TopicFlow: Visualizing Topic Alignment of Twitter Data over Time

Alison Smith
Jianyu Li
Panagis (Pano) Papadatos
Sana Malik

CMSC734 - Information Visualization
November 30, 2012

ABSTRACT

The analysis and visualization of social media [1] data is becoming an increasingly popular topic in the areas of Natural Language Processing and HCI. As of October 2011, Twitter users were producing approximately 250 million tweets per day [19]. This overwhelming amount of data is far beyond what any user could possibly read and understand without the help of analysis and visualization tools.

In this paper, we present two main contributions. The first, was to expand on the “Aggregate Overall Metrics Feature” in NodeXL to allow aggregation of summary statistics (such as word and word pair frequencies) across multiple workbooks into a single workbook. The second task was to develop a tool to visualize this Twitter data. In analyzing the tweets, we hope to gain insights on what users have talked about during events that occur over a larger or smaller period of time. To this end, we run an automated topic detection algorithm on the Twitter data to learn more about changes in users’ attention over time. The goal of this tool is to support user interaction and enable an analyst to easily create an overview of changing topics of interest and their relative rates of mention as well as study a dataset of this kind in depth.

INTRODUCTION

Many social media visualization tools that aim to create insights or explore relationships in Twitter data have been created. In particular, NodeXL[4] allows users to collect tweets from within the tool, and create workbooks of 1500 tweets each. However, NodeXL only allows importing one workbook at a time. Without multiple workbooks, NodeXL cannot provide significant temporal data if each workbook contains data for one day. Hence, we added a feature to allow aggregating word and word pair worksheets from multiple selected workbooks to a new workbook. In this way, the aggregated data would span across a wider time range, which allows for the creation of a line chart that provides an interesting overview.
A variety of tools exist that have been often used in visualizing social media trends on a timeline [2, 3, 7, 8, 9, 10, 17]. However, most of them visualize word, word pair, and hashtag frequencies. While these metrics are useful, they are not suitable for providing a clear view of the similarity between the various topics that are being talked about; using NodeXL made this problem apparent within our team. By studying current literature on temporal analysis of textual datasets, we did not find any tools attempting to perform topic modelling on a dataset and visualize topic similarities over time. This motivated us and led us to the idea of TopicFlow. To this end, we explored applying automatic topic modelling [6,15] by using the Latent Dirichlet Allocation (LDA) algorithm to understand the topics of tweets at a higher level.

Because we analyze the topics at discrete time slices separately, the topics of one time slice are not related to another by default. The issue of aligning topics is an open problem in NLP, and we aim to solve it through our visualization tool, TopicFlow. We were unable to find an existing temporal analysis tool that portrays the evolution of topics in a satisfactory manner. We believe that TopicFlow sufficiently meets the need for such a tool.

In this paper, we discuss related works for Twitter data analysis and their limitations and we describe our two contributions in depth. More specifically, we describe our design methodology, our interfaces, as well as the methodology and the results of our evaluation sessions.

RELATED WORK

A significant amount of work has been done in the fields of topic modeling of Twitter data and the visualization of temporal Twitter information. Conference Monitor [9] is a web-based Twitter visualization tool that was developed to study twitter data surrounding a particular conference. This tool provides a visualization for hashtag usage over time. TwitInfo [18] provides the peak of different topics containing same keywords over time. Vox Civitas [5] is a visual analytics tool that was designed to assist analysts attempting to extract important information from a large number of tweets and other social media content surrounding broadcast events. This tool visualizes Twitter data that has been analyzed with NLP techniques such as sentiment and keyword extraction. A group from Yahoo! research labs developed a tool, Statler [17], to examine Twitter data as it corresponds with broadcast events. This tool visualizes Twitter data over time and defines two new Twitter measures: chattiness and importance. TimeSearcher 1 and 2 [9, 10] are visualization tools for time-series data. However, TimeSearcher does not support easily visualizing trends between lines or analyzing the relationships between data points. Besides, it provides a sentimental analysis to tweets text contents. Visual Backchannel [2] is a dynamic visualization that allows for conversations around a topic to be observed through a multifaceted display. Nokia Internet Pulse is [19] a system for visualizing evolution of a discussion around a certain topic on Twitter with a time series of stacked tag clouds. Other tools attempt to conduct tag cloud analysis. SparkClouds [12] integrates sparklines into a tag cloud to convey trends between multiple tag clouds. Furthermore, the interactive visualization technique of “time-varying co-occurrence highlighting” which extends tag clouds in order to analyze term relations in textual content over time is presented in [20].
CONTRIBUTIONS

Aggregated Workbooks in NodeXL

NodeXL [4] is a social media analysis tool, integrated with Microsoft Excel, that aims to help create insights from social media data. Its main goal is to visualize network data in an effective way. It is very customizable and provides a wide variety of functions such as group matrix calculations and classification methods, and gives users a wide variety of visualization and interaction options. Currently, NodeXL only allows importing data from one workbook at a time.

The functionality we added is the ability to aggregate multiple workbooks and combine them into one automatically. By clicking on the “Import” section of the NodeXL ribbon and “Words and Word Pairs From the Open Workbooks” users can now select the workbooks that they wish to merge from their local file directory (Image 1).

![Import Word Pairs From Workbooks Dialog](image1.png)

Image 1

After selecting a folder containing xlsx files, a new workbook with aggregated edge, vertices, word, word and pairs worksheets is created, and this new workbook can be used for line charts over word or word pairs.

With this new feature, users could import multiple workbooks easily. Thus, we can obtain data across a time range, which allows for a line chart based on the time columns added inside the aggregated workbook.

Challenges

It took us around two weeks to add this feature to NodeXL, and it was not that easy as we expected at the beginning. The learning curve in understanding NodeXL code structures took us most of the time. In the backend, we solved problems such as how to read data, how to create a new worksheet, a new column, where to add our newly created C# classes, and how to realize the aggregating behavior when reading multiple workbooks. In the frontend, we resolve issues such as reading selected workbooks from directory one by one instead of forcing all of them to
Visualizing topics on Twitter over time: TopicFlow

Design Methodology

In designing this visualization tool, we went through multiple iterations within our group, where a lot of different options were considered, and often attempted and evaluated. While the design was still being finalized, throughout the design process we met with Cody Dunne, an information visualization expert. His input, as well input from Ben Shneiderman, the professor teaching the CMSC734 class for the Fall semester of 2012, combined with numerous brainstorming and idea evaluation driven meetings, helped us come up with an idea that we felt was worth realizing.

In order to provide a novel tool, after considering various different options for the type of visualization that we would be using we decided to use a variation of the Sankey visualization [11]. As suggested by professor Shneiderman, we felt that it would be useful to use coordinated views for the various interactions, as well as provide details of the topics, the similarity between them and the various tweets. Our interaction attempted to follow Shneiderman’s visualization mantra: [14] “Overview first, zoom and filter, then details-on-demand”.

Implementation

The TopicFlow backend was developed in Python and uses a Latent Dirichlet Allocation (LDA) algorithm implemented by Nakatini Shuyo for performing the statistical topic modeling of Twitter data (https://github.com/shuyo/iir/blob/master/lda/lda.py). The results of this processing are provided to the frontend as JSON files. The following outlines the flow of data through the interface.

1. **Data Ingestion:** The tool ingests NodeXL edge worksheets containing tweets that had been converted to the CSV format.

2. **Pre-Processing:**
   a. **binning:** The tweets are divided into some number of bins defined by an input parameter. Each bin represents a time slice of equal or almost equal length, therefore some bins contain more tweets than others (i.e. if the dataset contains more tweets of a specific time period).
   b. **stopwords:** The LDA algorithm uses a stop words list in order to remove common and noisy words from the topic modeling. This list was adapted to include common Twitter tokens. Also, due to the fact that the Twitter data was collected through keyword and hashtag search, for each dataset, the corresponding search terms were added to the stop words list.

3. **Topic Modeling:** The LDA algorithm is used for each bin separately, therefore producing a topic model for each bin where each topic model contains some number of topics defined by an input parameter. The result of topic modeling is typically a distribution of words for each topic in the topic model and a distribution of topics for each
input document. For the twitter data, it was important to also correlate tweets to each of the topics, so we took the top topic for each tweet using the maximum of P(topic|tweet) and attributed that tweet to the topic, effectively assigning P(tweet|topic).

4. **Topic Alignment:** Similarity between the topics across time bins is calculated by using a cosine similarity metric. This metric is used to compare each pair of topics from time A to time B by the probabilities of the top 20 words in the topics. Topics that share high probability words are therefore considered more similar than topics that share low probability words - usually noise. Topics are assigned a similarity value between 0 and 1, where 1 would represent the exact same topic.

5. **Output:** Finally, the output is formatted as JSON and provided to the frontend.

The TopicFlow frontend was developed in javascript, using the d3 library (https://github.com/mbostock/d3) created by Mike Bostock. In order to visualize the topic alignment over time, TopicFlow uses a novel visualization adapted from the d3 library’s “Sankey” diagram [11]. Our visualization shows the topics at each time range as nodes in the interface sized by the number of tweets attributed to the topic where edges between the nodes are weighted representations of the similarity between the topics.

**Description of the Interface**

The interface of TopicFlow consists of 6 separate but coordinated panels (Image 2):

1. The Main Visualization
2. The Topics
3. The Topic Details
4. The Tweet
5. The Tweet details
6. The Filters

The main visualization shows a Twitter dataset with automatically extracted topics over time. Each column represents tweets within a specific time slice, and each box represents a topic. The topics are sized and ranked by the number of tweets that relate to them. Topics that just
emerged appear in green, topics that evolved from and into other topics appear in blue, topics that are ending appear in pink and topics that emerged and ended within the same time slice - or standalone topics - are in orange. An edge connecting two topics signifies that they were found to be similar and the thickness of each edge is relative to the cosine similarity of the two topics. When you hover over a topic box, the words that are associated with it appear next to the cursor along with a numeric topic ID (since LDA - the algorithm used - does not allow for a more descriptive title.

Clicking on a topic (Image 3) shows which words are related to it (panels 2 and 3) and which tweets it is associated with (panel 4). Panel 3 shows the frequency of the words for that topic. In panel 1, the entire chain of similar topics is highlighted, while the unrelated ones are whited out. Clicking on the topic again deselects the topic. Clicking on the topic in panel 2 demonstrates the exact same behavior. Clicking on a tweet shows details about it and topics it is associated with, as well as the confidence in panel 5.

Clicking on an edge (Image 4) displays the words that are related to the two topics that the edge is connecting and their frequency for each topic respectively. These words are sorted by highest frequency for each topic. Words in white appear only in one of the topics, whereas words in pink appear in both.
Panel 6 allows for filtering based on topic size - with a slider widget - , topic type (emerging, continuing, ending or standalone topics with checkboxes) as well as edge weight - with a slider. Finally, the search box in panel 3 allows for filtering based on certain words (Image 5), i.e. typing a certain word highlights only topics that contain it and the similarities between them. It is possible to search for a combination of words by separating them with a space character.
EVALUATION

Subjects
Due to lack of time and resources, as well as the limited scope of this class project we performed expert reviews with 4 users over the course of two weeks. These users were Graduate students or Faculty at the university of Maryland, or professionals that did not participate in the design and development of our interactive visualization tool. However, we required that they be familiar with interactive visualizations; they have used at least one visualization for analysis before and are familiar with at least one visualization tool analyzing Twitter data. We chose users that are already familiar with visualization tools since we would like to minimize learning curves that might add confusion or introduce an exploratory processes unrelated to our tool. Since this tool mostly targets experts that are likely to have a lot of experience with such tools, we felt that expert reviews were the most appropriate way to get the type of feedback that we needed. We felt that, unlike other types of projects, our users are much more likely to have a deep understanding of interface and visualization design and therefore would be able to give us high quality feedback.

Testing Process - Summary
Our study consisted of separate sessions with different participants and different versions of the interface. Thus, our interface was developed in an iterative process where each participants comments were incorporated into our design. The study was conducted in different places due to limited availability of our participants. More specifically, two sessions were conducted in the Human-Computer Interaction Lab, using one of our laptops with our visualization tool. We used one of the large screens that are available in the lab, instead of the laptop screen, in order to demonstrate the interactive visualization in a sufficiently good resolution, to avoid issues of obscurity due to lack of space. However, one session (which included 2 participants) took place online. There was no monetary compensation, but the participants were offered snacks when they were physically present. The sessions below are separated into 2 categories (#1 and #2) since 2 different versions of the interface were used for them.

Session #1
The goal of the first session was to establish a base model for our software. In order to gauge how these experts would perceive our application, this session was vastly open ended. After explaining what was involved in the session and stating that the users’ participation was voluntary and could be withdrawn at any point, we gave our participants a very short description of what the function of our application is. More specifically, the following script was recited:

“TopicFlow is a browser based visualization tool that visualizes Twitter data over time. More specifically, we attempt to visualize topics - automatically extracted from tweets - over time. Currently, we only provide 6 sample datasets which you can visualize.”
The reason why we did not initially elaborate on the interactions with our product was so as to not impose ease of use and clarity of functionality, since our first goal was to assess the clarity of the TopicFlow goal and use cases.

After reciting the above quoted text, we gave each participant access to our visualization tool, and asked them to choose and explore one of the provided datasets. They were instructed to describe everything that they were doing and why, as well as express any other comment that they might have (think-aloud method). Their comments and our observations (mistakes they made, unreasonable learning curves, bugs, confusing interface elements, missing items etc) were documented in handwritten and typed notes, taken by the researchers present during the session. The reason why recording devices, such as cameras or audio recorders were not used was that they are often considered to be too distracting and invasive as a means of collecting data during such studies.

After our participants felt like they had fully explored our tool, they were asked to give us input regarding elements that they liked, elements that they disliked and design ideas that they felt would help improve our interface and visualization. Furthermore, our participants were asked to help us compile a list of use cases. This was especially important, since our team is not familiar with professional analysis of Twitter data and their input was very valuable.

**Results of Session 1**

The collected notes were aggregated and analyzed by our team. The results are as follows:

**Likes**

The participants appreciated the ease with which they could identify emerging, continuing and ending topics, especially due to the existence of colors. They were excited about the large number of possible interactions, as well as the functionality of the search bar.

**Dislikes**

Some of the interface elements were not as helpful or clear. These interface elements were mostly visual and not interactive. One notable exception to this was the fact that participants wanted selections to highlight relevant elements across all coordinated views.

**Design Ideas**

**Additional features**

One of the most notable feature that our users expressed interest in having was filtering, since it was not available in the version that they had been using. Users mentioned wanting to be able to very quickly identify the most important words through (perhaps) a word cloud. Furthermore, users expressed interest in being able to manually add ‘stop words’, thus removing the ones that they deemed to be unnecessary from the visualization.

**Specific Design Elements**
Our participants provided us with extensive input on the general style and design of our application and the available interactions, as well as specific improvements on interface elements like the topic comparison tooltip, the tweets panels and the topics panels and the legends.

*Use Cases*

Our participants helped us determine which tasks our visualization tool is best for. More specifically, they mentioned that one of the easiest tasks is identifying the most similar topics over time, which was after all the goal of our tool. Another task that our tool is helpful for is identifying topics that have been 'created' or 'removed', as well as looking at the evolution of a topic by searching for a specific word. Other use cases are finding topics that relate to specific tweets, as well as tweets that relate to specific topics and comparing two adjacent topics. However, one user asserted that our tool is not very good at comparing topics over time, since only the similarity of adjacent topics is visualized.

*How did Session 1 affect the design*

After discussing the results within the team, a number of changes were made. The most fundamental change that was made was selecting all related topics (and edges) across the timeline when a topic is clicked. Furthermore, a great number of the changes were made that related to specific visual elements that the participants noted, like the colors chosen, the general consistency of the interface, as well as sorting and positioning. Last, we change the initial “rank topics by least edge crossing” to “rank by size”, because users were confused.

*Session #2*

Using the improved version of our tool and the lessons learned from Session #1, we visited 2 more users to examine if the improvements that were suggested were implemented in a satisfactory way, as well as get feedback that would aid us in improving our product even more. Our users were now very familiar with topic modelling, as were as Twitter data visualization. After making our users familiar with TopicFlow, we shortly described the changes that were made and then asked them to perform specific tasks that were derived from the use cases in order to test, once again, what types of tasks our tool is best for, as well as specific likes, dislikes and design ideas. The tasks that our users were asked to perform were as follows:

1. Identify the two most similar topics
2. Identify the reason why these two topics are similar
3. Identify a topic that did not die out for this timespan
4. Identify a topic that emerged and died really soon
5. Identify a topic that diverged into more than one topic
6. Pick a topic and find out what it is about in depth
7. Identify 3 of the most important words
During each task, facilitators took notes about problems that he/she observed, as well as comments by the participants. After each task, the participants were asked to answer the following questions:

1. “How efficient was TopicFlow in aiding you achieve the task you just attempted on a scale from 1 to 9, 1 being very inefficient and 9 being very efficient?”
2. “How correct were the results for the task you just attempted on a scale from 1 to 9, 1 being very incorrect and 9 being very correct?” (i.e., how much do you trust your conclusions)
3. “How effective was TopicFlow in aiding you achieve the task you just attempted on a scale from 1 to 9, 1 being very ineffective and 9 being very effective?”
4. “How satisfied are you in regards to this task on a scale from 1 to 9, 1 being very unsatisfied and 9 being very satisfied?”

After the participants had attempted all the tasks, they were asked once more to provide us with likes, dislikes and ideas for future iterations of this tool. Finally, we discussed their general impression of our tool and the visualization (perceived effectiveness, efficiency and correctness) in an open ended manner.

Results of Session 2
As in session 1, the collected notes were aggregated and analyzed by our team. Talking about Efficiency (EI), Correctness (C), Effectiveness (EC) and Satisfaction (S). The tasks were ranked as follows.

Tasks

1. **Identify the two most similar topics** (EI:6,C:4,EC:6,S:6)
   Our participants felt that our interface made this task slightly hard because edge thickness is not very easy to use as a comparison but suggested that that would be fixed either by filtering or by coloring edges according to their thickness. Our participants also mentioned that they would like to see a measure (metric) of similarity in order for them to be able to trust the results more.

2. **Identify the reason why these two topics are similar** (EI:4,C:7,EC:8,S:7)
   Users found our tool to be very helpful with this task. However, they did not consider it to be very efficient since it was possible to forget which two topics they were looking at.

3. **Identify a topic that did not die out for this timespan** (EI:9,C:9,EC:9,S:9)
   There was some confusion about whether “ending out” meant that the topics were becoming “smaller” (had less tweets), however, users felt that our tool was ideal for this task.

4. **Identify a topic that emerged and died really soon** (EI:7,C:9,EC:6,S:7)
   Our users felt that this task was performed sufficiently though the aid of TopicFlow, however, the thin lines were a bit difficult to follow.

5. **Identify a topic that diverged into more than one topic** (EI:9,C:9,EC:9,S:9)
   This is a task that was performed almost instantly and with no issues whatsoever.
6. **Pick a topic and find out what it is about in depth**
   This task was also performed very efficiently and effectively. However, it was noted that how well a user performs this task strongly depends on how much they know about topic modelling.

7. **Identify 3 of the most important words**
   The users noted that TopicFlow helps a lot with this task, but it could do slightly better if there was a wordcloud.

*Likes*

The users considered TopicFlow to be very good at figuring out data in general and enjoyed the idea that topics are progressing and flowing into each other, which is something that they felt our product could do very well. They felt that our visualization was effective and appreciated the coordinated views that allow for selection of topics with brushing and linking. Last, they believed that visualizing topic similarity with thickness of edges is great, but connecting it to a second dimension (like color) could make it even better.

*Dislikes*

The users would have appreciated a similarity measure (metric) since the version of the interface that they saw kept it unclear what was the math behind the edges. After clicking a node, the color of the node was removed and therefore made it unclear if the topic was emerging, continuing or ending. Lastly, the number of edge crossings was disorienting to the users and it was hard to follow a narrative of topics.

*Design Ideas*

The users provided us with a lot of design ideas for our tool. Most notably:

- Showing some statistics, such as similarity score when hovering the edge between topics. Furthermore, when a node has been selected, the color of a topic could suggest similarity between the selected topic and the related ones.
- When clicking a node, show the outline of the node color with the original color.
- Make an option to switch between: sort by topics, or sort by least edge crossing.
- When clicking a node, instead of coloring the related topic node with white color, we can color based on the similarity of the node with each topic in the same chain.
- Show the best fitting lines for a topic, therefore suggesting a narrative.
- Clicking a node again could clear the selection.

*Suitable tasks*

This session reinforced our previous belief:

Our tool is most suitable for identifying emerging, continuing and related topics, diverging topics, as well as performing content analysis on these topics and the tweets. It is however not suitable for displaying similarity between any two topics over a timespan.

*How did Session 2 affect the design*

After discussing the results, more important changes were made. Most notably, we changed the selection mechanism to make it more clear and help counter the edge overlapping problem and
introduced a filtering mechanism. The filtering mechanism allows for filtering by topic size (i.e. number of tweets), filtering by topic type (emerging, continuing, ending and standalone topics) as well as similarity weight. We believe that these filters will also help users explore narratives between the data and make relationships clearer.

CONCLUSION
Moving from the traditional analysis on word, word pairs and hashtags, TopicFlow helps users gain insights on the evolution of topics over time. The evaluation showed that users found our tool very useful and interesting, and appreciated the interactions provided. However, a significant limitation of our tool is that, as mentioned in the sessions, it does not provide a narrative that helps users understand the entire history of a topic by comparing it to all its ancestors and descendants.

FUTURE WORK
After conducting the expert review sessions, we collected many useful suggestions on improving TopicFlow on different levels. A major improvement would be the ability to compare any 2 topics over time, thus providing a narrative for the evolution of a topic. Furthermore, users expressed the desire to be able to customize the stop words used by the LDA algorithm, the number of bins, as well as sort by different metrics and get more elaborate information on the topic similarity. Last, a word cloud would help TopicFlow function better as a tweet analysis tool.

ACKNOWLEDGEMENTS
We gratefully acknowledge the guidance of Ben Shneiderman, Cody Dunne, Marc Smith, Jana Diesner, John Alexis Guerra Gómez, Jen Golbeck and Awalin Sopan for their valuable help during the multiple stages of design, evaluation, and implementation of our tools.

REFERENCES