Automatic Story Evolution
Wikification from Social Data

Omar Alonso, Vasileios Kandylas,
Serge-Éric 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 perspec-
tives. 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 docu-
ment 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 ap-
proach by processing a dataset comprising millions of En-
lish language tweets generated over a one year period.

Introduction
Social activity in Twitter or Facebook provides real-time in-
formation 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 summa-
rizing unfolding events or providing the evolution of a story
and related topics. In this work, we are interested in combin-
ing social signals and news articles to construct a new doc-
ument 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 scenari-
os 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 Twit-
ter. 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 crowd-
sourcing crawler where links are constantly shared and an-
notated 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, dy-
namically 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 ev-
eybody knows how to read and navigate a Wikipedia page.
The main challenge for this problem is the detection of spe-
cific 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 us-
ing 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 sto-
ries, allowing the user to explore associated content. We can
think of it as a dynamic Wikipedia page that is edited auto-
matically 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 fre-
cuencies 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
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 re-tweets of the various elements (e.g., hashtags, links, etc.) or connections (e.g., hashtag-link, entity-hashag, 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 \( \text{docs} = \{d_1, \ldots, d_n\} \) and associated n-gram list, defined as contextual vector \( \text{cv} = \{t_1, \ldots, t_m\} \), 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.): \( \text{score}_i = \sum c \cos(d_{i,t}, t_j) \cdot d_{i,c} \) 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.

**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**