Proceedings of the Twelfth International AAAI Conference on Web and Social Media (ICWSM 2018)

Automatic Story Evolution Wikification from Social Data

Omar Alonso, Vasileios Kandylas, Serge-Eric Tremblay
Microsoft
{omalonso, vakandyl, sergetr}@microsoft.com

Abstract

We present the generation of a new and dynamic data asset that captures the evolution of a story from different perspectives. In contrast to news articles that are ranked by relevance and freshness in a search engine or a static Wikipedia article that provides an overview of the event or topic, our solution consists of the automatic construction of a wiki-like document that highlights the salient items of a topic as it evolves over time, with related pivots that allow the user to explore related stories. We demonstrate the effectiveness of our approach by processing a dataset comprising millions of English language tweets generated over a one year period.

Introduction

Social activity in Twitter or Facebook provides real-time information but, at the same time, can be overwhelming for two reasons: there is very little context for the uninformed user and there is quite a bit of noise or duplication for the informed user looking for the latest update. News articles tend to be less frequent but provide more context and are ranked in search engine results by topicality and freshness. Wikipedia articles are curated documents by Wikipedians that describe the topic to a certain extent but they may be difficult to consume when looking at the progression of a story. Sometimes, there is no Wikipedia article yet like in the case of breaking news or if such entry exists the content needs an update. Also, Wikipedia articles rarely show the progression of a topic.

Current online tools are not very well suited for summarizing unfolding events or providing the evolution of a story and related topics. In this work, we are interested in combining social signals and news articles to construct a new document that captures the backbone of a story over time with all associated information including entities and references.

We present a system that supports collating, storing, querying, and retrieving evolving stories about events. The aim of our solution is to fulfill information seeking scenarios by algorithmically generating the core of the story as it evolves over time by using selected relevant content derived from hashtag discussion and link sharing activity in Twitter. In contrast to specific news outlets that may bias content editorially or compared to the small number, on average, of Wikipedians per article, our work uses social sensing at scale by harvesting content that the crowd considers relevant. We use human sensing on Twitter as a large distributed crowdsourcing crawler where links are constantly shared and annotated with hashtags, tags, and/or entities. This large scale link selection can be seen as a collaboratively retrieved set of relevant documents for a query as expressed by a hashtag or named entity.

Our proposed new type of document is not as encyclopedia-centric as a Wikipedia article but, instead, dynamically adjustable to the many data components of the story, always up-to-date, and constructed in a fashion that allows different aggregations and applications. The concept of a wiki view has advantages organizing information as everybody knows how to read and navigate a Wikipedia page. The main challenge for this problem is the detection of specific social signals that are used to retrieve relevant content for the story.

Story Evolution Wikification

Instead of a single Wikipedia article edited by a few Wikipedians, we present a wiki document constructed using editorially written content that is selected by the crowd and ranked and refined by our wikification algorithm. Our techniques extract the backbone of a story (i.e., tax reform) and produce a new document that highlights how the topic has evolved over time along with connections to related stories, allowing the user to explore associated content. We can think of it as a dynamic Wikipedia page that is edited automatically based on social activity in Twitter.

Our work includes a fine-grained vote counting strategy that is used for weighting purposes, the use of social data for pseudo-relevance feedback and query expansion along with a timeline algorithm as the base for a story.

In order to deal with spammers and advertisers, we rely mainly on votes from individual accounts, instead of raw frequencies of hashtags, links, etc. The problem with frequencies is that spam and advertising accounts tend to post multiple tweets per day that contain the same information. For example, a company account might include a link to the company website in every tweet, or political supporters may add a hashtag about their favorite party in all their tweets. These behaviors can make links, hashtags, etc. artificially

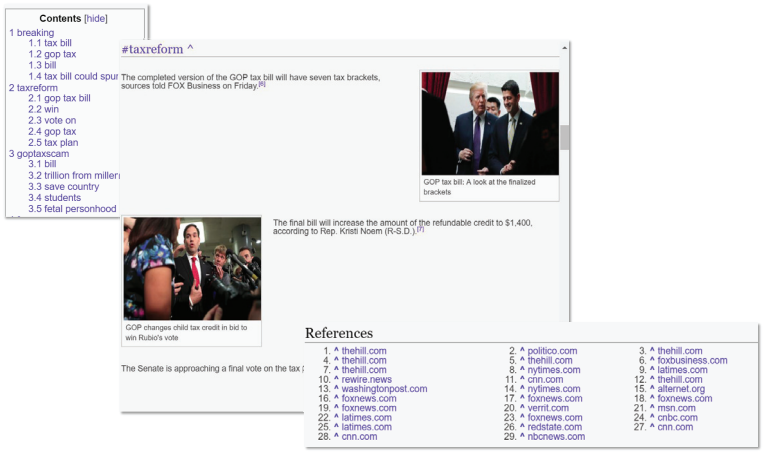

Figure 1: Wikification for the tax reform story: 1) table of contents, 2) specific items for subtopic, 3) references.

popular, when in fact most of their popularity comes from a small number of accounts. We therefore assign a single vote to each account for the time period under consideration, so that one account cannot skew the frequency. Our vote data structure maintains separate vote counters for tweets and retweets of the various elements (e.g., hashtags, links, etc.) or connections (e.g., hashtag-link, entity-hashtag, etc.). We utilize social data as a temporal context that can be used for query expansions and how they can provide relevant document links for deriving a story similar to the notion of information cartography (Shahaf et al. 2015).

Assuming an initial set of relevant tweets about a topic (expressed as hashtag or entity), we extract a set of document links $docs = \{d_1, \ldots, d_n\}$ and associated n-gram list, defined as contextual vector $cv = \{t_1, \ldots, t_m\}$, we compute a document score that measures the similarity of each term t_j in the contextual vector with the document title multiplied by its counters that aggregate a number of behavioral signals (e.g., RTs, likes, etc.): $score_i = \sum_c \cos(d_{i,t}, t_j) * d_{i,c}$ where $d_{i,t}$ is the title for d_i, $d_{i,c}$ is counter c for d_i, and term t_j in the contextual vector. Related hashtags (`#SuperBowl` and `#SB51`) are computed using SimHash on the contextual vectors and used for query expansion on a specific temporal anchor.

Once we have all the relevant document links, the problem then is how to select the best candidates per day and use them to generate a timeline. If we look at each timestamp (i.e., date), which we consider a marker on our timeline, we have a ranked list of titles to chose from. Using the extensions to pseudo-relevance feedback and query expansion, those links are re-ranked and selected as entries.

For the wikification part, we rely on a timeline algorithm that uses the methods described in the previous paragraphs to identify the most relevant document link titles according to a query. Information extracted from link metadata (e.g., title, image, description) forms the base for the generation of the timeline. Supporting evidence, that is, the original tweet, is also presented in the document as provenance.

Figure 1 shows the main structure of the wiki-like page.

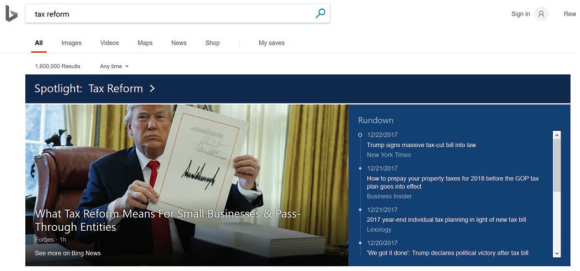

Figure 2: Tax reform story presented as news answer.

The page structure contains four main sections: table of contents (TOC), story evolution, related stories, and references. For building the TOC, we use a standard hit-list clustering approach that takes the link titles and descriptions, and populates a trie data structure. The top level entries are selected for the TOC based on relevance scores. The story evolution part contains the bulk of the timeline algorithm described previously. Related stories are basically the pivots to the main story and are computed using a combination of query suggestions and TOC. Finally, the references are the link URLs used to construct the story that serve as ultimate source.

Conclusion

In this demo, we showed the wikification of the evolution of a story by combining social data with news articles and present the outcome as a wiki-like page. The data set that is used to construct the document can also be repurposed for other experiences like an answer in the Bing search engine (Figure 2).

References

Shahaf, D.; Guestrin, C.; Horvitz, E.; and Leskovec, J. 2015. Information cartography. *Commun. ACM* 58(11):62–73.