TopExNet: Entity-centric Network Topic Exploration in News Streams

Andreas Spitz
Heidelberg University
Heidelberg, Germany
spitz@informatik.uni-heidelberg.de

Satya Almasian
Heidelberg University
Heidelberg, Germany
almasian@stud.uni-heidelberg.de

Michael Gertz
Heidelberg University
Heidelberg, Germany
gertz@informatik.uni-heidelberg.de

ABSTRACT

The recent introduction of entity-centric implicit network representations of unstructured text offers novel ways for exploring entity relations in document collections and streams efficiently and interactively. Here, we present TopExNet as a tool for exploring entity-centric network topics in streams of news articles. The application is available as a web service at https://topexnet.ifi.uni-heidelberg.de.

ACM Reference Format:
Andreas Spitz, Satya Almasian, and Michael Gertz. 2019. TopExNet: Entity-centric Network Topic Exploration in News Streams. In *The Twelfth ACM International Conference on Web Search and Data Mining (WSDM '19), February 11–15, 2019, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3289600.3290619

1 INTRODUCTION

Keeping up with the news is a common idiom that is increasingly describing a race that human readers cannot hope to win. Since the publication of news has all but shifted from traditional print media to a rapid stream of online news, we are faced with a constant deluge of news information from the global news cycle. Finding relevant information can be such a daunting task that many users resort to reading nothing but headlines, while news publishers advertise for their articles with prominently displayed reading times of as few minutes as possible. As a result, the larger context is often lost.

An automated aggregation of news is thus clearly beneficial, yet no less daunting from a computational perspective. While much research has been devoted to techniques for finding and linking incidents in news [7], such an approach is far from trivial and too restrictive in purely exploratory settings. Intuitively, topic models [2] should offer a solid solution to the extraction of relevant topics from collections of documents. However, their performance tends to suffer on large collections of news articles with a multitude of diverse topics, and they are ill-suited for the interactive exploration of documents. Furthermore, topics are usually represented as ranked lists of words, which can be difficult to interpret [5].

In this respect, a recent shift in focus towards network-centric representations of documents stands to provide more intuitive and more *visual* access to the complex relations contained in the texts. Examples include the use of concept maps as summaries instead of text snippets [6], or the network-centric view on entities as stitching points between interwoven news streams [17]. Here, we focus on the extraction of topics as network structures of entities and terms [16], and on how they can be used to explore news.

Contributions. We present an application that demonstrates how implicit networks can be used to discover and expand entity-centric topics in a stream of news articles. By representing entity and term relations as the edges of a network, this approach supports the selection of news outlets and date ranges as additional degrees of freedom, while retaining query speeds that support an interactive use. In contrast to traditional topic models, this results in a more dynamic exploration of topics that can be used in place of aggregation approaches for incidents or articles, such as Google News.

2 RELATED WORK

Related work covers the areas of topic models and news exploration.

Topic models. Since the introduction of Latent Dirichlet Allocation [4], numerous variations of topic models have been proposed [2]. The majority of these approaches are based on graphical models, which are computationally expensive and ill-suited to interactive use. While efforts have been made to develop more dynamic topic models [3], it is not viable to continuously re-compute topics during the interactive exploration of news streams. Furthermore, since traditional topics are fundamentally lists of ranked words that are difficult to visualize, a dynamic exploration of evolving document collections with topic models is problematic.

Nevertheless, some applications have been presented that support a visual and interactive analysis of topics. One such example are TopicNets [8], which allow the user to view document contents within the larger scope of overarching topics. Similarly, word network topics are designed for the discovery of topic relations in short texts [19]. Unlike our approach, these applications lack a focus on entities as anchors of event descriptions in news texts.

News exploration. In contrast to topic models, some tools specifically support the exploration of news streams with a focus on entities, such as STICS [9] or EventRegistry [10]. NewsStand takes a different approach by clustering news articles geographically [18]. Further approaches include the monitoring of multilingual European news [1] and the extraction of semantic word clouds with significance analysis to obtain a quick overview over current news [14].

However, none of these tools include an exploration of topics. To fill this gap, we rely on an implicit network representation of text as used in EVELIN [15], which was designed for static document collections and with single relations between entities in mind. We improve upon this concept by adding an exploration of the more complex graph structures that are inherent to network topics.

WSDM '19, February 11–15, 2019, Melbourne, VIC, Australia

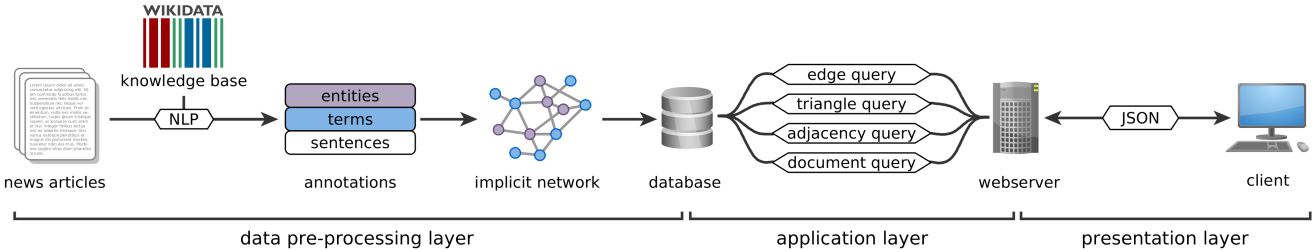

Figure 1: Schematic view of the architecture for extracting and querying implicit network topics from streams of news articles.

3 THEORETICAL BACKGROUND

We give a brief overview of the construction of network topics. For a more detailed description, we refer to our previous work [16].

3.1 Implicit Networks

Conceptually, an implicit network can be viewed as a word cooccurrence graph in which (1) entities are linked to a knowledge base, (2) long-range cooccurrences beyond sentence boundaries are considered, and (3) edge weights are derived from cooccurrence distances instead of counts. Individual edges are then aggregated over all documents to create a network representation, which may include sentences and documents as nodes. In a streaming setting, it can be sensible to partially aggregate edges in preprocessing [17].

In the following, we consider the network to be a list of edge tuples $e = \langle v, w, t, out, d, \delta \rangle$, where v and w are two nodes (entities or terms), t is the publication date, out is the news outlet, d is the document, and δ is the minimum textual distance between the two nodes in the document (measured in sentences). To support efficient queries over varying date ranges and selections of news outlets, we partially aggregate edges to at most one edge per entity pair, publication day, and outlet. Node statistics such as the occurrences in documents are partially aggregated in a similar fashion.

3.2 Network Topic Extraction

Motivated by the important role that entities play in news events, important edges between entities are considered as *topic seeds*, around which a shell of descriptive terms is constructed. If terms are ranked according to their importance for an edge, each such subgraph can be regarded as a ranked list of terms, similar to traditional topics, yet more visual. To discover the most important seed edges and select relevant terms, we thus require edge weights to rank the relations. By including the date range, we use a weighting scheme with three components that are combined as the harmonic mean. To derive the weight of an edge $e = (v, w)$ between nodes v and w in a date range $t = (t_1, t_2)$, let the score ω be

$$\omega(e,t) = 3\left[\frac{|D_v \cup D_w|}{|D_e|} + \frac{t_1 - t_2}{|T_e|} + \frac{D_{max}}{\Delta_e}\right]^{-1}$$

where D_v, D_w, and D_e denote the sets of documents in which v, w, and e occur, T_e is the set of days on which e is mentioned, D_{max} is the maximum number of documents any edge is mentioned in, and $\Delta_e = \sum_e \exp(-\delta_e)$ is the sum of decaying reciprocal distances.

Based on this scoring function, it is then a simple matter to select edges that correspond to important cooccurrences. In the application, entity edges can be instantiated either by selecting the globally highest ranked edges for a time interval and set of outlets, or by directly specifying pairs of entities that are of interest to the user. In either case, descriptive terms are added by ranking them according to their importance for the entities of each edge, thereby inducing triangular subgraphs. If edges of distinct subgraphs overlap in an entity, they can be merged into a larger topic subgraph.

4 SYSTEM ARCHITECTURE

In the following, we describe the system architecture for data preprocessing and network topic extraction as shown in Figure 1.

4.1 Data Preprocessing

Implicit networks can be extracted from any document that is annotated for entities. Since the cooccurrences in each document correspond to a small network and networks are additive, document streams can be iteratively composed into a larger network that represents the entire stream history as the documents arrive.

Document preprocessing includes part-of-speech and sentence tagging, named entity recognition and linking, and entity classification. Stanford CoreNLP [11] is used for sentence splitting and part-of-speech tagging. For the recognition and disambiguation of named entities to Wikidata IDs, we use Ambiverse[1]. To identify named entities of type actor, location, and organization, we map Wikidata IDs to YAGO3 entities [13] and classify them according to the YAGO hierarchy by using class `wordnet_person_100007846` for actors, class `wordnet_social_group_107950920` for organizations, and `yagoGeoEntity` for locations. Finally, all entities are augmented with Wikidata descriptions. Remaining tokens that are at least four characters long constitute the set of terms and are stemmed with a Porter stemmer [12]. The implicit network is constructed from the annotated data as outlined in Section 3. For entity cooccurrences, we set the window size to 5 sentences, and use intra-sentence occurrences for terms and entities.

4.2 Application Layer

While an in-memory representation of the data is possible and supports fast query processing, it does not scale arbitrarily and is not feasible for a long-running non-commercial demonstration. Therefore, we use a Core i7 with 32GB main memory and an SSD drive as demonstration server. The network is stored in a MongoDB, with separate collections for entities, terms, edges between entities, and edges between entities and terms. Entities are enriched with Wikidata information to provide entity descriptions at query time. Based on input strings, a text index on the English canonical label

[1] https://www.ambiverse.com/

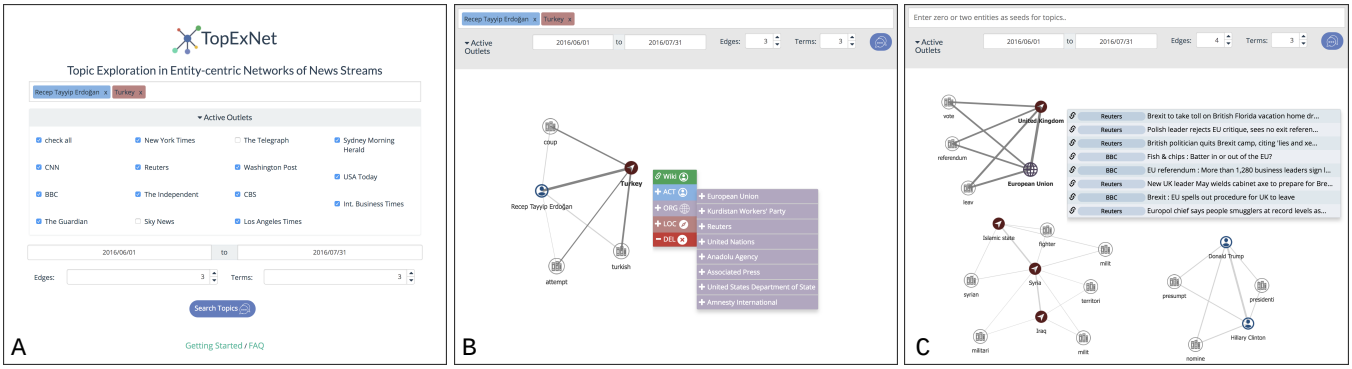

Figure 2: Overview of TopExNet's user interface. A: initial search page with selectable parameters. B: result of a targeted entity edge query with an entity exploration menu. C: result of a global edge ranking query with an article recommendation menu.

is used to compile a list of entity suggestions for the user. We rank entity suggestions by the text match score and break ties by the number of occurrences. All edges are partially aggregated at a granularity level of days to speed up subsequent aggregations at query time. Node occurrence information is stored in a similar aggregation to retrieve individual occurrence counts in documents.

The interactive topic extraction methods described in Section 3 are implemented in Java and enable query processing speeds in the order of a few seconds for all but the most highly connected entities. Like most other complex networks, implicit networks have a long-tailed degree distribution, which translates to the presence of few highly connected hubs in practice. While queries on the network generally benefit from the overall sparseness, hubs may cause longer query response times for incident edges, especially for large date ranges. However, due to their small number, this problem can be addressed by caching results in the database, which ameliorates the effect over time. Specifically, we use a separate collection for caching the results of individual triangular term expansion queries that then serve as building blocks of later queries.

While topic extraction queries can be parallelized by edge, we only allocate one thread per query to serve queries from multiple users simultaneously. To avoid system overload in the case of multiple users, we use an anonymized mapping of queries to browser fingerprints and limit the number of active queries per user.

4.3 Presentation Layer

The web interface is implemented in HTML and JavaScript, and accepts user input to extract suitable topics and visualize the output as graphs. For entity input and for sending queries to the application layer, we use jQuery. The Bootstrap libraries[2] and Mustache web templates enable the interactive layout. To recognize and classify input entities, we use the tags-input and typeahead libraries of Bootstrap, which we extend by adding the required functionality for the color coding of entities. The interactive visualization of the topic networks and the menus is handled by the vis.js JavaScript library[3]. Graphs are visualized with a force-directed layout.

The web server itself uses the Java Spark micro framework[4] and is directly integrated with the application layer. Communication between user interface and server is built on AJAX and uses JSON for transmitting data in both directions (i.e, input query entities, output graph data). Examples of the interface are shown in Figure 2, based on which we discuss the functionality in the following.

5 FUNCTIONALITY AND DEMONSTRATION

We briefly describe the data used in the demonstration, before discussing TopExNet's functionality and usage scenarios.

5.1 News Network Data

As data for the presentation, we collect articles from the RSS feeds of international news outlets with a focus on politics. The content is extracted with manually created rules to include multi-page articles and avoid the drawbacks of boilerplate removal. Specifically, we use articles from 14 English speaking news outlets located in the U.S. (CNN, LA Times, NY Times, USA Today, CBS News, The Washington Post, IBTimes), Great Britain (BBC, The Independent, Reuters, SkyNews, The Telegraph, The Guardian), and Australia (Sydney Morning Herald) during the period from June 1 to November 30, 2016. We remove articles that have less than 200 or over 20,000 characters (due to limitations in the NER step) or more than 100 disambiguated entities per article (i.e., lists). The final collection contains 127,485 articles with a total of 5.4M sentences. After preprocessing as described in Section 4, the resulting network has 27.7K locations, 72.0K actors, 19.6K organizations, and 322K terms, which are connected by 26.8M partially aggregated edges.

5.2 Input Parameters

User input for queries to the data is based on the four parameters *entities*, *date range*, the *number of edges* and the *number of terms*. Additionally, a subset of news outlets can be selected.

Entities are entered as input by selecting them from a list of suggestions that is generated from one or several strings provided by the user. Entity suggestions are automatically linked to network nodes upon selection. A *date range* is chosen by using a date range picker with a granularity of days, and is limited to the publication time frame of articles in the stored stream. The *number of edges* can be used to set the number of seed edges when global edge ranking serves as a starting point. Similarly, the *number of terms* that are extracted for each seed edge can be adjusted. An overview over the initial input screen is shown in Figure 2A.

[2]http://getbootstrap.com
[3]http://visjs.org/
[4]http://sparkjava.com

5.3 Exploration Approaches

TopExNet offers four primary modes of exploring network topics and the underlying news stream as we describe in the following.

Global edge ranking. The first primary use-case is the automatic extraction of seed edges from the network. If the user specifies a date range but no input entities, then the output is a global ranking of all entity edges for the specified date interval and news outlets. The highest ranked edges are selected as topic seeds, merged if they share some of their nodes, and expanded by adding descriptive terms. An example of the output for three edges and three terms per edge is shown in Figure 2C. For a larger number of edges and a limited time frame, this serves as a birds-eye view on current news.

Targeted entity exploration. In contrast to the global ranking of edges, which focusses on the topics surrounding the entities that are the overall most central during the selected time frame, the user may also focus on specific entities. When supplied with two entities as query input, TopExNet adds descriptive terms only to the edge between the two provided entities. An example is shown in Figure 2B. While such individual seed edges naturally generate smaller topic networks, they serve as selectable starting points and can be further expanded by adjacency exploration.

Topic network exploration. The network topics from the above two cases support further exploration. In addition to the obvious tuning by increasing or decreasing the number of seed edges or descriptive terms, the user may also expand the displayed network. When selecting any entity in the network, the user is given the choice to include highest ranked adjacent nodes. Here, we rely on the entity ranking method that was introduced for EVELIN [15], and adapt it to semi-aggregated edges. A visualization of the process is shown in Figure 2B. Once additional entity edges are added, descriptive terms may be included by clicking on edges or the canvas and selecting the option to add terms. Likewise, nodes that are not of interest can be deleted, or information about entities can be obtained by opening the corresponding Wikidata pages.

All three of the above approaches support a facetted and contrastive analysis of network topics. By viewing time slices or topics for subsets of outlets in parallel windows, the user can compare network topics and their evolution between different sources.

Article recommendation. Finally, once the user has identified topics of interest, TopExNet can recommend suitable news articles that describe the selection in-depth. When *multiple* entities and/or terms are selected, a right-click opens a menu of article recommendations. Each recommendation links to an original news article that is relevant to the selected nodes. Thus, the exploration and identification of network topics can serve as an entry point for the focused reading or analysis of related news articles.

6 SUMMARY AND OUTLOOK

In this demonstration, we presented TopExNet as a web-based application for the exploration of network topics in news streams. By leveraging implicit entity network representations of the underlying document stream, we demonstrated the feasibility of exploring entity-centric topics interactively, even for large document collections and in a streaming setting. On the technical side, the partial aggregation of entity and term cooccurrence edges allows the efficient retrieval of both seed edges and descriptive terms from the data, without the costly requirement of re-computing topics for the entire corpus due to changing parameters, thus making the extraction of topics truly dynamic. On the application side, the network representation of topics aggregates content without the need to display the content of potentially proprietary news articles to the user, and thus serves as a valuable alternative to existing news aggregation and summarization approaches in industrial settings.

Future work. For this demonstration, we presented TopExNet as a standalone application. However, the underlying network structure is similar to the data representation used for EVELIN [15], meaning that topic exploration can be integrated seamlessly. To ensure scalability with an increasing number of news outlets, we are considering a replication of the data in clustered database servers to benefit from the parallel nature of individual edge queries.

REFERENCES

[1] Martin Atkinson and Erik Van der Goot. 2009. Near Real Time Information Mining in Multilingual News. In *WWW*. https://doi.org/10.1145/1526709.1526903

[2] David M. Blei. 2012. Probabilistic Topic Models. *Commun. ACM* 55, 4 (April 2012), 77–84. https://doi.org/10.1145/2133806.2133826

[3] David M. Blei and John D. Lafferty. 2006. Dynamic Topic Models. In *ICML*. https://doi.org/10.1145/1143844.1143859

[4] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *J. Mach. Learn. Res.* 3 (March 2003), 993–1022. http://dl.acm.org/citation.cfm?id=944919.944937

[5] Jonathan Chang, Jordan L. Boyd-Graber, Sean Gerrish, Chong Wang, and David M. Blei. 2009. Reading Tea Leaves: How Humans Interpret Topic Models. In *NIPS*. http://papers.nips.cc/paper/3700-reading-tea-leaves-how-humans-interpret-topic-models

[6] Tobias Falke and Iryna Gurevych. 2017. Bringing Structure into Summaries: Crowdsourcing a Benchmark Corpus of Concept Maps. In *EMNLP*. https://aclanthology.info/papers/D17-1320/d17-1320

[7] Ao Feng and James Allan. 2007. Finding and Linking Incidents in News. In *CIKM*. https://doi.org/10.1145/1321440.1321554

[8] Brynjar Gretarsson, John O'Donovan, Svetlin Bostandjiev, Tobias Höllerer, Arthur U. Asuncion, David Newman, and Padhraic Smyth. 2012. TopicNets: Visual Analysis of Large Text Corpora with Topic Modeling. *ACM TIST* 3, 2 (2012), 23:1–23:26. https://doi.org/10.1145/2089094.2089099

[9] Johannes Hoffart, Dragan Milchevski, and Gerhard Weikum. 2014. STICS: Searching with Strings, Things, and Cats. In *SIGIR*. https://doi.org/10.1145/2600428.2611177

[10] Gregor Leban, Blaz Fortuna, Janez Brank, and Marko Grobelnik. 2014. Event Registry: Learning About World Events from News. In *WWW Companion*. https://doi.org/10.1145/2567948.2577024

[11] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *ACL System Demonstrations*. http://aclweb.org/anthology/P/P14/P14-5010.pdf

[12] Martin F. Porter. 1980. An Algorithm for Suffix Stripping. *Program* 14, 3 (1980), 130–137. https://doi.org/10.1108/eb046814

[13] Thomas Rebele, Fabian M. Suchanek, Johannes Hoffart, Joanna Biega, Erdal Kuzey, and Gerhard Weikum. 2016. YAGO: A Multilingual Knowledge Base from Wikipedia, Wordnet, and Geonames. In *ISWC*. https://doi.org/10.1007/978-3-319-46547-0_19

[14] Erich Schubert, Andreas Spitz, and Michael Gertz. 2018. Exploring Significant Interactions in Live News. In *NewsIR*. http://ceur-ws.org/Vol-2079/paper9.pdf

[15] Andreas Spitz, Satya Almasian, and Michael Gertz. 2017. EVELIN: Exploration of Event and Entity Links in Implicit Networks. In *WWW Companion*. https://doi.org/10.1145/3041021.3054721

[16] Andreas Spitz and Michael Gertz. 2018. Entity-Centric Topic Extraction and Exploration: A Network-Based Approach. In *ECIR*. https://doi.org/10.1007/978-3-319-76941-7_1

[17] Andreas Spitz and Michael Gertz. 2018. Exploring Entity-centric Networks in Entangled News Streams. In *WWW Companion*. https://doi.org/10.1145/3184558.3188726

[18] Benjamin E. Teitler, Michael D. Lieberman, Daniele Panozzo, Jagan Sankaranarayanan, Hanan Samet, and Jon Sperling. 2008. NewsStand: A New View on News. In *ACM-GIS*. https://doi.org/10.1145/1463434.1463458

[19] Yuan Zuo, Jichang Zhao, and Ke Xu. 2016. Word Network Topic Model: A Simple But General Solution for Short and Imbalanced Texts. *Knowl. Inf. Syst.* 48, 2 (2016), 379–398. https://doi.org/10.1007/s10115-015-0882-z