**Noname manuscript No.**
(will be inserted by the editor)

# A Topic Recommender for Journalists

**Alessandro Cucchiarelli · Christian Morbidoni · Giovanni Stilo · Paola Velardi**

Received: date / Accepted: date

**Abstract** The way in which people acquire information on events and form their own opinion on them has changed dramatically with the advent of social media. For many readers, the news gathered from online sources become an opportunity to share points of view and information within micro-blogging platforms such as Twitter, mainly aimed at satisfying their communication needs. Furthermore, the need to deepen the aspects related to news stimulates a demand for additional information which is often met through online encyclopedias, such as Wikipedia. This behaviour has also influenced the way in which journalists write their articles, requiring a careful assessment of what actually interests the readers. The goal of this paper is to present a recommender system, What to Write and Why, capable of suggesting to a journalist, for a given event, the aspects still uncovered in news articles on which the readers focus their interest. The basic idea is to characterize an event according to the echo it receives in online news sources and associate it with the corresponding readers' communicative and informative patterns, detected through the analysis of Twitter and Wikipedia, respectively. Our methodology temporally aligns the results of this analysis and recommends the concepts that emerge as topics of interest from Twitter and Wikipedia, either not covered or poorly covered in the published news articles.

**Keywords** Recommender Systems · Wikipedia · Twitter · Online News · Event Detection · Temporal Mining

Alessandro Cucchiarelli
Università Politecnica delle Marche, Ancona, Italy E-mail: a.cucchiarelli@univpm.it

Christian Morbidoni
Università Politecnica delle Marche, Ancona, Italy E-mail: c.morbidoni@univpm.it

Giovanni Stilo
Sapienza University of Rome, Rome, Italy E-mail: stilo@di.uniroma1.it

Paola Velardi
Sapienza University of Rome, Rome, Italy E-mail: velardi@di.uniroma1.it

## 1 Introduction

In a recent study on the use of social media sources by journalists [20] the author concludes that "social media are changing the way news are gathered and researched". In fact, a growing number of readers, viewers and listeners access online media for their news [14]. When readers fell involved by news stories they may react by trying to deepen their knowledge on the subject, and/or confronting their opinions with peers. Stories may then solicit a reader's *information* and *communication* needs. The intensity and nature of both needs can be measured on the web, by tracking the impact of news on users' search behavior on online knowledge bases as well as their discussions on popular social platforms. What is more, online public reaction to the news is almost immediate [26] and even anticipated, as for the case of planned media events and performances, or for disasters [25].

Assessing the focus, duration and outcomes of news stories on public attention is paramount for both public bodies and media in order to determine the issues around which the public opinion forms, and in framing the issues (i.e., how they are being considered) [3]. Furthermore, real-time analysis of public reaction to news items may provide a useful feedback to journalists, such as highlighting aspects of a story that need to be further addressed, issues that appear to be of interest for the public but have been ignored, or even to help local newspapers echo international press releases.

The aim of this paper is to present a news media recommender, What to Write and Why ($W^3$), for analyzing the impact of news stories on the readers, and finding aspects – still uncovered in news articles – on which the public has focused their interest. The purpose of $W^3$ is supporting journalists in the task of reshaping and extending their coverage of breaking news, by suggesting topics to address when following up on such news items. It does so on the basis of a temporal mining algorithm that detects bursty topics independently from online news, Twitter messages, and Wikipedia clicklogs. Next, it aligns clusters related to the same topic in each source, to compare users' information and communication needs with the story coverage provided by news media. The recommendation is based on the result of this comparison.

For example, we have found that a common pattern for news readers is to search events of the same type occurred in the past on Wikipedia, which is not surprising per se: however, among the many possible similar events, our system is able to identify those that the majority of readers consider (sometimes surprisingly) highly associated with breaking news, e.g., searching for the 2013 CeaseFire program in Baltimore during Egypt's ceasefire proposal in Gaza in July 2014.

The contribution of our paper is manyfold:

1. We present the first system to provide journalists with *prospective information* of possibly interesting new topics to cover in their articles;
2. Although we exploit a topic detection algorithm that we defined in our previous work [40], the algorithm is enhanced by the use of semantic and graph-based techniques to obtain better topics and to align them across different sources;
3. To face the notoriously difficult problem of evaluating recommenders in absence of datasets, we propose an exhaustive methodology based on novel metrics and combined evaluation approaches.

The paper is organized as follows: in Section 2 we review related works, in Section 3 we describe our dataset and additional resources used in our methodology, which is presented in Section 4. Finally, Section 5 is dedicated to experiments and evaluation and Section 6 contains concluding remarks and future work directions.

## 2 Related Work

To the best of our knowledge, this is the first system for recommending journalists what to write, focusing on presenting users' needs that come from different sources while keeping their original motivation (*information* and *communication* needs). Only a few papers aim to help journalists find relevant content in social media, as we do. In [8] the authors present a tool to assist journalists in the task of identifying eyewitnesses in the context of an event. In [48] a system is described to support journalists in the use of social media. The authors use SVM to identify newsworthy messages on Twitter based on a manually annotated dataset. Very recently, two workshops have been held focusing on the use of data/text mining techniques to help journalists in their work: Natural Language Processing meets Journalism@EMNLP'17[1] and Data Science + Journalism@KDD'17[2]. All these contributions are concerned more with the design of interfaces to help journalists in digging into trending topics or detecting related contents, than with providing *prospective information* of possibly interesting new topics to deal with.

A number of papers analyze a problem which is symmetric compared to the one considered here, i.e., recommending news items to social media users. Among these, the authors in [29] consider the task of recommending articles to readers in a stream-based scenario, when large user-item matrices are not available and time constraints are strict. In their work, they derive a number of statistics extracted from the PLISTA [19] dataset used during the ACM Recsys News Challenge 2013. They also compare performances of several existing recommending algorithms showing that the precision of algorithms depends upon the particular news articles domain. The study in [37] considers, as before, the task of recommending topical news to users. The authors present Buzzer, a system able to mine real time information from Twitter and RSS feeds, using overlapping keywords in most recent tweets and feeds as a basis for recommendation. Evaluation is performed on a small group of 10 participants over a period of 5 days. In [11] the authors propose a news recommender for Wikipedia editors. They aim to integrate Wikipedia pages with events which may either not be mentioned or added with a considerable delay (e.g., Odisha cyclone not mentioned in the page of Odisha despite the very high number of human casualties).

A more weakly related research area is *entity recommendation*, a recommendation engine that links a users' query to a named entity, to help them exploring other topics related to an initial interest. In [1] the authors use a probabilistic three-way Entity Model to provide personalized entity recommendation using three data sources: a knowledge base, search click logs, and entity pane logs. In [2] it is described Spark, a semantic search assistant that links a user's initial query (extracted from Yahoo!) to

[1] `http://nlpj2017.fbk.eu`

[2] `https://sites.google.com/view/dsandj2017/home`

an entity within a knowledge base and provides a ranking of the related entities. Information extraction on entities is performed on Wikipedia, Freebase and other sources such as Movie, Music and TV databases. Wikipedia categories are used also in [5] for entity recommendation.

Other studies related to our work are those aimed at combining and/or aligning information in news, social media, and Wikipedia for purposes other than recommending items. Among these, several papers consider the task of *predicting* the response of social media to news articles [21] [42], rather than *extracting* users' interest related to news articles - as we do - to help journalists focus on additional, yet uncovered, aspects of a reported event. In a similar vein, other scholars aim at *identifying social content* related to online news. A survey of this research area, partly overlapping with the broader area of *event detection*, has been presented in [6]. Among the many contributions, the study described in [43] shares with our work the objective of mapping news articles describing some event with related Twitter messages. More specifically, the following task is considered: given a news article, find the Twitter messages that "implicitly" refer to the same topic, i.e. messages not including an explicit link to the considered article. They are interested in discovering utterances that link to a specific news article rather than the news event(s) that the article is about. First, the authors analyze the KL-divergence between the vocabulary of news articles (using the NY Times as a primary source) and various social media, such as Twitter, Wikipedia, Delicious, etc. They find that, unless part of the original article is copied in the message, which subsumes explicit reference, the vocabularies might be quite different. The method is in three steps: they derive multiple query models from a given article, which are then used to retrieve utterances from a target social media index, resulting in multiple ranked lists that are finally merged using data fusion techniques. Evaluation is performed, in line with other scholars (e.g., [11]), using messages with explicit mention to an article, and then removing the mention. However, as observed by the same authors, evidence suggests that these messages often copy part of the article, an eventuality that could boost performances.
In [23] the objective is to combine news articles and tweets to identify not only relevant events but also the opinions expressed by social media users on the very same event. They use a news article as the query, and a dataset of Twitter messages as the document collection. Next, a latent topic model is defined to find the most relevant tweets wrt a given news topic. Besides topic similarity, they use additional features such as recency, followers count etc., which are then combined using logistic regression or Adaboost. Relevance judgments for evaluating the system have been collected from 11 computer science students. In [36], Wikipedia page views are used to improve the quality of event detection in Twitter in a first story detection task. The authors in [39] present a system to monitor Wikipedia page edits in different languages to detect popular events. In [24] it is presented a method for extracting events from news articles, and organizing them in semantic classes to populate a knowledge base. Finally, the authors in [33] address the problem of linking excerpts from Wikipedia summarizing events related to past news articles.

## 3 Datasets and Resources

To conduct our study, we have created three datasets: Wikipedia PageViews (W), Online News (N) and Twitter messages (T). Data was collected during 4 months from June 1st, 2014 to September 30th, 2014 in the following way:

1. **Wikipedia PageViews**: we downloaded Wikipedia page views statistics from data dumps provided by the WikiMedia foundation[3]. We considered only English queries and we retained only those matching a Wikipedia document, removing redirected requests. Overall, we obtained 27.708.310.008 clicks on about 388 million pages during the considered period.

2. **Online News**: We collected news from GoogleNews (GN)[4] and HighBeam (HB)[5]. Due to existing limitations, we extracted at most 100 news per day from GN, while for HB we downloaded all available news. Each news item has a title, source, day of publication and an associated snippet. Overall, we extracted 351,922 news from 88 sources in GN and 1,181,166 from 325 sources in HB during the considered period. Snippets were about 25 words long in average.

3. **Twitter messages**: we collected 1% of Twitter traffic, the maximum freely allowed traffic stream using the standard Twitter API[6]. Overall, we collected 235 million tweets.

We here assume that Twitter is an indicator of a user's communication needs while Wikipedia page views are an indicator of information needs. The latter assumption is supported by the authors in [47], who investigated the relationships between Wikipedia page views and Google Trends, suggesting that Wikipedia page views trends are closely related to popular global web search trends. This result is also confirmed in [34].

Furthermore, in our study we used the following resources:

1. **NASARI embedded semantic vectors** for Wikipedia pages, generated as described in [4]. NASARI provides a large coverage of concepts and named entities and has achieved state-of-the-art results on several benchmarks. We downloaded the second release[7], covering 4.40 million Wikipedia pages. In our work, we use NASARI to improve clustering of Wikipedia page views (Section 4.1) and to compute the semantic relatedness of recommended entities with news items (Section 5.4).

[3] https://dumps.wikimedia.org/other/pagecounts-raw/
[4] https://news.google.com/
[5] https://www.highbeam.com
[6] https://dev.twitter.com/docs/streaming-apis
[7] http://lcl.uniroma1.it/nasari/\#two

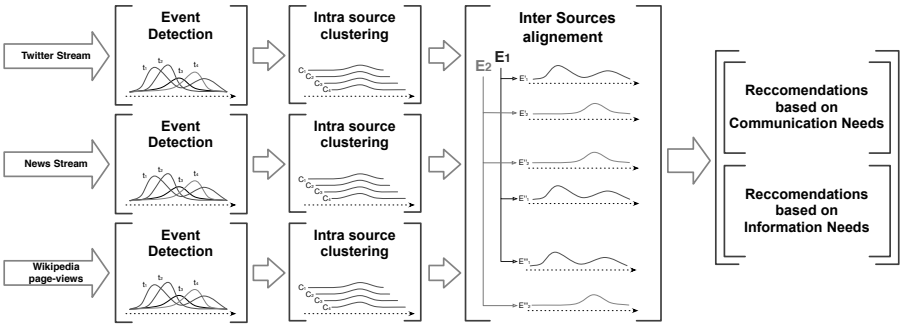


**Fig. 1** Workflow of $W^3$

2. **Dandelion** [8] and **TextRazor**[9]. Both are commercial tools providing named entity recognition (NER) REST APIs that, given a text snippet, identify, disambiguate and link named entities to Wikipedia articles. Dandelion is based on previous research [10], and has been recently further developed and engineered [38]. The reason for using two NER systems is that Dandelion has better performances on news articles and TextRazor on Twitter.

## 4 Methodology

Our methodology is in four steps, as shown in the workflow of Figure 1:

1. **Event detection**: We use a state-of-the-art temporal mining algorithm to cluster tokens (words, entities, hashtags, page views) within temporal windows $L_k$, based on the *synchrony* and *shape similarity* of their associated signals $s(t)$. "Signals" are daily frequencies of words in news items, Twitter messages or Wikipedia page views. Each cluster is interpreted as related to an event $i$. Clusters are extracted independently from online news ($N$), Twitter messages ($T$) and Wikipedia page views ($W$).

2. **Intra-source clustering**: Since clusters are detected in *sliding windows* $L_k$ of equal length $L$ and temporal increment $\Delta$, clusters referring to the same event but extracted in partly overlapping windows may slightly differ, especially for long-lasting events, when news updates motivate the emergence of new sub-topics and the decay of others. For a better characterization of an event, we merge clusters referring to the same event and extracted in adjacent windows, creating *meta-clusters*, denoted with $m_i^S$, where the index $i$ refers to the event (or news items) $n_i \in N$, and $S = \{N, T, W\}$ to the data source.

[8] `https://dandelion.eu/semantic-text/entity-extraction-demo/`

[9] `https://www.textrazor.com`

3. **Inter-source alignment**: Next, an alignment algorithm explores possible matches across the three data sources $N$, $T$ and $W$. For any event $i$, we thus obtain three "aligned" meta-clusters $m_i^N$, $m_i^T$ and $m_i^W$ mirroring respectively the media coverage of the considered event, and its impact on readers' communication and information needs.

4. **Generating a recommendation**: The final step consists in comparing the three aligned meta-clusters and to identify in $m_i^T$ and $m_i^W$ the set of most relevant entities to recommend, respectively $R_i^T$ and $R_i^W$. The quality of recommendations is measured in relation to their *saliency* with respect to news items, and *novelty* w.r.t. what has already been published in news $n_i \in N$. These entities can then be used to suggest journalists additional aspects to cover or deepen when following up on a news item.
   Note that the last step is entirely automatic to avoid subjectivity. However in a realistic setting, rather than using news meta-clusters $m_i^N$ to retrieve the related $m_i^T$ and $m_i^W$, a journalist can be asked to submit a number of seed words related to the news, or the text of a news item.

In what follows, we provide additional details on the four steps of our methodology. For better comprehensibility, we will use the example of the Malaysia Airlines disaster in July 2014 [10] to follow the whole pipeline of our methodology. Furthermore, a summary of symbols used throughout the paper is provided in Appendix A.

### 4.1 Event Detection

To detect event clusters, we use a multi-source enhanced version of a state-of-the-art event detection algorithm, named SAX*, that we first presented in [40]. For the sake of completeness, we summarize hereafter the main features of SAX* (the interested reader is referred to [40] for additional details and a comparison with other competing event detection algorithms).
Our original SAX* algorithm detects bursty events from temporal signals (in [40] signals are words and hashtags in Twitter) in three steps:

1. **Converting temporal signals into SAX strings:** The temporal series $s(t)$ associated to each token are sliced into sliding windows $L_k$ of equal length $L$, normalized and converted in symbolic strings using Symbolic Aggregate ApproXimation (SAX) [27]. The parameters of this step are the dimension of the alphabet $|\Sigma|$ and the number $M = \frac{L}{\Delta}$ of partitions of equal length $\Delta$. An example is in Figure 2, showing the SAX string associated with the normalized time series $s(t)$ for the token *Ukraine*. The series refers to a 10 days window $L_k$ starting on 14 July, 2014, with a 1-day discretization $\Delta$ and binary alphabet. The x axis represents the breakpoint $\beta$ (with $|\Sigma| = 2$ and z-normalization, there is only one breakpoint at y=0). A symbol of the alphabet is associated to each partition $\Delta$, depending on

[10] `http://www.bbc.com/news/world-europe-28357880`

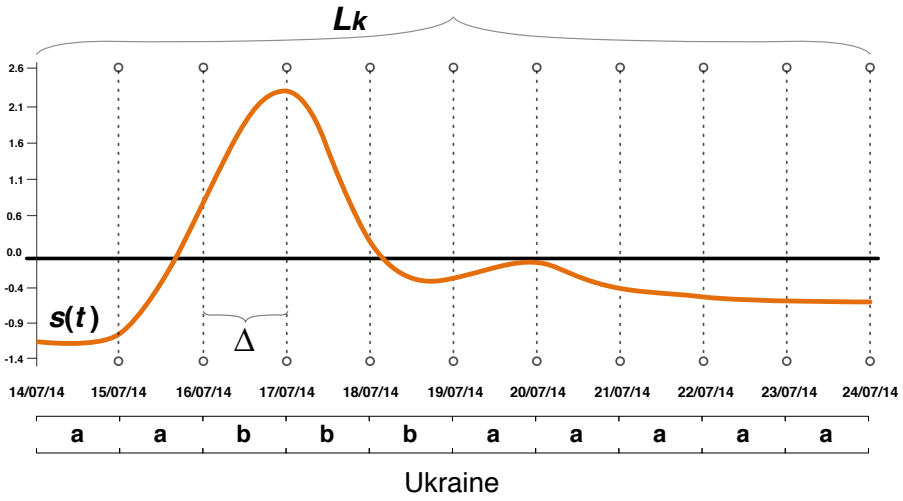


**Fig. 2** SAX conversion of the signal $s(t)$="Ukraine" in a Twitter stream, during a 10-days window in July 2014.

the average value of the signal in the considered slot. Using the binary alphabet {a,b}, the correspondent SAX string for *Ukraine* is *aabbbaaaaa*.

2. **Detectig bursty signals:** Symbolic strings in each window $L_k$ (spanning from time $t = t_k$ to $t_{k+M\times\Delta}$) are matched against automatically learned regular expressions representing common patterns of users' attention. For example, with an alphabet of 2 symbols and $M = 10$, the following regex is used:

$$(a+b?bb?a+)?(a+b?bba*)?$$

which captures all the temporal series with one or two peaks and/or plateaus in the analyzed window (as for example the signal of previous Figure 2). These are common temporal patterns of breaking news, as also found in [46].
Only tokens with a frequency higher than a threshold $f$ and matching the learned regular expressions are considered in the subsequent steps. These are denoted as *active tokens*.

3. **Clustering signals related to the same event or topic:** The detected active tokens are clustered in each window $L_k$ using a bottom-up hierarchical clustering algorithm with *complete linkage* [17]. This clustering algorithm does not require to specify the number of clusters to be generated.

In [40] "tokens" were either words or hashtags (since the objective was event detection in Twitter) while in our present implementation, tokens are named entities - either proper names in news and tweets, or Wikipedia articles clicklogs- , words and hashtags. To detect *active* tokens, we use different thresholds $f$ for different token *types* (entities, words and hashtags) and sources (tweets, news, page views), since frequency ranges are very different (see Section 4.6 for parameter settings). We extract named entities from Twitter messages and online news snippets using two different available systems, Dandelion and TextRazor (see Section 3) selecting the tool

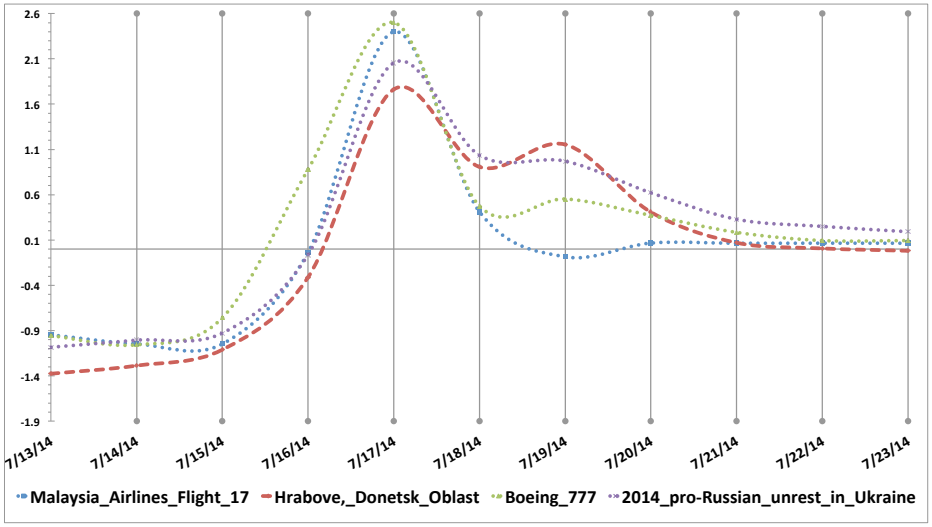


**Fig. 3** Excerpt of clustered normalized time series of newswire tokens related to the Malaysia Airlines crash in July 2014.

which produced more accurate results: Dandelion for news articles and TextRazor for tweets. Both tools provide a mapping to Wikipedia articles. In this way, *entities in all three sources are linked to Wikipedia*, facilitating the subsequent alignment of clusters in different sources (step 3 of Figure 1). Figure 3 shows an excerpt of a SAX* cluster of signals, generated from news items (signals are tokens in news), related to the crash of the Malaysia Airlines flight 17 in July 2014. The corresponding "textual" cluster is (token weights are omitted for readability):

**Window** $L_k$**:** July 13-23, Cluster ID:C0 *[malaysian, Malaysia, Hrabove,_Donetsk_Oblast, 2014_pro-Russian_unrest_in_Ukraine, crash, airlin, flight, malaysia, Ukraine, Malaysia_Airlines, Malaysia_Airlines_Flight_17, Kuala_Lumpur, Boeing_777, Russia, Amsterdam, Washington,_D.C., Eastern_Ukraine, . . . ]* SAX* centroid string: **aaaabbbbbb** peak date: July 17th, 2014

We remark that *SAX* blindly clusters signals without prior knowledge of the event and its occurrence date, and furthermore, it avoids time-consuming processing of text strings, since it only considers active tokens*. The algorithm is then untrained, and computationally effective[11] when compared with lexical and other temporal mining methods. However, especially when applying SAX* to a bulk of short Twitter messages and to Wikipedia clicklogs, two undesired phenomena may occur. First, a given temporal window $L_k$ may include signals belonging to co-occurring breaking events. In this case, if the signal shapes are not sufficiently different, clusters may merge tokens from different events (we call this phenomenon *temporal collision*). Second,

[11] For a comparative analysis of the computational complexity of SAX* and other event detection algorithms see [40] and [27].

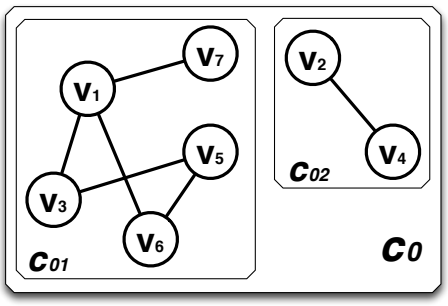


**Fig. 4** Sub-clusters $c_{01}$ and $c_{02}$ extracted from the original cluster $c_0$.

since we use sliding windows, the same event with some slight difference can be captured in partly overlapping windows. The challenge is then to separate different events in different clusters within the same windows, and to merge clusters belonging to the same event in overlapping windows $L_k$ and $L_{k+j}$, where $j < M$. The subsequent Sections 4.2 and 4.3 describe enhancements of our original SAX* algorithms, designed to cope with these two issues.

### 4.2 Splitting clusters of colliding events

To better cope with the problem of temporal collision, with respect to SAX*, we here introduce an additional cluster splitting step .

First, we build a graph $G = (V,E)$ for each cluster $c$ previously detected in a window $L_k$. A graph $G$ is built associating each vertex $v_j \in V$ with a token $w_j$, and adding an edge $(v_j,v_n)$ if tokens $w_j$ and $w_n$:

- co-occurs in a number of documents greater than a threshold $\tau$ (for Twitter and News);
- *or* show "sufficient" semantic similarity (for Wikipedia). Specifically, we use NASARI semantic vectors (see Section 3) to compute similarity between two Wikipedia pages, that must be higher than a threshold *nas*[12].

Next, we detect connected components in $G$. Each connected component is a split of the original cluster, as shown in Figure 4.

For example, the following Wikipedia clicklogs cluster with a peak starting on July $17^{th}$, 2014, refers mainly to the Malaysia Airline Flight 17 crash but also includes a synchronous media event concerning the death of blues legend Johnny Winter peaking during the same days:

*[Surface_to_air_missile, Amsterdam, Kuala_Lumpur, Buk_missile_system,*
*Malaysia_Airlines_Flight_17, Johnny_Winter, Elaine_Stritch,*
*Korean_Air_Lines_Flight_007, Malaysia_Airlines, Boeing_777, Malaysia,*

[12] Experiments have shown that relying on the Wikipedia hyperlink graph does not split clusters effectively.

*Siberia_Airlines_Flight_1812, Edgar_Winter, Malaysia_Airlines_Flight_370]*

After graph splitting, we obtain the following two clusters:

*[Siberia_Airlines_Flight_1812, Surface-to-air_missile, Buk_missile_ system, Malaysia_Airlines_Flight_17, Malaysia_Airlines_Flight_370, Iran_ Air_ Flight_655, Korean_Air_Lines_Flight_007, Malaysia_Airlines, Kuala_Lumpur]*

and

*[Edgar_Winter, Johnny_Winter, Elaine_Stritch]*

### 4.3 Intra-source clustering

Since in SAX* clusters are generated from continuous streams in *sliding windows* $\{L_{t_1}, L_{t_2}, \ldots, L_{t_n}\}$ of equal length and temporal increment $\Delta$, each starting at time $t_k$ ($t_{k+1} - t_k = \Delta$), clusters referring to the same event but extracted in partly overlapping windows may slightly differ, especially for long-lasting events, when news updates motivate new information needs. For example, consider the following three News clusters, generated in three subsequent windows (note also the sliding SAX* strings describing the temporal shape of the event):

*Window $L_k$: July 12-22, ID:C80 [Boeing_777, Surface-to-air_missile, crash, shot, airlin, Hrabove,_Donetsk_Oblast, flight, victim, malaysia, russian, malaysian, Kiev, Amsterdam, Malaysia_Airlines_Flight_17, Malaysia_Airlines, Eastern_Ukraine, Airliner, Buk_missile_system, ...]* **aaaaabbbbb** July 17th 2014

*Window $L_{k+\Delta}$: July 13-23, ID:C0 [malaysian, Malaysia, Hrabove,_Donetsk_Oblast, 2014_pro-Russian_unrest_in_Ukraine, crash, airlin, flight, malaysia, Ukraine, Malaysia_Airlines, Malaysia_Airlines_Flight_17, Kuala_Lumpur, Boeing_777, Russia, Amsterdam, Washington,_D.C., Eastern_Ukraine, ...]* **aaaabbbbbb** July 17th 2014

*Window $L_{k+2\Delta}$: July 14-24, ID:C8 [flight, crash, airlin, Ukraine, Malaysia_Airlines, Malaysia_Airlines_Flight_17, Airline, Russia, Kuala_Lumpur, Airliner, 2014_pro-Russian_unrest_in_Ukraine, Boeing_777, Amsterdam, Government_of_Ukraine, United_States, Barack_Obama, Washington,_D.C., Eastern_Ukraine, Russia-Ukraine_border,...]* **aaabbbbbbb** July 17th 2014

Note that, although the three clusters share many tokens, cluster C80 in window $L_k$ is mostly concerned about the disaster and related entities, while in the subsequent two clusters (C0 in $L_{k+\Delta}$ and C8 in $L_{k+2\Delta}$) new terms, concerning the political debate and involved authorities, gain popularity (e.g., *Government_of_Ukraine, United_States, Barack_Obama, Washington,_D.C.*).

In order to obtain a better characterization of an event, we aggregate similar and temporally adjacent clusters (step 2 of our methodology), forming *meta-clusters*

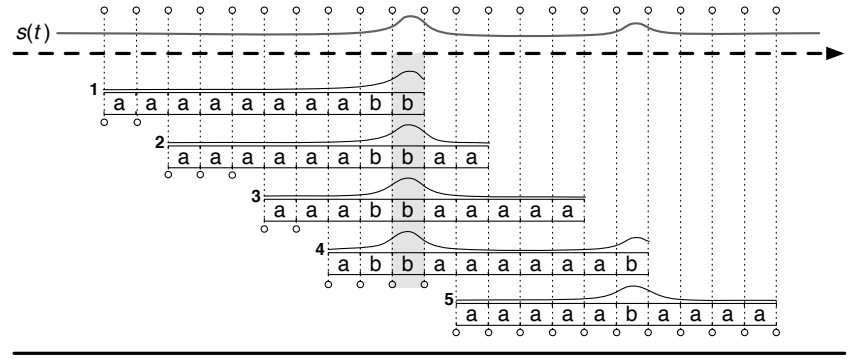


**Fig. 5** SAX* strings associated to a temporal series $s(t)$ in 5 adjacent or overlapping windows.

which contain the most relevant tokens for an event $i$. When considering clusters related to the same events we note that *pivot clusters*, i.e. clusters whose peak day $d$ is closer to the center of the window $L_k$ in which they have been extracted, show a higher precision as compared to those clusters with a peak day closer to the extremes of the window. The problem is illustrated in a sketchy way in Figure 5. The Figure shows a continuous signal $s(t)$ - which we can also interpret as the *centroid* of a set of clustered signals - as captured in 5 different and partly overlapping sliding windows. Although the signal (and the related event) is the same, different SAX* strings are generated in each window, and furthermore, only window 4 captures both peaks. The signal in window 3 is the *pivot* cluster, since it peaks in the center of the considered window [13]. Accordingly, to merge related clusters in adjacent windows, we compute their Jaccard similarity with reference to the pivot cluster. Merged clusters form a *meta-cluster* $m_i^S$, where $S = \{N, T, W\}$ is the source from which the meta-cluster has been extracted. Token *scores* in each meta-cluster are calculated as the normalized ratio between the number of merged clusters in which the token occurs and the number of clusters.

Note that meta-clusters are computed *independently* in each source T, N and W, as also shown in Figure 1. For example, a News meta-cluster for the Malaysia airlines crash is (as before, we omit weights for brevity):

**News Meta-cluster ID: MCN8** *[Ukraine, Malaysia_Airlines, Surface-to-air_missile, Malaysia, Kuala_Lumpur, Eastern_Ukraine, Malaysia_Airlines_Flight_17, Boeing_777, Amsterdam, Airliner, Russia, Government_of_Ukraine, Buk_missile_system, Hrabove,_Donetsk_Oblast, 2014_pro-Russian_unrest_in_Ukraine, Soviet_Union, Kiev, War_in_Donbass, Barack_Obama, United_States, Malaysian, Amsterdam_Airport_Schiphol, Russian_Empire, Jet_airliner, . . . ]*

[13] With reference to previous Malaysia example, the pivot would be cluster C80, since the transition $a \rightarrow b$ is between day 5 and 6 of the 10-days window.

As an additional example, Table 1 shows the resulting meta cluster (always with reference to our Malaysia airlines example), when applying the intra-clustering process to the Twitter stream. The Figure also shows token weights for meta-clusters.

### 4.4 Inter-source cluster alignment

The subsequent phase (step 3 in Figure 1) aligns meta-clusters from the three sources (T, N and W) corresponding to the same popular event. We use as "seeds" the News meta-clusters, and find the most similar meta-clusters from Twitter and Wikipedia. As there might be a slight difference in peak days in different data sources for the same event (news often precede but sometimes follow users' reaction to an event, as shown in [25]), we use a similarity measure *TempSym* with two components: a content-based component and a time-based one. The content-based component is the Jaccard similarity between terms of the meta-clusters, while the time-based component takes into account the distance between the two peak days: the closer the two, the higher the similarity. Considering two meta-clusters $m_a^{S1}$ and $m_b^{S2}$ belonging to two different sources $S1$ and $S2$ (e.g., N and W), we use the following formula:

$$TempSym\left(m_a^{S1}, m_b^{S2}\right) = Jaccard\left(m_a^{S1}, m_b^{S2}\right) \times \alpha^{|peak\left(m_a^{S1}\right) - peak\left(m_b^{S2}\right)|} \quad (1)$$

where the exponent determines the decay coefficient and $peak()$ is the peak day of a meta-cluster.

For example, when using as a seed the news meta-cluster *MCN8* of previous Section 4.3, we find the following alignments:

**Twitter**: *[Malaysia, Aviation_accidents_and_incidents, Malaysia_Airlines, Ukraine, Airline, Malaysia_Airlines_Flight_17, Russia, Airliner, Passenger, Boeing_777, Interfax, Eastern_Ukraine, Kuala_Lumpur, Jet_aircraft, Missile, Amsterdam, Boeing, Reuters, Vladimir_Putin, Aviation, Jet_airliner, United_States, Airplane, CNN, President_of_Russia, Surface-to-air_missile, AirAsia, Barack_Obama, Kiev, United_Kingdom, Government_of_Ukraine, Aircraft, Buk_missile_system, Sky_News, Flight_recorder, BBC, Southwest_Airlines, Terrorism, Carpet_bombing, Altitude, Iran_Air, France, Ministry_of_Internal_Affairs_(Ukraine), USS_Vincennes_(CG-49), . . . ]*

**Wikipedia**: *[Kuala_Lumpur, Siberia_Airlines_Flight_1812, Korean_Air_Lines_Flight_007, Malaysia_Airlines, Boeing_777, Surface-to-air_missile, Malaysia, Buk_missile_system, Malaysia_Airlines_Flight_370, Malaysia_Airlines_Flight_17, Iran_Air_Flight_655, Ukraine, 2014_Crimean_crisis, Pan_Am_Flight_103, Ukraine, Malaysia_Airlines_Flight_17, 2014_pro-Russian_unrest_in_Ukraine, Crimea, Igor_Girkin, Russia, 2014_Russian_military_intervention_in_Ukraine, bermuda_triangle . . . ]*

### 4.5 Generating recommendations

The final phase (step 4) of our workflow in Figure 1 is recommending emergent topics extracted from users' communication (Twitter) and information (Wikipedia)

**Table 1** An excerpt of the Twitter meta-cluster capturing the Malaysia Airlines flight crash event and some excerpts of its composing clusters

| **Clusters** |
|---|
| Window: 10-20 July, ID: C40 [ukrainian, aircraft, airlin, amsterdam, flight, malaysia, malaysian, missil, passeng, plane, condol, Malaysia_Airlines, Malaysia, Airline, Malaysia_Airlines_Flight_17, Aviation_accidents_and_incidents, Ukraine, Amsterdam, Kuala_Lumpur, Russia, Eastern_Ukraine, Interfax, Passenger, Malaysia_Airlines_Flight_370, United_States, Boeing_777, Missile, Airliner, Boeing, Jet_aircraft, Reuters, CNN, Jet_airliner, Vladimir_Putin, Airspace, RussiaUkraine_border, Airplane, Pilot_(aeronautics), ... ] **aaaaaaabbaa** 17 Jul 2014 |
| Window: 11-21 July, ID: C21 [ukrain, aircraft, airlin, amsterdam, condol, flight, malaysia, malaysian, missil, passeng, plane, Malaysia_Airlines, Malaysia, Airline, Malaysia_Airlines_Flight_17, Ukraine, Aviation_accidents_and_incidents, Russia, Kuala_Lumpur, Amsterdam, Interfax, Airliner, Eastern_Ukraine, Passenger, United_States, Boeing_777, Missile, Jet_aircraft, Dubai, Reuters, Emirates_(airline), Boeing, CNN, Vladimir_Putin, Kiev, Barack_Obama, Malaysia_Airlines_Flight_370, ... ] **aaaaaabbaaa** 17 Jul 2014 |
| Window: 12-22 July, ID: C41 [ukrain, aircraft, airlin, amsterdam, condol, malaysia, malaysian, missil, passeng, plane, Malaysia, Aviation_accidents_and_incidents, Ukraine, Malaysia_Airlines, Airline, Russia, Airliner, Passenger, Malaysia_Airlines_Flight_17, Interfax, Boeing_777, Eastern_Ukraine, Jet_aircraft, Kuala_Lumpur, Missile, Aviation, Reuters, Vladimir_Putin, Boeing, Amsterdam, Jet_airliner, Airplane, President_of_Russia, ... ] **aaaaabbaaaa** 17 Jul 2014 |
| Window: 13-23 July, ID:C23 [ukrain, aircraft, airlin, amsterdam, condol, malaysia, malaysian, missil, passeng, plane, Malaysia, Ukraine, Aviation_accidents_and_incidents, Malaysia_Airlines, Airline, Airliner, Russia, Passenger, Malaysia_Airlines_Flight_17, Boeing_777, Jet_aircraft, Missile, Interfax, Eastern_Ukraine, Boeing, Vladimir_Putin, Aviation, Jet_airliner, President_of_Russia, Kuala_Lumpur, Airplane, United_States, Surface-to-air_missile, Amsterdam, Terrorism, CNN, Kiev, Altitude, Iran_Air, ... ] **aaaabbaaaaa** 17 Jul 2014 |
| Window: 14-24 July, ID:C4 [ukrain, aircraft, airlin, amsterdam, condol, malaysia, malaysian, missil, passeng, plane, Malaysia, Aviation_accidents_and_incidents, Ukraine, Malaysia_Airlines, Airline, Russia, Airliner, Passenger, Malaysia_Airlines_Flight_17, Interfax, Boeing_777, Eastern_Ukraine, Jet_aircraft, Missile, Kuala_Lumpur, Reuters, Boeing, Aviation, Jet_airliner, Vladimir_Putin, Amsterdam, Airplane, President_of_Russia, Surface-to-air_missile, CNN, United_States, ... ] **aaabbaaaaaa** 17 Jul 2014 |
| ... |
| ... |

| **Twitter Meta-cluster** |
|---|
| [Malaysia 0.64, Aviation_accidents_and_incidents 0.61, Malaysia_Airlines 0.61, Ukraine 0.59, Airline 0.58, Malaysia_Airlines_Flight 17_0.44, Russia 0.38, Airliner 0.37, Passenger 0.34, Boeing_777 0.24, Interfax 0.23, Eastern_Ukraine 0.22, Kuala_Lumpur 0.20, Jet_aircraft 0.20, Missile 0.19, Amsterdam 0.18, Boeing 0.17, Reuters 0.17, Vladimir_Putin 0.16, Aviation 0.16, Jet_airliner 0.14, United_States 0.14, Airplane 0.13, CNN 0.12, President_of_Russia 0.11, Surface-to-air_missile 0.10, AirAsia 0.09, Barack_Obama 0.09, Kiev 0.09, United_Kingdom 0.07, Government_of_Ukraine 0.07, Aircraft 0.07, Buk_missile_system 0.07, Sky_News 0.07, Flight_recorder 0.07, BBC 0.07, Southwest_Airlines 0.07, Terrorism 0.06, Carpet_bombing 0.06, Altitude 0.06, Iran_Air 0.05, France 0.05, Ministry_of_Internal_Affairs_(Ukraine) 0.05, USS_Vincennes_(CG-49) 0.05, ... ] |

behaviors, as detected by our algorithm, and related to news items. We use aligned meta-clusters to generate real-time recommendations, as follows:

1. Let $d_0$ be the day of news items $N_i$ related to an event $i$ (hereafter we use the symbol $d$ rather than $t$ since, as detailed later in Section 4.6, we use a temporal grain of one day). Let's say that a journalist is interested in analyzing the social impact of the news on day $d_{0+x}$ (for example x=2, two days after). We first retrieve the meta-clusters $m_i^N$ (remember that $i$ is the event index) generated from online news $n_i \in N_i$ in the interval $I : d_0 \leq d \leq d_{0+x}$. Further let $M_i^N(I)$ be the set of such meta-clusters. Note that, if the interval $I$ is large, more than one meta-cluster can be generated reflecting different sub topics of the same event, like e.g., during the Malaysia airlines crash the discussion has turned from a concern for the victims to the Ukranian rebels-Russia dispute about the ownership of BUK missiles.
   We use $M_i^N(I)$ as input query for the recommender;
2. For all $m_i^N \in M_i^N(I)$, we select all aligned meta-clusters $M_i^T(I')$ and $M_i^W(I')$, if any, in the interval $I' : d_{0-x} \leq d_0 \leq d_{0+x}$, since as we said, users may anticipate or follow online news;
3. From the sets $M_i^T$ and $M_i^W$ (we now omit the dependence from $I'$ for simplicity) of retrieved meta-clusters, we present the journalist with the top $K$ ranked items $R_i$ in $M_i$, where the ranking is obtained as explained in Section 4.1 and $K$ is a user-adjustable parameter. The set of recommended items $r_j \in R_i$ is further partitioned in two sets: $R_i^{in\_news}$ and $R_i^{novel}$, where the first are entities also found in news meta-clusters $M_i^N$ and the second represents novel, "unexpected" recommendations.

Finally, note that we generate recommendations starting from news meta-clusters. Although in a real-world scenario journalists could be asked to submit seed terms of their choice for a news item $n_i$ of interest and be recommended with items in the best matching meta-clusters in $T$ and $W$, in our experiments we prefer to avoid subjective choice of news items and seed terms. Using news meta-clusters $M_i^N$ as a starting point implies some noise in the query, since a number of tokens in $M_i^N$ could be unrelated with the considered event, but on the other hand, manually grouping all news items related to the same event $i$ in our large news dataset would have been excessively time-consuming.

As an example of recommended items, on July 18th (one day after the Malaysia event) we obtain:
**Twitter** $R_i^{in\_news}$: *[ ukraine, russia, malaysia_airlines, kuala_lumpur, malaysia_airlines_flight_17, surface-to-air_missile, boeing_777, buk_missile_system, 2014_pro-russian_unrest_in_ukraine, malaysia, crimea, igor_girkin, malaysia_airlines_flight_370 ]*
**Twitter** $R_i^{novel}$**:** *[ministry_of_internal_affairs_(ukraine), southwest_airlines, iran_air, interfax, trans_world_airlines, flight_recorder, jet_aircraft, military_aircraft, buffer_state, carpet_bombing, uss_vincennes_(cg-49)]*

**Wikipedia** $R_i^{in\_news}$: *[ ukraine, malaysia_airlines, malaysia_airlines_flight_17, russia, kuala_lumpur, boeing_777, malaysia, crimea, iran_air_flight_655, buk_missile_system, 2014_pro-russian_unrest_in_ukraine ]*
**Wikipedia** $R_i^{novel}$**:** *[ malaysia_airlines_flight_370, 2014_crimean_crisis,*

*2014_russian_military_intervention_in_ukraine, korean_air_lines_flight_007, bermuda_triangle, siberia_airlines_flight_1812, pan_am_flight_103 ]*

As far as Twitter is concerned, although some of the novel recommended items are not particularly relevant, there are several interesting topics. Although for the sake of space we do not analyze all possibly relevant terms, we note that web articles about Ukraine being a ”buffer state” can be retrieved only well before and well after the Malaysia disaster. Similarly, USS Vincennes (cg-49) refers to the missile that, on July 1988, has accidentally shot the Iran Air Flight 655. The first retrievable web article mentioning this analogy dates October, 2014. Finally, the term *interfax*, apparently unrelated, turned out to be related to the event, since Interfax is a Moscow-based wire agency which reported that Ukrainian rebel forces had the airplane black boxes and they had agreed to hand them over to the Russian-run regional air safety authority.

When considering recommendations extracted from Wikipedia, we note that *analogy* is the main thread. It is not surprising that many people search similar incidents in the past, e.g., Siberia Airlines flight 1812, shot down by the Ukrainian Air Force over the Black Sea in 2001, and other somehow related topics, e.g. *bermuda_triangle*. Finding past similar events is a common information need, frequently highlighted in our data.

In Table 2 we show two additional examples of aligned events and generated recommendations. The considered events are: the celebration of the USA Independence Day and the FIFA 2014 World Cup final match. For each event we show the news meta-cluster, used as seed in the alignment step, and the most similar meta-clusters retrieved in Twitter and Wikipedia. In addition, we mark in bold the *novel* terms in $T$ and $W$ wrt $N$ meta-clusters, which could be candidate recommended terms. Note in the table that some emerging terms, especially in Wikipedia clusters, clearly highlight information needs related to the corresponding events. For example, the emergence of terms like *american_revolutionary_war* and *the_start-spangled_banner* suggests a keen interest to deepen the knowledge of the historical events that led to the US independence and of the US national anthem, respectively. These could be topics that are worth deepening, e.g., in editorials. Looking at Twitter meta-clusters, popular terms like *bbq*, *grill* or *parad* (stem of *parade*) in tweets immediately before Independence Day may simply suggest that most people are preparing to celebrate, while other terms like *pittsburg_steelers* and *heinz_field* refer to co-occurring related sports events and could be reasonably labeled as noise.

Regarding the second event, terms like *20(18|22|26)_fifa_world_cup* in the Wikipedia meta-cluster show a widespread interest in future editions of the football World Cup, which, again, could suggest related topics to be deepened. In the Twitter meta-cluster, the appearance of the term *shakira*, referring to the popular singer, in association with the FIFA football match seems apparently unrelated. However ”googling” the term highlights a strong connection, as the singer sang the theme song of the 2014 World Cup during the FIFA world cup closing ceremony; *ceremoni* is another term in the same cluster, confirming this interpretation. The terms *gerarg* and *argvsger* are popular hashtags used to comment the match on Twitter; while not novel per-se, finding relevant hashtags for an event may prove useful in some contexts.

**Table 2** Examples of aligned meta-clusters and $R_i^{novel}$ recommendations (in bold) for two popular events. As for in Table 1, numbers are token weights.

| **Independence Day (04 Jul 2014)** |
|---|
| *News:* |
| 03 Jul 2014 [united_states_declaration_of_independence 0.25, independence_day 0.24, life,_liberty_and_the_pursuit_of_happiness 0.24, natural_and_legal_rights 0.14, continental_congress 0.13, all_men_are_created_equal 0.12, thomas_jefferson 0.12, self-evidence 0.12, washington,_d.c. 0.12, human_events 0.12, fireworks 0.11, united_states_house_of_representatives 0.10 ... ] |
| *Twitter:* |
| 03 Jul 2014 [independence_day 0.29, textbffourth 0.28, **safe** 0.22, **bbq** 0.16, **grill** 0.16, **sparkler** 0.15, fireworks 0.14, united_states 0.14, **barbecue** 0.13, **parad** 0.12, **coffee** 0.10, god 0.10, **pittsburgh_steelers** 0.10, **heinz_field** 0.09, **canada** 0.09, ... ] |
| *Wikipedia:* |
| Jul 04 2014 [**the_star-spangled_banner** 0.16, independence_day 0.16, **american_revolutionary_war** 0.12] |
| **FIFA World Cup 2014 final match (13 Jul 2014)** |
| *News:* |
| 13 Jul 2014 [germany_national_football_team 0.30, fifa_world_cup_0.27, overtime_(sports)_0.27, argentina 0.26, germany 0.26, argentina_national_football_team 0.24, brazil_0.24, mario_goetze 0.24, rio_de_janeiro 0.23, maracana_stadium 0.22, 2014_fifa_world_cup 0.22, brazil_national_football_team 0.21, lionel_messi 0.21 ... ] |
| *Twitter:* |
| 13 Jul 2014 [**shakira** 0.33, gervsarg 0.29, **kramer** 0.29, **gerarg** 0.29, **argvsger** 0.29, germany_national_football_team 0.29, lionel_messi 0.28, argentina_national_football_team 0.26, argentina 0.24, champion 0.22, germany 0.21, fifa_world_cup 0.21, **neuer** 0.19, **ceremoni** 0.19 ... ] |
| *Wikipedia:* |
| 13 Jul 2014 [**2018_fifa_world_cup** 0.19, **2026_fifa_world_cup** 0.19, **2022_fifa_world_cup** 0.12] |

## 4.6 Parameter tuning and system statistics

A well known limitation of clustering algorithms is the requirement to tune adjustable parameters [31], often including the number $N_c$ of generated clusters. Although SAX* is not parametric in $N_c$ (see Section 4.1), it is nevertheless highly parametric. For parameter setting and sensitivity analysis, in addition to the study already presented in [40], we performed multiple runs using different parameter values for each of the three sources, then we systematically evaluated the quality of resulting clusters computing their Jaccard similarity[14] against 10 known events for which we manually selected about 50 relevant tokens.

The best parameter configurations – under the simplifying hypothesis of uncorrelated parameters – are shown in Table 3. The value of the $\Delta$ parameter (the time granularity) was set to 24 hours (1 day) in all the datasets since this is the minimum available granularity in news, where the exact time of publication is not present. As shown in Table 3, the minimum frequency $f'$ of active tokens in Twitter is much lower, which depends on the fact that we capture only the 1% of total traffic.

[14] And other measures, that we omit for the sake of space.

**Table 3** Parameter settings for the different sources

**LEGEND**: $\Sigma$= dimension of alphabet, $\Delta$= discretization, $f'$= min. frequency of terms, $f''$= min. frequency of entities or hashtags, $\tau$= min token co-occurrence in news and Twitter messages, *nas*= min NASARI similarity between Wikipedia pages, $\alpha$= decay factor, $\gamma$= min cluster similarity in meta-cluster generation.

| Source | $\lvert\Sigma\rvert$ | $\Delta(h)$ | $f'$ | $f''$ | $\tau$ | *nas* | $\alpha$ | $\gamma$ |
|---|---|---|---|---|---|---|---|---|
| Twitter | 2 | 24 | 250 | 50 | 6 | - | 0.1 | 0.75 |
| News | 2 | 24 | 1000 | 50 | 20 | - | 0.1 | 0.75 |
| Wikipedia | 2 | 24 | - | 50000 | - | 0.1 | 0.1 | 0.75 |

**Table 4** Results statistics

| dataset | # clusters | # meta-clusters | av. size meta-clusters |
|---|---|---|---|
| News | 9396 | 829 | 122.46 |
| Twitter | 4737 | 413 | 136.76 |
| Wikipedia | 5450 | 535 | 6.44 |

In Table 4 we show some statistics of the obtained results for the three data sources during the period June-September 2014, using the parametrization of Table 3. Note that, since meta-clusters extracted from Wikipedia include only named entities, their average dimension is much smaller.

## 5 Evaluation

Despite the vast amount of proposed algorithms, the evaluation of recommender systems is very difficult [12]. In particular, if the system is not operational and no real users are available, the quality of recommendations must be evaluated on existing datasets, whose number is limited and what is more, they are focused on specific domains (i.e., music, movies, etc.). This problem is acknowledged as being one of the main obstacles to a wider diffusion of recommenders [15].

We begin with an analysis of evaluation methods and measures proposed in literature (Sections 5.1 and 5.2). Next, in Section 5.3, we describe the experimental protocol adopted in our work. Finally, in Sections 5.4 and 5.5 we present the results of our manifold evaluation experiments.

### 5.1 Methods to evaluate recommender systems

As summarized in [15], the evaluation of recommender systems is performed in one of the following three ways:

1. *Online*, using some available implementations of the system. Online evaluation implies that a system is already available (for example, Amazon [28] and Youtube [7]), which is an uncommon circumstance because companies do not distribute their customers' data, and because many recommender applications are new and therefore there are no implemented systems.

2. With *user studies*, in which a team of users are asked to evaluate comparatively the recommendations produced by several systems, providing a personal judgment. Human evaluation is commonly used in recommenders[15], although it requires a careful design of the experiment to ensure subjectivity. For example, in [23] human evaluators are used in a Twitter recommendation task, and in [30] the users of a crowdsourcing platform are asked to choose a movie recommendation from among five options. A drawback of this method is the forcefully limited number of judgements.
3. *Simulation*, which is an attempt to simulate the judgement of real users. A common simulation method is to "predict the past": although real system users are not available, some information concerning their preferences can be extracted (e.g., from social networks). Previous users' choices are wholly or in part hidden to the recommender, and the evaluation task consists in measuring how well it can predict these past choices. This approach is adopted, for example, in [11] for the task of suggesting news articles to update Wikipedia pages. Given the history of Wikipedia page updates, they extract the list of news references added along a timeline, they train the system in the interval $(t_0, t_j)$ and test if they can predict references introduced at time $t > t_j$. Another approach to simulation consists in "simulating" a human judgment on a recommendation, using some measurable quality criterion. For example, the authors in [35] and [13] define automatically measurable performance metrics, and use these metrics to compare the proposed system with a baseline system.

## 5.2 Evaluation metrics

Concerning *evaluation metrics*, several measures have been adopted in literature (see [22] for a survey). A popular metric is *serendipity*. Serendipity is a more complex notion than "novelty", that we used in Section 4.5. It refers to the ability of a system to generate recommendations that are both *novel* (interchangeably denoted by different scholars as *surprising* or *unexpected*) and *salient* (also denoted as *useful* or *relevant*). In [22] existing approaches are divided into: *component* metrics, measuring different components of serendipity, such as novelty and relevance, and *full* metrics, measuring serendipity as a whole.

Among the proposed *component* metrics, the authors in [44] introduce a novelty metric based on measuring the distance of a recommended item from other items a user has already seen in the past, where the choice of an appropriate distance measure can be made depending on the kind of applications to be evaluated. In [18] two pairwise similarity metrics are used to measure the novelty of a recommendation. The first one is based on point-wise mutual information, where the idea is measuring the similarity of items by counting users who have rated both items and those who rated each item separately. The second one is a content-based measure and is equivalent to the one proposed in [44]. Another content-based measure is proposed in [9] with

[15] A dedicated workshop can be accessed on `http://crowdrecworkshop.org/papers/CrowdRec2015_Proceedings.pdf`

reference to a new task: "entity saliency", that is, to measure if an entity is relevant for a given text document. This problem is related to the one considered in our work, since we wish to determine if entities extracted by $W^3$ are relevant for news items. In [9], the authors model entity saliency as a binary classification problem (salient, not salient). First, they automatically create an annotated corpus, which basically consists in identifying entities in the document that also appear in the abstract: these are considered salient, the others not salient. Then, they propose a method to classify salient/non salient entities using a binary logistic regression model and a set of experimentally chosen features (position of the first mention of an entity, POS tag of entities mentions, etc.). A similar approach is also adopted in [11] in a task of suggesting news items for populating Wikipedia entity pages: here, saliency of entities is estimated as a function of their frequency in news articles, with a decay factor depending on the distance of the positional index of the first occurrence in the text, inspired by the news-specific discourse structure that tends to give short summaries of the most important facts and entities in the opening paragraphs.

A *global* serendipity measure based on the notion of *primitive recommending system* is proposed in [35]. The idea is to arbitrarily choose a primitive (baseline) recommendation strategy that provides low serendipity. The serendipity of a system can be measured as:

$$serendipity(R_u) = \sum_{i \in R_u} max\left(Pr_u(i) - Prim_u(i), 0\right) \cdot rel_u(i)$$

Where $Pr_u(i)$ and $Prim_u(i)$ represent the confidence of recommending an item $i$ of a set of recommended items $R_u$ for, respectively, the evaluated recommender and the primitive recommender, and $rel_u$ is the relevance of the item. The measure can be extended by considering a user rank for each item. In [13] the previous metric is modified by considering only items recommended by the evaluated system and not by the primitive recommender.

In a similar vein, the authors in [41] undertake the task of recommending entities extracted from the KBA 2014 Filtered Stream Corpus. The task is similar to ours since, like for $W^3$, items to be recommended are potentially unlimited, and there is no prior knowledge on users' preferences, therefore ground truth from past choices of the user, or from other similar users, cannot be exploited. They propose the following global serendipity measure:

$$serendipity(e) = \frac{\sum_{e \in UNEXP}\left(rel\left(e\right)\right)}{|UNEXP|}$$

where $e$ is an entity in the set of novel (unexpected) recommendations, and $rel()$ is a measure of its relevance, like in [35].

### 5.3 Outline of W$^3$ Experimental Protocol

Evaluation, either performed by human judgment or automatically, is not easy for our $W^3$ system. As far as manual evaluation is concerned, in Section 4.5 we have

shown that terms intuitively unrelated may turn out to be related upon googling for the considered event, therefore labeling entities for relevance may require careful and time expensive judgment. On the other hand, no ground truth is available. As summarized in Section 5.1, a common stratagem adopted in literature is to artificially create a ground truth, exploiting the "known" future. We verified that a similar approach would be unfeasible for $W^3$, since in many cases relevant entities mentioned in social media and in Wikipedia are never found in subsequent news articles, demonstrating that journalists still lack appropriate methods to analyze readers' informative and communication needs.

To obtain a reliable estimate of $W^3$ performance, we defined the following manifold evaluation protocol, which applies two of the three evaluation methodologies surveyed in previous Sections:

1. **Simulated evaluation**: in analogy with [35] and [13], we define a measure of *saliency* of $R_i^{in\_news}$ and *serendipity* of $R_i^{novel}$ (see Section 4.5) that simulates human judgment, and we compare the performances measured on the full set of extracted recommendations with those of a *primitive recommender*;
2. **Manual crowdsourced evaluation**: we select the top K scored recommendations in $R_i^{novel}$ for 21 world-wide breaking news, and we perform manual evaluation using the *Crowdflower.com* platform, after providing detailed evaluation guidelines to human annotators. We measure the global *serendipity* of $W^3$ recommendations as compared with those produced by the primitive recommender, with blind human judgment on all systems;
3. **Manual evaluation by experts**: we perform a second human evaluation experiment as previously described, but now the evaluators are five journalists from different newspapers.

## 5.4 Simulated Evaluation

We adopt the following protocol: Given an event $i$ first reported on day $d_0$, and related published news $n_i \in N_i$, we generate recommendations $R_i^{in\_news}$ and $R_i^{novel}$ as already explained in Section 4.5, from the set of aligned meta-clusters $M_i^N$ extracted during an interval $I : d_0 \leq d \leq d_{0+x}$, where $d_{0+x}$ is the day in which the query is performed by the journalist. We also generate recommendations using two alternative (primitive) systems, as explained hereafter. Next, we define two measures of saliency and serendipity, and we compare the performance of $W^3$ and the primitive recommender.

### *5.4.1 Primitive recommenders*

We build two *primitive recommenders* (PRs) for Wikipedia and Twitter, which we use as a baseline.

*Wikipedia PR:* The Wikipedia primitive recommender $PR(W)$ is based on finding connected components of the Wikipedia hyperlink page graph (like in [16]), when considering only the topmost visited pages in the interval $I$. More precisely, for each day $d$ in the considered interval $I$, we select the top $H \geq K$ visited entities

(i.e., Wikipedia articles) of the day $E_d^W$. Entities are ranked by frequency of page views[16]. Next, we create clusters $c_j^d$ obtained by extracting the connected components of $E_d^W$ in the Wikipedia hyperlink graph. Let $C^{I'}$ be the set of all clusters $c_j^{I'}$ in $I' : d_0 - x \leq d \leq d_0 + x$. From this set, we select the top $r$ clusters based on the Jaccard similarity with the considered news meta-clusters $M_i^N$. A "primitive" recommendation for event $i$ on day $d_{0+x}$ is the set $PR_i^W$ of topmost $K$ ranked entities in the $r$ previously selected clusters. Like in $W^3$ recommendations, $PR_i^W$ is a ranked list of entities some of which are also found in news, and some others are novel. Note that parameters $H$ and $r$ are not relevant provided that the final number of retrieved entities $|PR_i^W|$ is $\geq K$.
For example, with reference to our Malaysia airlines example, the generated $PR_i^{novel}$ cluster is:

**Wikipedia PR** $PR_i^{novel}$ *[ malaysia_airlines_flight_370, korean_air_lines_flight_007, twa_flight_800, siberia_airlines_flight_1812, history_of_bli, subaru_justy, hamas ]*

The cluster has some items in common with the correspondent $R_i^{novel}$ Wikipedia cluster of Section 4.5, and other items which are clearly unrelated to the news.

*Twitter PR:* The Twitter primitive recommender $PR(T)$ is implemented in the following way: For each token $e \in M_i^N$ we retrieve the top $H \geq K$ co-occurring entities in tweets in the considered interval. We then re-rank and recommend the top $K$ tokens, let $PR_i^T$ be this set. As before, the recommended items in $PR_i^T$ are split into two sets, those which are also found in news and the novel ones.
For the Malaysia example, the generated cluster is:

**Twitter PR** $PR_i^{novel}$ *[ airasia, aviation, jet_aircraft, aircraft, passenger, united_states, hol, earth, fox_news ]*

Note that both primitive recommenders are *far from being naive*. A hyperlink graph to characterize users' intent in Wikipedia search is used in [16] (although the authors use Random Walks rather than connected components analysis to identify related pages). Co-occurrences with top ranked terms in news has been used in [45] to track the evolution and the context around events on Twitter.

### *5.4.2 Generating and Scoring recommendations*

We generate recommendations using four systems: $W^3(T), W^3(W), PR(T)$ and $PR(W)$. The first two recommenders originate from What to Write and Why when applied to Twitter and Wikipedia, respectively. The second two are the primitive recommenders (PR) described in previous Section 5.4.1. *All systems generate their recommendations from the same set of news meta-clusters* $M_i^N$. For all systems, we consider the first $K$ top ranked recommended items, as we said (we recall that $K$ is a user-adjustable parameter).

[16] Note that $E_d^W$ could be straightly used for recommendation, however it would be an excessively rough strategy.

To assess the relevance (*saliency*) of "not novel" recommendations in $W^3$ (and similarly for the other systems), for any recommended item $r_j \in R_i^{in\_news}$ we retrieve all the news $N_i$ related to $M_i^N$ meta-clusters, and compute the *saliency* of $r_j$ as follows:

$$saliency(r_j, n_i) = \beta \times occ^{title}(r_j, n_i) + (1-\beta) \times occ^{snip}(r_j, n_i) \quad (2)$$

where $n_i \in N_i$, $occ^{title}(r_j, n_i)$ is the number of occurrences of $r_j$ in the title of $n_i$, while $occ^{snip}(r_j, n_i)$ is the number of occurrences of $r_j$ in the text snippet of $n_i$ and $\beta$ has been experimentally set to 0.7. The intuition is that recommended items in $R_i^{in\_news}$ *are salient if they frequently occur in the title and text of news snippets*, where occurrences in title have a higher weight. This measure is in analogy to those proposed in [11] and [9]. The global saliency of $r_j$ is then:

$$saliency(r_j) = \frac{\sum_{n_i \in N_i} saliency(r_j, n_i)}{|N_i|} \times IDF(r_j) \quad (3)$$

where $IDF(r_j)$ is the inverse document frequency of $r_j$ in all news of the considered temporal slot, and is used to smooth the relevance of terms with high probability of occurrence in all documents. The average saliency of $R_i^{in\_news}$ is:

$$saliency\left(R_i^{in\_news}\right) = \frac{\sum_{r_j \in R_i^{in\_news}} saliency(r_j)}{\left|R_i^{in\_news}\right|} \quad (4)$$

To provide an estimate of the *serendipity* of novel recommendations, we compute the NASARI similarity (see Section 3) of items $r_k \in R_i^{novel}$ with in-news entities $r_j \in N_i$ and we weight these values with the saliency of $r_j$. The intuition is that *serendipitous recommendations are those concerning topics which have not been discussed so far in online news, but are highly semantically related with highly salient topics in news*:

$$serendipity\left(r_k \in R_i^{novel}\right) = \frac{\sum_{r_k \in R_i^{novel}, r_j \in E_i^N} \left(NASARI(r_k, r_j) \times saliency(r_j)\right)}{\left|R_i^S\right|} \quad (5)$$

Note that this global formulation is not conceptually different from other measures surveyed in Section 5.1, that commonly assign a value to serendipitous recommendations proportionally to their relevance and informativeness, however given the absence of prior knowledge on users' choices, we assume that *semantic similarity with salient entities* in news items is the main clue for relevance[17].

### *5.4.3 Results of Simulated Evaluation*

In Table 5 we summarize the results of our comparative experiments, which we run over the full dataset whose parameters and statistics have been shown in Tables 3 and 4, respectively. We set the maximum number of provided recommendations $K = 10$

[17] We remark however that this measure does not capture *factual* relatedness of novel entities with in-news entities (see e.g., the *shakira* example in Section 4.5). On the other hand, this limitation should affect all compared systems in the same way.

**Table 5** Percentage difference in performances between $W^3$ and PRs on Twitter and Wikipedia

| Source | Saliency | Serendipity | F-Value |
|---|---|---|---|
| Twitter d0 | -28% | +91% | +15% |
| Wikipedia d0 | +172% | +656% | +371% |
| Twitter d2 | -34% | +81% | +8% |
| Wikipedia d2 | +106% | +547% | +286% |

for Wikipedia (where clusters are smaller) and $K = 50$ for Twitter. All recommendations are gathered either the same day ($d_0$) of the first news item on the event $i$, or two days after ($d_2 = d_0 + 2$). In analogy with [35] and [13], we show the percentage difference in performance between $W^3$ and Primitive Recommenders (PRs). Besides *saliencey* and *serendipity*, we also compute the harmonic mean between the two (the $F$ value).

The Table shows that for Wikipedia, $W^3$ outperforms the PR both in saliency and serendipity (it is up to 656% more serendipitous than the baseline) while in Twitter, $W^3$ shows higher serendipity (+91%) but lower salience (-28%). Comparatively, the performance of $W^3$ is much better on Wikipedia than on Twitter, probably due to the limited evidence provided by the 1% available traffic stream. We also noted that two days after the main event (x=2), both serendipity and saliency only slightly decrease showing that newswires have covered only a small portion of users' communication and information needs. Finally, additional experiments with variable $K$ have shown that the distance between our method and the baseline increases with K: for example in Twitter we tested with K=10, 20 and 50 obtaining a growing percentage difference in performance wrt the baseline. This is justified by the fact that both primitive recommenders, as we already remarked, are not naive: it is therefore reasonable that the first top ranked results are fairly good for all systems.

### 5.5 Manual Evaluation

While the saliency of "not novel" recommendations is reasonably assessed by formula (4) (and other similar measures adopted in literature), the serendipity of novel recommended entities is captured by formula (5) only if they are semantically, rather than factually, related with entities also found in news. Back to the example of the 2014 FIFA World Cup event in Section 4.5, formula (5) would likely assign a very low serendipity to the recommended item *Shakira*. For a more accurate estimate of serendipity, we resorted to manual evaluation. We carried out two experiments: the first is based on a popular crowdsourcing platform, *Crowdflower.com*[18], the second is based on the judgment of a restricted team of experts, five journalists from different newspapers.

As detailed in Section 4.5, in automated evaluation we retrieve the set of news items associated with the same event starting from news meta-clusters. This potentially adds some noise, since meta-clusters are error prone and consequently, generated recommendations can be affected by the presence of some unrelated items in

[18] `https://www.crowdflower.com/`

**Table 6** Events selected for Manual Evaluation

| Date | Event |
|---|---|
| 11/06/2014 | Al-Qaeda Faction Seizes Key Iraqi City |
| 14/06/2014 | England vs. Italy at the 2014 World Cup |
| 30/06/2014 | Limiting Rights: Imposing Religion on Workers |
| 05/07/2014 | Wimbledon: Novak Djokovic and Roger Federer Reach Men's Final |
| 06/07/2014 | Neymar Injury |
| 11/07/2014 | Jeremy Lin Acquired by the Los Angeles Lakers from Houston Rockets |
| 12/07/2014 | Tommy Ramone of Punk Rock Pioneers the Ramones Dies Aged 62 |
| 14/07/2014 | Gaza Ceasefire with Hamas |
| 17/07/2014 | Malaysia AirLines Crash |
| 18/07/2014 | Israel Accuses Hamas of Breaking Ceasefire |
| 02/08/2014 | West Africa Ebola Threat |
| 08/08/2014 | Yazidi Kurds Trapped on Mount Sinjar |
| 12/08/2014 | Robin Williams Found Dead |
| 14/08/2014 | Mike Brown Shooting |
| 20/08/2014 | IS Has Beheaded the American Journalist James Wright Foley |
| 24/08/2014 | Earthquake in California |
| 25/08/2014 | Emmys 2014 |
| 28/08/2014 | Ukraine Accuses Russia of Stepping Up Military Activity in Crimea |
| 11/09/2014 | Al-Qaeda Morphs Into a New Movement Since 9/11 |
| 11/09/2014 | Pistorius Arrives at Court for Verdict in Murder Trial |
| 22/09/2014 | Nasa's Newest Mars Mission Spacecraft Enters Orbit Around Mars |

$M_i^N$. However, in the comparative evaluation of previous Section 5.4, all systems were provided with the same set of news $N_i$ and tokens $M_i^N$ for event $i$, therefore possible noise would equally affect all systems.
In manual evaluation, in order to start from a clean representation of each event for all systems, we selected 21 breaking news (i.e., with topmost number of news, tweets and Wikipedia views) in the considered 4-month period, and we manually identified the relevant news items $N_i$ for each event $i$ in an interval $I$ centered on the event peak day $d_0$. The list of events is shown in Table 6. We then automatically extracted the set of relevant meta-clusters $M_i^N$ from these cleaned news items.

### *5.5.1 Crowdsourced Evaluation*

For each of the four systems $W^3(T)$, $W^3(W)$, $PR(T)$ and $PR(W)$ and each event $i$, we generate the first $K = 5$ *novel* recommendations, and we use the *CrowdFlower.com* platform to assess the relevance of these recommendations[19]. Since, as explained in Section 4.5, the task is quite complex, we organized the evaluation as follows: for any event $i$, two/three relevant news items are shown (title and snippet), and For any recommended entity to be evaluated, we also provide the link to the related Wikipedia page, as well as a Google link to a query with the following structure:

$$seed\_entity\_n_i + novel\_recommended\_entity + date\_of\_event$$

where $seed\_entity\_n_i$ is the first ranked entity in $M_i^N$. Google links are useful for evaluating otherwise not obvious factual relationships between an event and the considered entity, such as, e.g., *Interfax* in relation to the Malaysia air crash (see the

[19] We are only interested in evaluating novelty, since the saliency of $R_i^{in\_news}$ is reasonably assessed by formula (2)

discussion in Section 4.5). The evaluator can verify for factual relatedness inspecting the results of the query: *Malaysia Airline MH17 + Interfax + July 16 17 18 2014*. For each news item, annotators are asked to decide whether a recommended entity IS or IS NOT relevant with reference to the reported news ("not sure" is also allowed). "Relevant" means that either the entity is semantically related (e.g., a similar entity related to past events) to the domain of the news, like e.g., *Daniel Pearl* in relation to the assassination by IS of *James Wrigth Foley*, or that it is factually related, like for the previously discussed entity *Interfax* in relation with the Malaysia air crash. Recommended items are listed in random order of relevance and source recommender system, and clearly, this information is not shown to evaluators. We paid each task (consisting in the evaluation of three news items) $0.25, and we prepared a number of test questions, in order to guarantee a high quality of annotators. Crowdflower.com assigns "weights" to annotators depending on their trust on previous tasks and performance on test questions, and these weights are used to generate the final judgment on each item.

The task was run on April 23rd, 2017, and we collected 1344 total judgments[20]. To compute the performance of each system, we use the Mean Average Precision (MAP) [32], which takes the rank of recommendations into account. Rather than averaging over the full space $m$ of recommendable items (which is unknown), we follow the common practice[21] of setting $m$ equal to the set of items proposed by all compared systems, that annotators considered correct.

The results of this experiment are reported in Table 7, which shows, in agreement with the automated evaluation experiment of Table 5, a superiority of $W^3$. The Table also confirms that the difference between $W^3$ and the primitive recommender is much higher for Wikipedia than for Twitter. We further note that the absolute performance of the recommender is higher in Twitter, which is not in contradiction with Table 5, since here we are focusing on world-wide high impact news, those for which our 1% Twitter stream provides sufficient evidence to obtain very clean clusters. On the other hand, the almost tripled performance figures of $W^3$ with respect to the Wikipedia PR are mostly due to the fact that the primitive recommender, like the algorithm proposed in [16], exploits a static structure (the Wikipedia hyperlink graph), therefore its novelty is inherently limited.

To compute inter-rater agreement, we used data provided by Crowdflower on the inter-rater agreement of single annotators, and we averaged all annotators, obtaining a global score of 89.2%.

Upon a more in-depth analysis of the evaluation results, we found that in many cases both systems present reasonable and similar recommendations, e.g.:

**News:** ***Earthquake in Napa, California***
$W^3(T)$: $earthquake\_prediction, tsunami\_warning\_system, vallejo, \_california, vineyard, san\_francisco\_bay$

[20] For a comparison, in [23], one of the largest reported crowdsourced evaluation experiments on recommenders, 1602 judgments have been collected from 17 computer science students.

[21] `http://sdsawtelle.github.io/blog/output/mean-average-precision-MAP-for-recommender-systems.html`

**Table 7** MAP of compared systems

| Source | $W^3$ | PR |
|---|---|---|
| Twitter | 0.716 | 0.604 |
| Wikipedia | 0.746 | 0.232 |

**Table 8** MAP of compared systems (Evaluation by journalists)

| Source | $W^3$ | PR |
|---|---|---|
| Twitter | 0.811 | 0.797 |
| Wikipedia | 0.719 | 0.316 |

$PR(T)$: $vallejo, \_california, san\_francisco\_bay, west\_napa\_fault,$
$hayward\_fault\_zone, california\_wine$

however, recommendations from $W^3$ are often more "interesting" and accurate, like:

**News:** ***Pistorius Arrives at Court for Verdict in Murder Trial***
$W^3(T)$: $common\_law, apartheid, homicide, life\_imprisonment$
$PR(T)$: $homicide, bail, september\_11\_attacks, academy\_awards$

**News:** ***Limiting Rights: Imposing Religion on Workers***
$W^3(T)$: $constitutionality, federal\_government\_of\_the\_united\_states,$
$ruth\_bader\_ginsburg, jeffrey\_toobin, religious\_persecution$
$PR(T)$: $samuel\_alito, supreme\_court, corporate\_personhood, lawsuit$

**News:** ***IS behead an American journalist James Wright Foley***
$W^3(W)$: $abu\_bakr\_al-baghdadi, islamic\_state, daniel\_pearl, al-qaeda, islam$
$PR(W)$: $dumbarton\_oaks\_conference, thelma\_ritter, isis,$
$shooting\_of\_michael\_brown, deaths\_in\_2014$

### *5.5.2 Evaluation by Experts*

An advantage of crowdsourced evaluation is the relatively large number of collected independent judgments, however, despite the presence of filters to identify and remove inaccurate judges, the quality of evaluation might be lower than expected when evaluators are domain experts. On the other hand, finding domain experts is not always easy. We found 5 journalists from five different newspapers who volunteered to perform a manual evaluation, using the same data, information and platform used for the crowd-sourced evaluation.

The results in Table 8 show that the judgment obtained by few experts is not strikingly different (but hopefully more accurate) from that obtained by many crowdsourced annotators, although slightly better for all systems. In this experiment, the global inter-rater agreement was 82.7%, measured by assigning all journalists the same weight.

## 6 Conclusions and Future Work

In this paper we presented a methodology, named $W^3$, to recommend serendipitous entities to journalists, based on the detection and analysis of readers' information needs on Wikipedia and their communication needs on Twitter. Although our data span a 4-month period, the methodology is general and not limited by the dimension of the data, thanks to the use of an efficient temporal clustering algorithm.

Experiments suggest that $W^3$ succeeds in discovering patterns of interest with reference to highly popular events in all three analyzed information sources: online news, Twitter and Wikipedia. In the future, we plan to perform additional experiments to classify frequent patterns of information and communication needs in relation to event types, exploiting categories in the Wikipedia Category Graph.

## Acknowledgements

This work has been partially supported by the MIUR under grant "Dipartimenti di eccellenza 2018-2022" of the Department of Computer Science of Sapienza University and by the IBM Faculty Award #2305895190.

Finally, we would like to thank SpazioDati [22] and Textrazor[23] for supporting this research by granting extensive access to their APIs.

## References

1. Bi, B., Ma, H., Wang, K., Hsu, B., Cho, J., Chu, W.: Learning to recommend related entities to search users. In: Proc. of WSDM'15, pp. 177–186. ACM (2015)
2. Blanco, R., Cambazoglu, B.B., Mika, P., Torzec, N.: Entity recommendations in web search. In: Proc. of ISWC'13, pp. 33–48. Springer (2013)
3. Brooker, R., Schaefer, T.: Public Opinion in the 21st Century: Let the People Speak? New directions in political behavior series. Houghton Mifflin Company (2005)
4. Camacho-Collados, J., Taher Pilehvar, M., Navigli, R.: Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. Artificial Intelligence **240**, 36–64 (2016)
5. Cheekula, S.K., Kapanipathi, P., Doran, D., Jain, P., Sheth, A.: Entity recommendations using hierarchical knowledge bases. In: Proc. of Know@LOD 2015, vol. 1365. CEUR-WS (2015)
6. Cordeiro, M., Gama, J.: Online Social Networks Event Detection: A Survey, pp. 1–41. Springer International Publishing (2016)
7. Davidson, J., Liebald, B., Liu, J., Nandy, P., Van Vleet, T., Gargi, U., Gupta, S., He, Y., Lambert, M., Livingston, B., et al.: The youtube video recommendation system. In: Proc. of RecSys'10, pp. 293–296. ACM (2010)
8. Diakopoulos, N., De Choudhury, M., Naaman, M.: Finding and assessing social media information sources in the context of journalis. In: Proc. of ACM CHI'12, pp. 24,151–2460. ACM (2012)
9. Dunietz, J., Gillick, D.: A new entity salience task with millions of training examples. In: Proc. of EACL'14, pp. 205–209. ACL (2014)
10. Ferragina, P., Scaiella, U.: Tagme: On-the-fly annotation of short text fragments (by wikipedia entities). In: Proc. of CIKM'10, pp. 1625–1628. ACM (2010)

[22] htttp://spaziodati.eu

[23] http://textrazor.com

11. Fetahu, B., Markert, K., Anand, A.: Automated news suggestions for populating wikipedia entity pages. In: Proc. of CIKM'15, pp. 323–332. ACM (2015)
12. Fouss, F., Saerens, M.: Evaluating performance of recommender systems: An experimental comparison. In: Proc. of WI-IAT'08, vol. 1, pp. 735–738. IEEE (2008)
13. Ge, M., Delgado-Battenfeld, C., Jannach, D.: Beyond accuracy: Evaluating recommender systems by coverage and serendipity. In: Proc. of RecSys'10, pp. 257–260. ACM (2010)
14. Gloviczki, P.J.: Journalism in the Age of Social Media, pp. 1–23. Palgrave Macmillan US (2015)
15. Gunawardana, A., Shani, G.: A survey of accuracy evaluation metrics of recommendation tasks. Journal of Machine Learning Research **10**, 2935–2962 (2009)
16. Hu, J., Wang, G., Lochovsky, F., Sun, J., Chen, Z.: Understanding user's query intent with wikipedia. In: Proc. of WWW'09, pp. 471–480. ACM (2009)
17. Jain, A.: Data clustering: 50 years beyond k-means. Pattern Recogn. Lett. **31**(8), 651–666 (2010)
18. Kaminskas, M., Bridge, D.: Measuring surprise in recommender systems. In: P. Adamopoulos (ed.) Proc. of the Work. REDD'14. ACM (2014)
19. Kille, B., Hopfgartner, F., Brodt, T., Heintz, T.: The plista dataset. In: Proc. of NRS'13, pp. 16–23. ACM (2013)
20. Knight, M.: Journalism as usual: The use of social media as a newsgathering tool in the coverage of the iranian elections in 2009. Journal of Media Practice **13**(1), 61–74 (2012)
21. König, A.C., Gamon, M., Wu, Q.: Click-through prediction for news queries. In: Proc. of SIGIR '09, pp. 347–354. ACM (2009)
22. Kotkov, D., Wang, S., Veijalainen, J.: A survey of serendipity in recommender systems. Knowledge-Based Systems **111**, 180–192 (2016)
23. Krestel, R., Werkmeister, T., Wiradarma, T.P., Kasneci, G.: Tweet-recommender: Finding relevant tweets for news articles. In: Proc. of WWW'15, pp. 53–54. ACM (2015)
24. Kuzey, E., Vreeken, J., Weikum, G.: A fresh look on knowledge bases: Distilling named events from news. In: Proc. of CIKM'14, pp. 1689–1698. ACM (2014)
25. Lehmann, J., Gonçalves, B., Ramasco, J.J., Cattuto, C.: Dynamical classes of collective attention in twitter. In: Proc. of WWW'12, pp. 251–260. ACM (2012)
26. Leskovec, J., Backstrom, L., Kleinberg, J.: Meme-tracking and the dynamics of the news cycle. In: Proc. of KDD '09, pp. 497–506. ACM (2009)
27. Lin, J., Keogh, E., Lonardi, S., Chiu, B.: A symbolic representation of time series, with implications for streaming algorithms. In: Proc. of DMKD'03, pp. 2–11. ACM (2003)
28. Linden, G., Smith, B., York, J.: Amazon. com recommendations: Item-to-item collaborative filtering. IEEE Internet computing **7**(1), 76–80 (2003)
29. Lommatzsch, A., Albayrak, S.: Real-time recommendations for user-item streams. In: Proc. of SAC'15, pp. 1039–1046. ACM (2015)
30. Maccatrozzo, V., Aroyo, L., Van Hage, W.: Crowdsourced evaluation of semantic patterns for recommendations. In: Proc. of UMAP'13 (2013)
31. Magland, J.F., Barnett, A.H.: Unimodal clustering using isotonic regression: ISO-SPLIT. ArXiv e-prints:1508.04841 (2015)
32. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press (2008)
33. Mishra, A., Berberich, K.: Leveraging semantic annotations to link wikipedia and news archives. In: Proc. of ECIR'16, pp. 30–42. Springer (2016)
34. Mongiovì, M., Bogdanov, P., Singh, A.K.: Mining evolving network processes. In: Proc. of ICDM'13, pp. 537–546. IEEE (2013)
35. Murakami, T., Mori, K., Orihara, R.: Metrics for evaluating the serendipity of recommendation lists. In: Proc. of the 2007 Conf. on New Frontiers in AI, pp. 40–46. Springer (2008)
36. Osborne, M., Petrovi, S., Mccreadie, R., Macdonald, C., Ounis, I.: Bieber no more: First story detection using twitter and wikipedia. In: SIGIR 2012 Workshop on Time-aware Information Access (2012)
37. Phelan, O., McCarthy, K., Smyth, B.: Using twitter to recommend real-time topical news. In: Proc. of RecSys '09, pp. 385–388. ACM (2009)
38. Scaiella, U., Prestia, G., Del Tessandoro, E., Verì, M., Barbera, M., Parmesan, S.: Datatxt at microposts2014 challenge. In: Proc. of Work. MSM'14, pp. 66–67. ACM (2014)
39. Steiner, T., Van Hooland, S., Summers, E.: Mj no more: Using concurrent wikipedia edit spikes with social network plausibility checks for breaking news detection. In: Proc. of WWW'13, pp. 791–794 (2013)

40. Stilo, G., Velardi, P.: Efficient temporal mining of micro-blog texts and its application to event discovery. Data Min. Knowl. Discov. **30**(2), 372–402 (2016)
41. Tran, T., Niederée, C., Kanhabua, N., Gadiraju, U., Anand, A.: Balancing novelty and salience: Adaptive learning to rank entities for timeline summarization of high-impact events. In: Proc. of CICM'15, vol. 19, pp. 1201–1210. Springer (2015)
42. Tsagkias, E., de Rijke, M., Weerkamp, W.: Predicting the volume of comments on online news stories. In: Proc. of CIKM'09. ACM (2009)
43. Tsagkias, M., de Rijke, M., Weerkamp, W.: Linking online news and social media. In: Proc. of WSDM'11, pp. 565–574. ACM (2011)
44. Vargas, S., Castells, P.: Rank and relevance in novelty and diversity metrics for recommender systems. In: Proc. of RecSys'11, pp. 109–116. ACM (2011)
45. Weiler, A., Grossniklaus, M., Scholl, M.: Event identification and tracking in social media streaming data. In: Proc. of the Work. of the EDBT/ICDT'14, pp. 282–287. CEUR-WS (2014)
46. Yang, J., Leskovec, J.: Patterns of temporal variation in online media. In: Proc. of WSDM '11, pp. 177–186. ACM (2011)
47. Yoshida, M., Arase, Y., Tsunoda, T., Yamamoto, M.: Wikipedia page view reflects web search trend. In: Proc. of WSC'15, pp. 65:1–65:2. ACM (2015)
48. Zubiaga, A., Ji, H., Knight, K.: Curating and contextualizing twitter stories to assist with social newsgathering. In: Proc. of IUI'13, pp. 213–224. ACM (2013)

## A Appendix - List of symbols

| Symbol | Description |
|---|---|
| $s(t)$ | temporal series associated to a token |
| $L_k$ | temporal window $k$ |
| $M$ | # of equal length partitions in $L_k$ |
| $\Delta$ | length of a partition $M$ |
| $L$ | length of $L_k$, i.e. $L = M \times \Delta$ |
| $\Sigma$ | alphabet of symbols used in SAX strings |
| $N, T, W$ | News, Twitter and Wikipedia datasets |
| $N_i$ | set of news items related to an event $i$. $N_i \subset N$ |
| $n_i$ | news item $\in N_i$ |
| $M_i^S$ | set of *meta-clusters* related to the event $i$ extracted from the dataset $S \in \{N,T,W\}$ |
| $m_i^S$ | *meta-cluster* $\in M_i^S$ |
| $R_i^S$ | set of recommendations for the event $i$ selected from $m_i^S$, where $S \in \{T,W\}$ |
| $R_i^{in\_news}$ | subset of recommendation also found in news meta-clusters $M_i^N$. $R_i^{in\_news} \subseteq R_i$ (the source $S$ is omitted for simplicity) |
| $R_i^{novel}$ | subset of novel recommendation. $R_i^{novel} \subseteq R_i$ |