Social Knowledge Graph Explorer

Omar Alonso
Microsoft
omalonso@microsoft.com

Vasileios Kandylas
Microsoft
vakandyl@microsoft.com

Serge-Eric Tremblay
Microsoft
sergetr@microsoft.com

ABSTRACT

We present SKG Explorer, an application for querying and browsing a social knowledge graph derived from Twitter that contains relationships between entities, links, and topics. A temporal dimension is also added for generating timelines for well-known events that allows the construction of stories in a wiki-like style. In this paper we describe the main components of the system and showcase some examples.

ACM Reference format:
Omar Alonso, Vasileios Kandylas, and Serge-Eric Tremblay. 2019. Social Knowledge Graph Explorer . In *Proceedings of Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, July 21–25, 2019 (SIGIR '19),* 4 pages.
https://doi.org/10.1145/3331184.3331410

1 INTRODUCTION

Given the enormous amount of human sensing in the world at any given moment that is surfaced by prominent social networks like Twitter and Facebook, there is quite a bit of potential for building richer data aggregations and applications that go beyond standard product features like recommendations and trending topics.

Twitter, for example, produces real-time information which can be extremely valuable for detecting new relationships that would take some time to be captured by media outlets or have an entry on Wikipedia. We believe that by mining and extracting specific content from inside a social network like Twitter, a knowledge graph can be automatically derived and later be used as underlying structure for supporting a number of applications.

The SKG Explorer is not just another browser user interface for users to navigate the collection. Besides basic functionality like search and navigation, it allows users to query and derive a story for a given entity that is produced in a wiki-like style. We can think of this as a "wikification" of the search results where the retrieval units are presented in temporal order with more context, evidence from Twitter, and the ability to pivot to related stories for exploration. In that sense, the stories and their associated pivots (related topics) help users navigate an information structure (our SKG graph), similar in principle to information cartography [3].

Compared to other research on knowledge base generation and information extraction from Twitter, we take a bottom-up approach with an emphasis on identifying good quality. That is, selecting good content, users and links in an efficient manner that enables the creation of connections for high quality elements. The main components are:

(1) Users. We use a set of Twitter users called trusted users which are discovered by 2-way communications initiated by verified users. This computation is repeated, expanding the set of trusted users by another ring in each iteration and no more users are added after the 10^{th} repetition.
(2) Links. Popular links that are shared by those trusted users. Top popular links (most often shared). Top viral links using the average pairwise distance between nodes in the diffusion tree.
(3) Topics. Extraction of topics such as entities, hashtags, and n-grams from tweets. High confidence entities are stamped with an identification from Satori, Microsoft's knowledge base.
(4) Posts. Posts are tweets used as supporting evidence (or provenance) for each node and connection in the graph.
(5) Time. The graph is archived at regular intervals with snapshots which can be used to produce a timeline view of key topics.

The SKG schema captures the main components of the social network and the connections between them. It focuses on "first order" connections, i.e. those immediately discoverable. More complicated connections, which can be derived by more advanced analysis of the data, are left to be computed by other applications based on the SKG data and as required by the application. The design philosophy is that the data are computed on a continuous basis (e.g., hourly) and capture the main information from the social network, but they do not try to do everything.

2 SCHEMA AND IMPLEMENTATION

The SKG schema consists of 4 component tables, which contain information about the nodes of the graph, and 9 connection tables, which link the nodes of the graph and contain any extra information needed to select the desired types of connections. The component tables are:

(1) `Users`: Contains information about the user
(2) `Links`: Contains information about the links mentioned/shared in the posts
(3) `Topics`: Contains information about the entities, hashtags, cashtags, or n-grams mentioned in the posts
(4) `Posts`: Contains a small number of top posts to be used as supporting evidence for the nodes and edges of the graph and their information

The connection tables contain information about the connection between two elements. They are:

SIGIR '19, July 21–25, 2019, Paris, France

ACM ISBN 978-1-4503-6172-9/19/07…$15.00
https://doi.org/10.1145/3331184.3331410

(1) `Users-Links`: user-link relation (authored, coreferenced), score, supporting posts.
(2) `Users-Topics`: topic type, user-topic relation (authored, coreferenced, fanOrAuthority), score, supporting posts.
(3) `Users-Users`: user-user relation (mentioned, retweeted, replied, coreferenced), score, supporting posts.
(4) `Links-Topics`: topic type, score, supporting posts.
(5) `Topics-Topics`: topic type, score, supporting posts.
(6) `Links-Links`: score, supporting posts.
(7) `Users-Posts`: timestamp, relation, score.
(8) `Links-Posts`: timestamp, score.
(9) `Topics-Posts`: topic type, timestamp, score.

The auxiliary tables are:

(1) User index: For fast lookups by user screenname.
(2) Top elements: The top elements of each component type (topic, user, link) from the graph.

The topics are always annotated by their type (hashtag, n-gram, entity, cashtag), so that it is easy to select, for instance only the hashtags associated with a user.

Bidirectional connections are indexed in both directions, both by the first and by the second element of the connection (e.g., the table `Links-Topics` is indexed both by link and by topic). This allows fast lookups and connecting from either direction. When the topics are the second element of the connection, they are clustered, but when they are the first, they are not. The topic clustering allows searching for connections, such as links associated with the topic, using only parts of a topic (e.g., "Obama" for the topic "Barack Obama"), but when the topics are returned as results, only the full topic is returned (e.g., the n-gram "Barack Obama") and not its parts (not "Obama").

Most connection tables contain a description of the relation represented by the connection. When a user is involved, the relation could be: authored (the user authored the post that contains the topic, link, etc.), coreferenced (the user was mentioned in the same post together with the other topic, link, etc.) or mentioned (the user was mentioned in the post). For `Users-Topics`, the user could also be a fan or authority on the topic, discovered with an expertise detection algorithm. For `Users-Users`, the relation captures if one user mentioned the other in the post, replied to the other user, retweeted the other user, or if both users were coreferenced in the same post.

All tables contain one main score. Other scores are also available depending on the table. For the component tables, the main score is used for static ranking users, topics, links, and posts. This score is computed from a combination of other scores that may be available, such as the frequency of occurrence, spam score, authority score etc. For the connection tables, the main score represents the strength of the connection and is based on a normalized frequency of occurrence. The normalization is enforced by allowing every account to have a single "vote" for the connection. So, repeated posts by the same account, mentioning for instance a link and a topic, only contribute one vote to the connection of that link and topic. This helps reduce the effect of spammer and advertising accounts who, by repeatedly tweeting the same things, would otherwise over-inflate the importance of the connection.

The previously mentioned computations are executed on an hourly basis, with each execution processing the messages posted in the most recently available one-hour interval. Each produced graph is thus a snapshot of the social activity during that hour. To analyze the activity over a longer time interval (e.g., a day or a week), we could apply the same approach to messages from the longer interval, but this is inefficient, both because it would require more processing time and because all these messages have already been processed. So instead, we developed a process to aggregate and summarize the graphs of multiple, shorter time intervals into a single graph corresponding to a longer time interval. In our case, at the end of each day, we aggregate the 24 hourly graphs into a single daily summary graph. This is repeated hierarchically, so at the end of each month we aggregate all the daily graphs of the month, and at the end of the year we aggregate all monthly graphs to produce a yearly summary graph.

The complete back-end data pipeline and algorithms have been implemented in Cosmos and run on a distributed cluster continuously. We generated SKG using 2 years worth of Twitter data. The underlying data processing pipeline processes 120M tweets daily filtered by spam/adult/etc. We keep only those tweeted by our 15M trusted users which leaves around 22M tweets. We select 65K links, 600K topics, and 65K hashtags according to tunable thresholds which may change. We then compute many connections, for example, 3.5M user-user interactions and 100K topic-topic relations. Finally, we select 5 tweets per user, per topic and per link for a total of 6M tweets. This amounts to around 11 Gigabytes daily.

There have been some related work on extracting relationships from Twitter data but they do work on samples ([2], [4]), in contrast, we ingest the entire Twitter firehose.

3 DEMONSTRATION

We describe a couple of examples that showcase the system in action where the user is looking for the latest information on a topic and can discover connections and explore related relationships. The system provides a feature to show all the retrieved elements in a wiki-like page that is constructed dynamically.

We now explore the topic of Brexit. We start by searching for the hashtag #brexit, which returns other related topics, in the form of entities, hashtags and n-grams. Related entities are UK, EU, Britain, Theresa May. Related hashtags are #eu, #stopbrexit, #fbpe. Some related n-grams are vote, leave, people, referendum. We can dive into any of these connections by looking at tweets that mention #brexit with any of the other topics. For example, we can see tweets that refer to Theresa May in the context of Brexit (Figure 1). We can continue exploring links or images people share about it, or who are some user accounts who are strongly associated with the topic, such as Richard Corbett, the Daily Express and other activist accounts. Or, we can pick another associated topic and explore that. For example, we might not know what the hashtag #fbpe stands for. We can search for it and see that it is related to pro-remain topics, like #stopbrexit, or remain. Scrolling down the list of related topics we find what the initials #fbpe mean: Follow Back, Pro EU. The set of topics, links and users for #fbpe is different and reflects the pro-remain association of the hashtag.

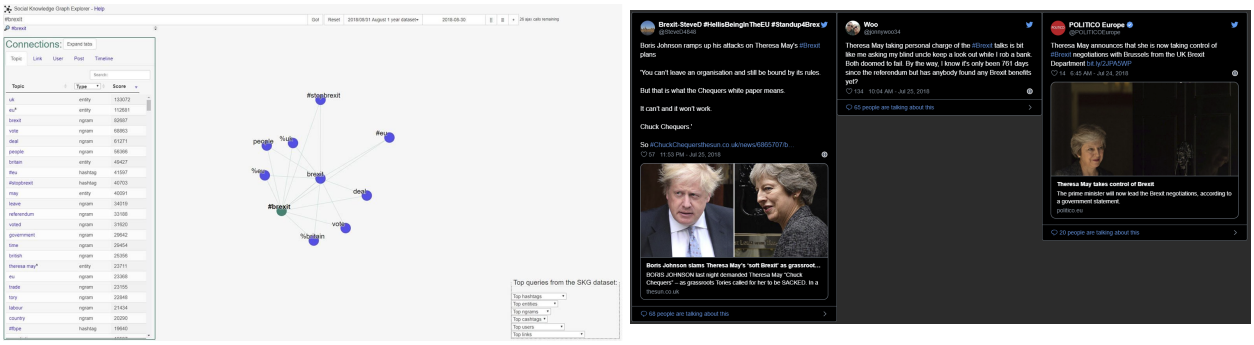

Figure 1: The SKG Explorer application. We search for a topic (brexit) on a specific day and can filter the connections by entities (e.g., Theresa May) and hashtags (e.g., #brexit, #stopbrexit, etc.). Clicking on the "Link" tab shows a few examples of links shared on Twitter. Same goes for the tabs "User" and "Post". The visualization in the middle pane allows us to explore the graph visually by clicking on the nodes.

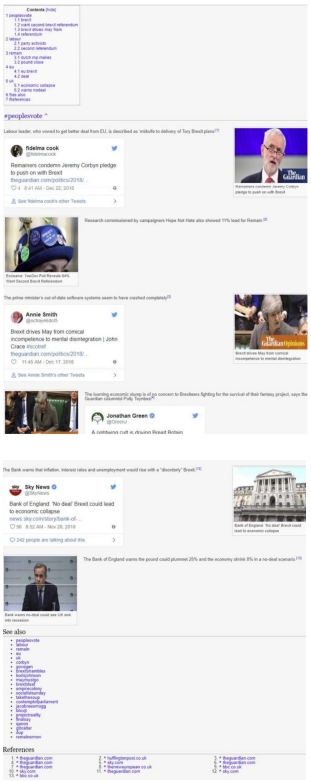

Figure 2: Wikification for the brexit story over time derived from the SKG graph: 1) table of contents, 2) specific items for subtopics, 3) related stories, and 4) references. Entries in the story show supporting evidence from Twitter.

In a different presentation, we can examine Brexit as a wiki page. This automatically generated page is organized in sections and subsections, selected from other topics related to Brexit. Each section contains short paragraphs of text and images, derived from the snippets (descriptions) and top images of the best links shared on the topic. These are extracted from Open Graph or HTML meta attributes of each link. The top tweets that shared the links are also included and their content provides contextual commentary on the topic. The wiki page also contains at the end a list of related topics, which lead to other automatically generated wiki pages, as well as references to the URLs that were used for the page construction. The structure of the page thus resembles the structure of a wiki page, with a title, table of contents, sections with text and images, related topics and references (Figure 2).

An alternative view shows how the story of Brexit evolved over time. Since we know what was tweeted or shared and when, we can construct a timeline of the major events around Brexit. The details of this construction are described in [1]. The timeline view has a similar structure as the wiki view, but the sections correspond to dates and the section contents are generated from topics and links that were shared that date. For example, the section for December 7, 2018 mentions the results of a public poll and an announcement of the Bank of England about the Brexit effects on the economy. Because every link and topic is associated with a score, we can vary a threshold to generate a shorter timeline with only the dates of major events, or a longer one that contains less significant events.

As a second example, we want to explore what is happening in Syria. We start by searching for #syria, but the entity or n-gram Syria would also lead to the same results. We find that the strongest connections of #syria are to the topics Russia, ISIS, Israel, Iran, war. Even if we knew nothing about Syria, this would give us a good idea that a war is happening and the sides that are involved. We can find more details by looking at the links that were shared about Syria. Suppose we want to understand how Iran is related to Syria. We can look at what people are tweeting about both Syria and Iran. Or

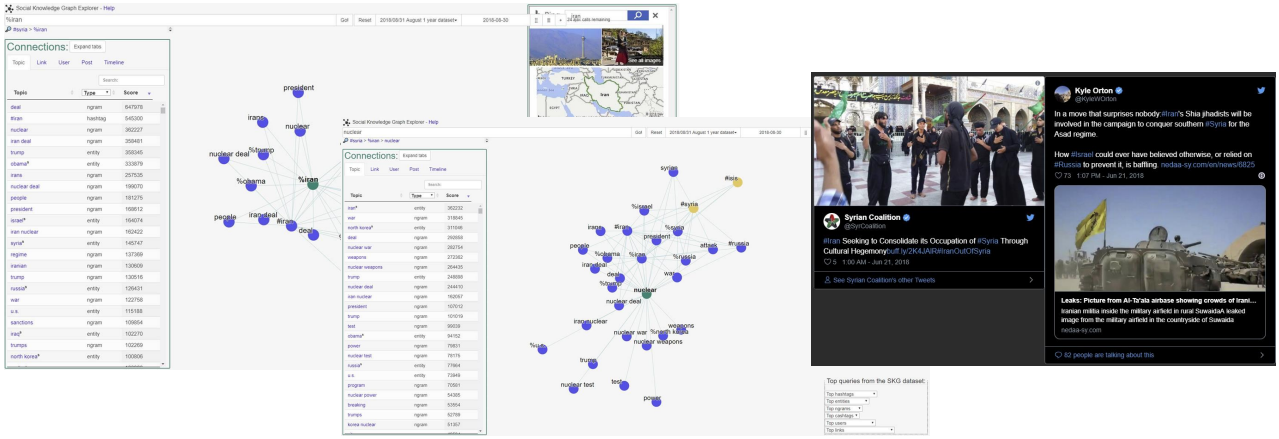

Figure 3: We search for the topic Syria and explore the following connections in order: Iran and the nuclear topic. We then observe that #isis is mentioned and hover over to see a sample of tweets to get more context. Note that because Iran is a country, the entity is stamped with more factoid information that is then displayed on the top right corner.

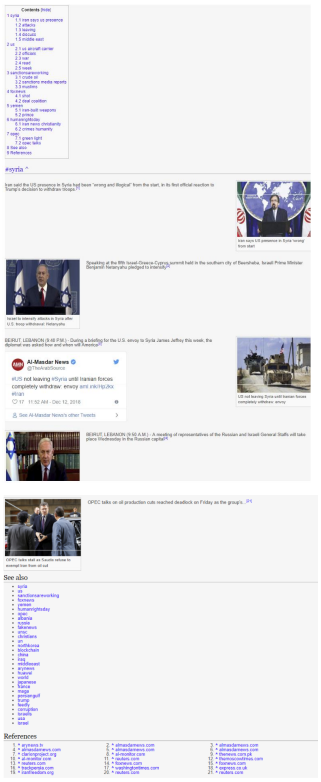

Figure 4: Wikification for the Syria-Iran story over time derived from the SKG graph. Similarly to the Brexit story, the structure is the same: 1) table of contents, 2) specific items for subtopics, 3) related stories, and 4) references.

we can follow the connection to the entity Iran and look at related topics and links. There, besides Syria, we find mentions to a nuclear deal, Trump, and Obama, from which we can read posts and links about that deal. Even though the deal is not directly related to Syria, we can discover that it is an important topic and gain a spherical understanding of Middle Eastern politics and how for example Iran Syria and USA are inter-connected (Figure 3).

The wiki view of #syria structures the page across the major topics: Russia, Turkey, US, ISIS, Aleppo, Trump, Israel etc. We are curious why Aleppo is mentioned as an important topic. The section of the auto-generated wiki page mentions heavy clashes and a toxic attack. The text and links in the section provide further details, while the images show scenes of fighting and from hospitals. Similar to Brexit, the timeline view shows the major events and topics that happened, organized by date (Figure 4).

A final item on the wikification process is that it is possible for the user to control the length of the generated document by a slider widget on the explorer.

4 CONCLUSION

We described a knowledge graph explorer that allows users to query and discover relationships derived automatically from Twitter. The utility of the application is to retrieve, extract, and present social information as a unit that can be dissected in different ways (e.g., entities, links, topics, etc.) and, at the same time, provide a wiki-like view that shows relationships in an evolving story. The user can explore related topics, entities, and observe the sources that were used to derived the story.

REFERENCES

[1] Omar Alonso, Vasileios Kandylas, and Serge-Eric Tremblay. 2018. How it Happened: Discovering and Archiving the Evolution of a Story Using Social Signals. In *Proc. of JCDL*. 193–202.

[2] Alan Ritter, Mausam, Oren Etzioni, and Sam Clark. 2012. Open Domain Event Extraction from Twitter. In *Proc. of KDD*. 1104–1112.

[3] Dafna Shahaf, Carlos Guestrin, Eric Horvitz, and Jure Leskovec. 2015. Information cartography. *Commun. ACM* 58, 11 (2015), 62–73.

[4] Anders Søgaard, Barbara Plank, and Héctor Martínez Alonso. 2015. Using Frame Semantics for Knowledge Extraction from Twitter. In *Proc. of AAAI Conference on Artificial Intelligence*. 2447–2452.