved from different political blogs, such as Americablog, Crooks and Liars, the Huffington Post, and other sources to demonstrate the performance of *DycomDetector*.

Figure 3 shows an example of *DycomDetector* on Crooks And Liars blogs (which contains 9,663 blogs and 4,935 terms) while Figure 4 uses the Ensquire data (which contains 2,208 blogs and 2,117 terms). The figures depict lensing effects at different time stamps but with the same settings: terms are selected and ranked based on *net frequency* while connections are filtered by *relationship cut* to maximize the modularity scores on each network snapshot. In Figure 4, notice that the nodes have different sizes depending on their *betweenness centrality*. Moreover, the best *relationship cut* for revealing network structure is different (highlighted in different bar colors underneath each network).

More examples and other use cases are provided on our project, available at https://github.com/iDVLTTU/DycomDetector.

4.2 Comparisons with LDA

In order to validate the performance of our proposed prototype, we make a comparison between *DycomDetector* and a well-known LDA model [3]. The result in Figure 5 shows that our proposed *DycomDetector* model generates some overlapping terms with the LDA model. For example, in July 2014 some overlapping terms are *hamas*, *gaza*, and *strip*. As depicted, it is easier to keep track of the evolution of terms over time in *DycomDetector*. Notice that the term *russia* mentioned most in March 2014 (highest frequency value) does not show up in the top ranking words in the next two months then is mentioned again in June and becomes a hot topic yet again in September.

Figure 5(c) allows users to look at the filtered topics at a different angle. The network is constructed by *sudden change* in terms frequency. *Betweenness centrality* is applied to highlight important terms (terms serve as bridges in these networks). Notice that node sizes are computed to reflect their connecting roles (*betweenness centrality*).

4.3 Implementation

DycomDetector is implemented in D3.js [5]. The application, source code, sample data, and demo video are provided via our GitHub project repository, located at https://github.com/iDataVisualizationLab/DycomDetector.

5 CONCLUSION

In this paper, we present a novel approach that incorporates a community detection algorithm to find topics and reveal network structure in temporal data automatically. We also introduce an interactive data analytics prototype which helps users to visualize and analyze topic abstractions and how they change over time. Our experiments on various datasets show that *DycomDetector* provides a better lens into large corpus of texts obtained from news/blogs. Furthermore, our study demonstrates the usefulness of each component of our *DycomDetector* model. Our model also supports a wide range of capabilities such as dynamic clustering

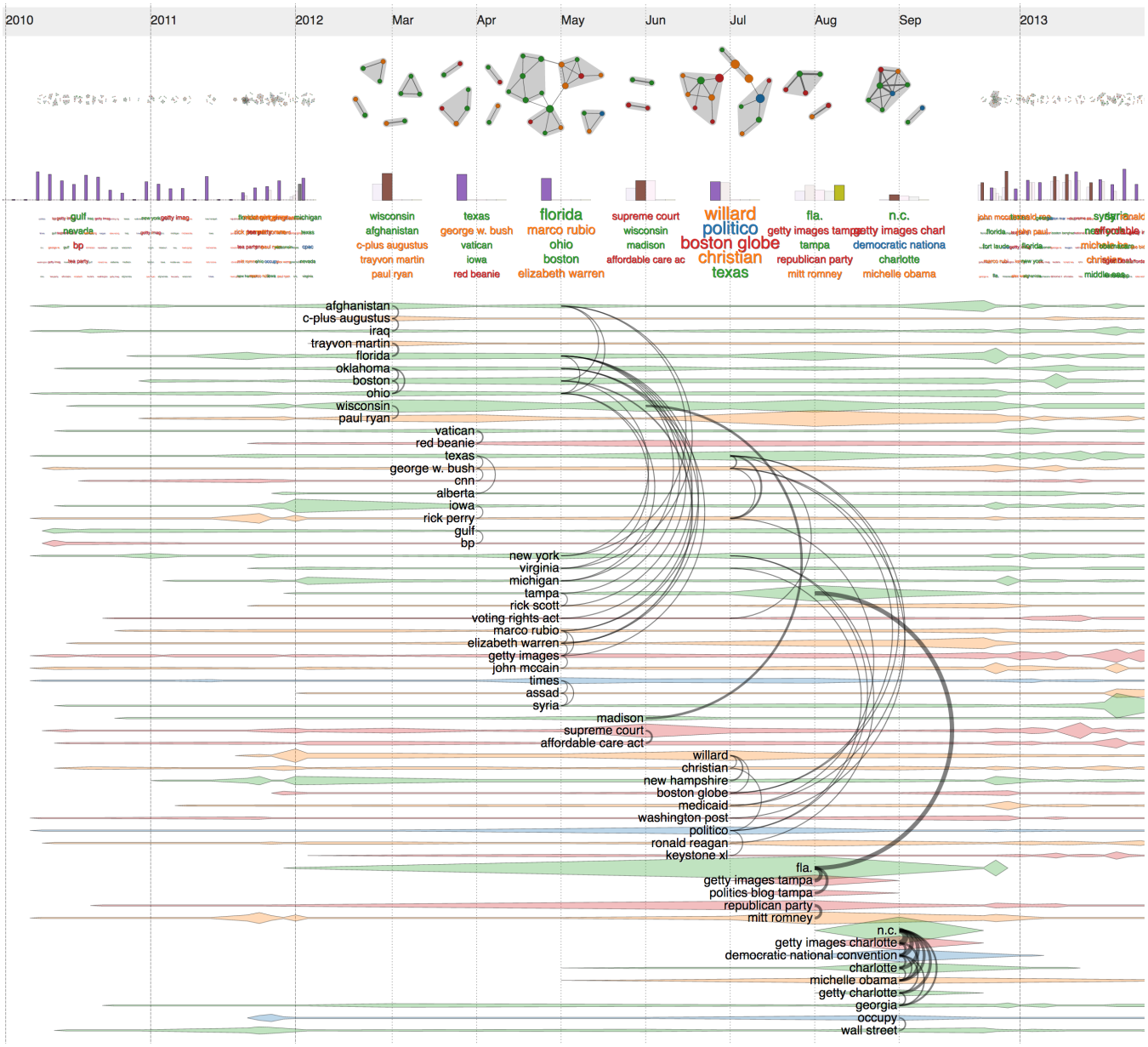


Figure 4: *DycomDetector* on Ensquire blogs with lensing into 2012. The top 5 terms under modularity histograms are ordered by *betweenness centrality*. For example, *willard*, *politico* and *boston globe* are the hottest terms in July 2012.

community over time and providing additional informative filtering selection.

In future work, we are planning to introduce topic recommendations into our prototype (the complete inferred sentence extracted from clustered topics). This new direction is very promising since it helps users gain a better understanding of given topics from an extensive collection of text documents. Moreover, applying community detection into dynamic network to automate the process of revealing network structures has many applications in other domains where tracking the structural changes is a vital part [11, 24].

REFERENCES

[1] Benjamin Bach, Emmanuel Pietriga, and Jean-Daniel Fekete. 2014. Visualizing Dynamic Networks with Matrix Cubes. In *Proc. ACM Conf. on Human Factors in Computing Systems*. 877–886.

[2] Fabian Beck, Michael Burch, Stephan Diehl, and Daniel Weiskopf. 2016. A taxonomy and survey of dynamic graph visualization. In *Computer Graphics Forum*. Wiley Online Library.

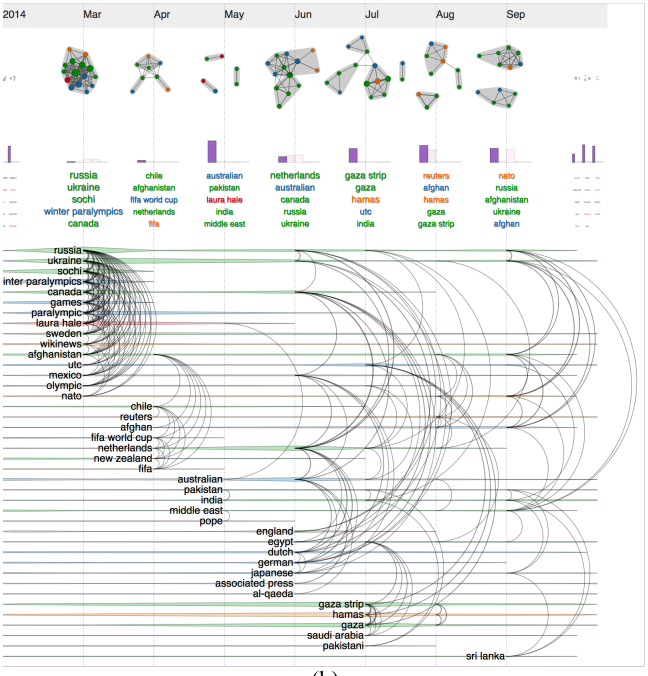
[3] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[4] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment* 2008, 10 (2008), P10008.

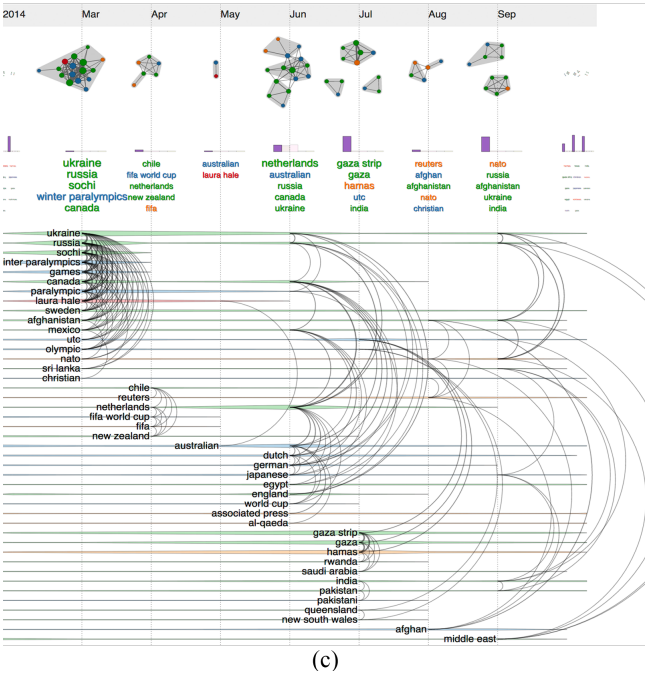
[5] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. 2011. D3 Data-Driven Documents. *IEEE Trans. Vis. Comput. Graph.* 17, 12 (2011), 2301–2309.

Mar	Apr		May	Jun			Jul			Aug		Sep	
Topic Term/Freq	Topic Term/Freq		Topic Term/Freq	Topic Term/Freq			Topic Term/Freq			Topic Term/Freq		Topic Term/Freq	
1	1	2	1	1	2	3	1	2	3	1	2	1	2
russia	police	species	gillett	governme nt	canada	team	israeli	coaching	university	center	forces	students	university
united	team	wikipedia	patrick	gold	united	ndash	gaza	basketball	stage	mcneil	reported	police	nato
hale	black	race	university	people	game	world	hamas	kyle	try	brian	military	people	state
team	macdo nald	genus	brian	speech	australia	netherland s	border	started	ozone	thumb	asteroid	noosa	final
mdash	united	stumira	whale	record	party	points	military	tom	minutes	street	security	paisley	toads
senator	univers ity	committee	mcneil	men's	australian	play	israel	coach	assistant	university	state	minister	afghanistan
laura	people	school	protesters	col	national	germany	strip	i'd	noosa	festival	passport	adams	president

(a)



(b)



(c)

Figure 5: A comparison between *DycomDetector* and LDA model based on frequency with *relationship cut* = 1 and data from March to September in 2014: a) topics obtained by LDA model b) topics generated by *DycomDetector* model based on *sudden changes* in term frequency c) topics generated by *DycomDetector* model based on *betweenness centrality*.

[6] U. Brandes and S. R. Corman. 2002. Visual unrolling of network evolution and the analysis of dynamic discourse. In *IEEE Symp. on Information Visualization*. 145–151.

[7] Ulrik Brandes and Bobo Nick. 2011. Asymmetric Relations in Longitudinal Social Networks. *IEEE Trans. Vis. Comput. Graph.* 17, 12 (2011), 2283–2290. https://doi.org/10.1109/TVCG.2011.169

[8] Lee Byron and Martin Wattenberg. 2008. Stacked Graphs – Geometry & Aesthetics. *IEEE Trans. Vis. Comput. Graph.* 14, 6 (2008), 1245–1252. https://doi.org/10.1109/TVCG.2008.166

[9] Tuan Nhon Dang, A. Anand, and L. Wilkinson. 2013. TimeSeer: Scagnostics for High-Dimensional Time Series. *IEEE Trans. Vis. Comput. Graph.* 19, 3 (March 2013), 470–483. https://doi.org/10.1109/TVCG.2012.128

[10] Tuan Nhon Dang, Paul Murray, and Angus G. Forbes. 2015. PathwayMatrix: Visualizing Binary Relationships between Proteins in Biological Pathways. *BMC Proceedings* 9, 6 (2015), S3.

[11] Tuan Nhon Dang, Nick Pendar, and Angus G. Forbes. 2016. TimeArcs: Visualizing Fluctuations in Dynamic Networks. *Computer Graphics Forum* (2016). https://doi.org/10.1111/cgf.12882

[12] Wenwen Dou, Xiaoyu Wang, Remco Chang, and William Ribarsky. 2011. Paralleltopics: A probabilistic approach to exploring document collections. In *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*. IEEE, 231–240.

[13] Gabriel Doyle and Charles Elkan. 2009. Financial topic models. In *Working Notes of the NIPS-2009 Workshop on Applications for Topic Models: Text and Beyond Workshop*.

[14] Mohammad Ghoniem, Jean-Daniel Fekete, and Philippe Castagliola. 2005. On the Readability of Graphs Using Node-link and Matrix-based Representations: A Controlled Experiment and Statistical Analysis. *Inf. Vis.* 4, 2 (2005), 114–135. https://doi.org/10.1057/palgrave.ivs.9500092

[15] N. Henry and J. d. Fekete. 2006. MatrixExplorer: A Dual-Representation System to Explore Social Networks. *IEEE Trans. Vis. Comput. Graph.* 12, 5 (2006), 677–684. https://doi.org/10.1109/TVCG.2006.160

[16] Fang Jin, Rupinder Paul Khandpur, Nathan Self, Edward Dougherty, Sheng Guo, Feng Chen, B Aditya Prakash, and Naren Ramakrishnan. 2014. Modeling mass protest adoption in social network communities using geometric brownian motion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1660–1669.

[17] Daniel A. Keim, Christian Panse, and Mike Sips. 2004. Information Visualization : Scope, Techniques and Opportunities for Geovisualization. In *Exploring Geovisualization*, J. Dykes (Ed.). Elsevier, Oxford, 1–17.

[18] René Keller, Claudia M. Eckert, and P. John Clarkson. 2006. Matrices or Node-link Diagrams: Which Visual Representation is Better for Visualising Connectivity Models? *Inf. Vis.* 5, 1 (2006), 62–76. https://doi.org/10.1057/palgrave.ivs.9500116

[19] Stanley Kok and Pedro Domingos. 2008. Extracting semantic networks from text via relational clustering. *Machine Learning and Knowledge Discovery in Databases* (2008), 624–639.

[20] M. Krstajic, E. Bertini, and D.A. Keim. 2011. CloudLines: Compact Display of Event Episodes in Multiple Time-Series. *IEEE Trans. Vis. Comput. Graph.* 17, 12 (2011), 2432–2439.

[21] Lisa Lowe. 2016. Socialpilot. https://socialpilot.co/blog/125-amazing-social-media-statistics-know-2016/. (March 2016).

[22] Dongning Luo, Jing Yang, Milos Krstajic, William Ribarsky, and Daniel Keim. 2012. Eventriver: Visually exploring text collections with temporal references. *IEEE Trans. Vis. Comput. Graph.* 18, 1 (2012), 93–105.

[23] Chihua Ma, Robert V. Kenyon, Angus G. Forbes, Tanya Berger-Wolf, Bernard J. Slater, and Daniel A. Llano. 2015. Visualizing Dynamic Brain Networks Using an Animated Dual-Representation. In *Proc. Eurographics Conf. on Visualization*. 73–77.

[24] Chayant Tantipathananandh and Tanya Y Berger-Wolf. 2011. Finding communities in dynamic social networks. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*. IEEE, 1236–1241.

[25] Xiaogang Wang and Eric Grimson. 2008. Spatial latent dirichlet allocation. In *Advances in neural information processing systems*. 1577–1584.

[26] Martin Wattenberg. 2005. Baby Names, Visualization, and Social Data Analysis. In *Proc. IEEE Symp. on Information Visualization*. 1–7.

[27] Furu Wei, Shixia Liu, Yangqiu Song, Shimei Pan, Michelle X Zhou, Weihong Qian, Lei Shi, Li Tan, and Qiang Zhang. 2010. Tiara: a visual exploratory text analytic system. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 153–162.

[28] wikipedia. 2017. Sandy Hook Elementary School shooting. (2017). https://en.wikipedia.org/wiki/Sandy_Hook_Elementary_School_shooting

[29] Ji Soo Yi, Niklas Elmqvist, and Lee Seungyoon. 2010. TimeMatrix: Analyzing Temporal Social Networks Using Interactive Matrix-Based Visualizations. *Int. J. Hum. Comput. Int.* 26, 11-12 (2010), 1031–1051.

[30] Chunyao Zou and Daqing Hou. 2014. LDA analyzer: A tool for exploring topic models. In *Software Maintenance and Evolution (ICSME), 2014 IEEE International Conference on*. IEEE, 593–596.