Studies on Interactive Event Detection and Labeling from Timestamped Texts

Corentin Forler[1,†], Előd Egyed-Zsigmond[1,*,†]

[1] Institut National des Sciences Appliquées de Lyon (INSA Lyon), 20 av. Albert Einstein, Villeurbanne, 69100, France

Abstract

In this paper we present a work on interactive event detection, visualization and labeling. We started from an unsupervised one-shot event detection system created by A. Guille and A.-C. Favre and modified it to optimize its interactive usage. We also added a new event labeling approach to enable a more meaningful event description for the human users, by selecting representative documents, in addition to the keywords extracted by the original method. The source code of our work is available online at https://github.com/CorentinForler/mabed-interactive-labeling.

Keywords

dataset creation, event detection, event labeling, text mining

1. Introduction

Detecting events in a timestamped text stream is getting more attention each year. In the literature, there are several definitions of events and several event detection methods [1]. There are methods to detect predefined events. Some [2, 3] define a set of keywords or real world facts, actions to describe events and try to detect their occurrences in the text stream. In this case, an event is a punctual fact, it occurs at a given time instance (that can be more or less precise: from milliseconds to weeks). Examples of such events can be found in the stock market domain: company acquisitions, fusions, or in the civil security domain: earthquakes, tornados. Under this approach, an event can be described by a single timestamp. There are other event types that last longer and whose effects on the text stream can be described in terms of impact. These events have amplitudes that evolve over time, requiring other detection and tracking methods.

When the events to detect are undefined, the problem of their detection changes. Indeed, these techniques must then detect "any" topic that emerges in the text flow and determine whether it can be considered as an event or not and, if so, how to describe it [4, 5, 6]. The work of Guille and Favre [7] tackles this problem. They propose a system that detects events in a text stream collected from social media (Twitter) based on the detection of anomalies in the frequency of user mentions. The input of their system is made of timestamped tweets. The

CIRCLE (Joint Conference of the Information Retrieval Communities in Europe) 2022

* Corresponding author.
† These authors contributed equally.
corentin.forler@insa-lyon.fr (C. Forler); elod.egyed-zsigmond@insa-lyon.fr (E. Egyed-Zsigmond)
0000-0002-1218-8026 (E. Egyed-Zsigmond)

events are described by a set of primary and secondary keywords and a time-based impact score histogram. This histogram describes, in a way, the life cycle of the event.

Our work focuses on two extensions of the method of Guille and Favre: its optimization for an interactive use, and an event labeling by full-text fragments, instead of keywords, for a better understanding. Indeed, one of the limitations of the original method is the need to precise the number of events to detect. Our system proposes an interactive interface to adjust in an assisted manner this number. We also extended the method to detect events from any timestamped text, not only tweets like the original method. In this paper, we will present our event detection interface, emphasizing the improvements we made on the original method to decrease its response time in an interactive event detection use case. We then present our label extraction approach having as objective selecting the most representative text(s) for a given event, described by keywords and an impact histogram.

2. Approach and Implementation

We did a first implementation of the method described in [7], called MABED, using the code available on GitHub[1]. The method requires several parameters, such as: the basic time-slice length (*tsl*) to consider as atomic[2], the number of events to extract (*ec*), the number of primary and secondary keywords to extract for each event (*kwc*). The original method was created to run once in a command-line environment. We implemented an interactive interface and optimized its execution to reduce calculation times when changing the parameters.

2.1. Dataset

The chosen dataset is a list of 238 872 news articles, collected using RSS feeds of mainly economy-related English language newspapers, spanning from 2020-01-01 to 2020-09-30 (9 months), with a vocabulary amounting to 25 097 distinct words/tokens. This dataset was then used to extract 50 events, using the following parameters:

- **length of each time-slice *tsl* :** 24 hours (1440 minutes)
- **number of events to extract *ec* :** 50
- **number of keywords to extract *kwc* :** 10

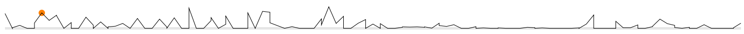
Once events were computed, the event labeling step was then run. Example events produced by MABED are:

- Event 1
  - **Begin/End date :** 2020-03-16 – 2020-06-04
  - **Impact over time :** 
  - **Main terms :** global, pandemic, health, due, covid-19, amid, economic

[1]https://github.com/AdrienGuille/pyMABED

[2]The *tsl* parameter controls the temporal precision of the algorithm, because the input dataset is split into *time-slices* of equal duration of *tsl* hours (or minutes, seconds,... unit to choose).

  - **Related terms :** package, trillion, ministry, postponed, impact, closed, concerns, confirmed, demand, deaths
  - **Human-made description :** On March 25, 2020, the U.S. Senate passed a $2 trillion coronavirus economic relief package.
  - **Selected article :** U.S. senators were set to vote on Wednesday on a $2 trillion bipartisan package of legislation to alleviate the devastating economic impact of the coronavirus pandemic.

- Event 33
  - **Begin/End date :** 2020-03-13 – 2020-07-29
  - **Impact over time :** 
  - **Main terms :** order, gov
  - **Related terms :** andrew, issued, cuomo, stay-at-home, executive, businesses, spread, york, court, health
  - **Human-made description :** On March 20, 2020, New York governor Andrew Cuomo announced a stay-at-home executive order. It ended on May 15, 2020.
  - **Selected article :** New York Gov Andrew Cuomo is ordering all workers in nonessential businesses to stay home and banning gatherings statewide.

2.2. User interface

We created a new user interface for the method provided in [7]. Ours is composed of three parts, and is based on a client-server architecture, where the front-end uses JavaScript libraries such as `D3.js` and `billboard.js`, and the back-end uses Python 3 and the Flask web server framework. We added a web-form in order to change the parameters.

A simple form is shown to the user. While the event detection algorithm can be adjusted with various parameters, most of them could be difficult to explain in the interface, and some are only used to fine tune the results. Therefore, we chose to emphasize three parameters: the duration of each time-slice, the number of events to detect, and the maximum number of words to describe an event.

The form is shown on Figure 1. It includes a button to start the computation once the desired parameters are entered. Below the form, and once a computation is done, the **table of events**, depicted in Figure 2, is presented to the user.

Its first column, titled *“Event Terms”*, contains main and secondary terms of the event given by the original algorithm (the two kinds of terms are styled differently and secondary terms (in brown) are sorted by decreasing *magnitude*). The second column, titled *“Proposed Articles”*, contains a list of proposed descriptions for each event. Each description can be marked *correct* or *incorrect* by clicking on it, initially being in a neutral state (not shown on Figure 2). Moreover, each description can be moved up or down among the proposed descriptions of a given event, effectively allowing the users to rank the descriptions. These behaviors have been added for an easier data annotation during the experimental evaluation of the system, and won’t be shown to the final users. The third and last column, titled *“Event Impact”*, contains a graph showing the impact of the event over time.

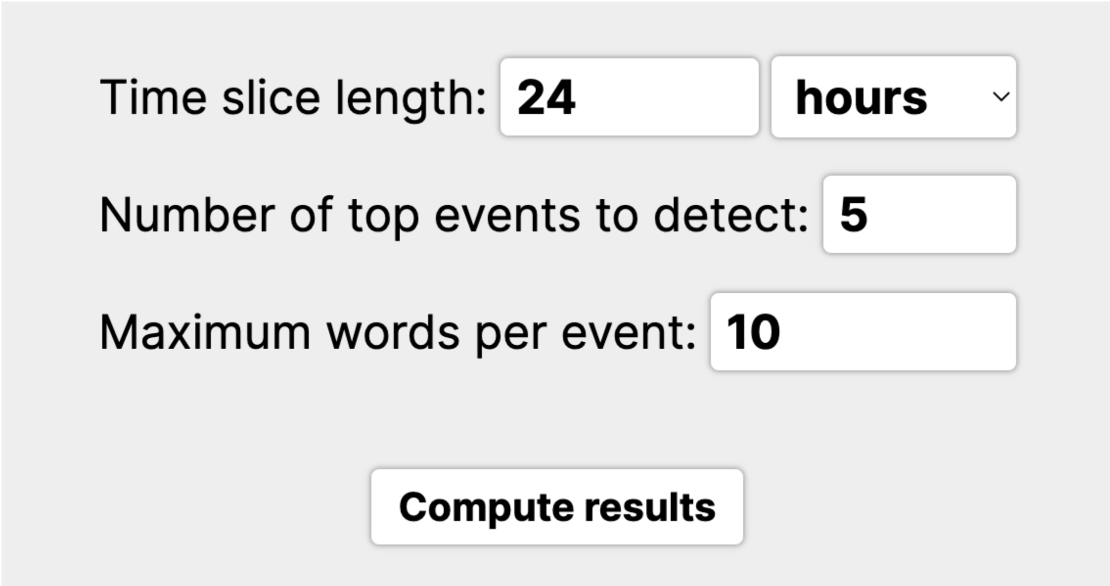


Figure 1: Form controls to change the MABED parameters.

The user can, at any time, change the values and click on the "*Compute results*" button to update the table of events and the graphs. This work on the user interface is supported by a set of improvements to the original MABED implementation.

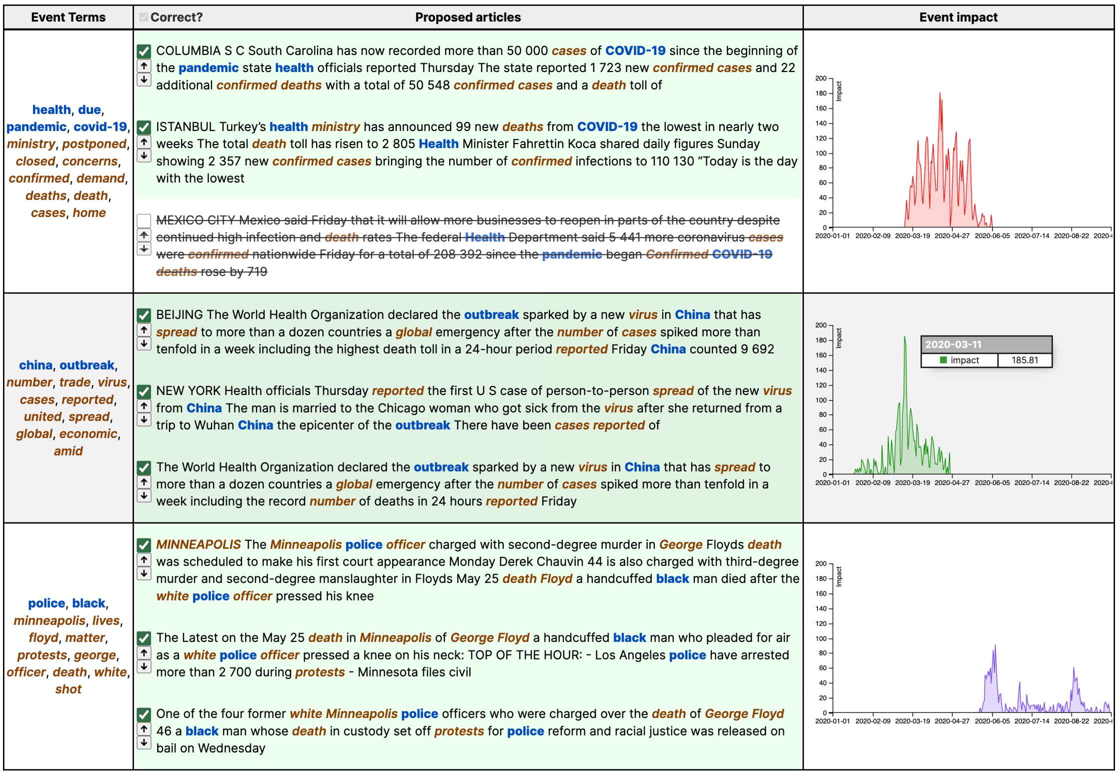
2.3. Optimizations and improvements

Given that our goal is to provide an interactive event visualization service, we had to make the whole system faster, especially for consecutive runs where few parameters change. While we added many improvements to the original method, which we are going to present in the following sections, we mostly improved the response time of the service with a cache system.

2.3.1. Cache system

In the initial MABED implementation, intermediary computations are not permanently stored, therefore all executions take the same time (when given the same parameters). To allow faster successive computations, we needed to store the results of these intermediary steps. We designed a cache system, allowing to keep these results instead of discarding them. For the same dataset and the same parameters, the computation time is zero, and only the parsing time of the cache files is required. If some parameters change, only the computations that depend on them need to be carried out again.

To measure the effect of this optimization, we performed a series of experiments whose results are presented in Table 1. First, we did three unoptimized runs that do not use the cache system, labeled *“from scratch”*. In the initial version of MABED, all runs would have lasted the same duration as these unoptimized computations.



Event Terms	Correct?	Proposed articles	Event impact
health, due, pandemic, covid-19, ministry, postponed, closed, concerns, confirmed, demand, deaths, death, cases, home	☑ ☑ ☐	COLUMBIA S C South Carolina has now recorded more than 50 000 cases of COVID-19 since the beginning of the pandemic state health officials reported Thursday The state reported 1 723 new confirmed cases and 22 additional confirmed deaths with a total of 50 548 confirmed cases and a death toll of ISTANBUL Turkey's health ministry has announced 99 new deaths from COVID-19 the lowest in nearly two weeks The total death toll has risen to 2 805 Health Minister Fahrettin Koca shared daily figures Sunday showing 2 357 new confirmed cases bringing the number of confirmed infections to 110 130 "Today is the day with the lowest ~~MEXICO CITY Mexico said Friday that it will allow more businesses to reopen in parts of the country despite continued high infection and death rates The federal Health Department said 5 441 more coronavirus cases were confirmed nationwide Friday for a total of 208 392 since the pandemic began Confirmed COVID-19 deaths rose by 719~~	
china, outbreak, number, trade, virus, cases, reported, united, spread, global, economic, amid	☑ ☑ ☑	BEIJING The World Health Organization declared the outbreak sparked by a new virus in China that has spread to more than a dozen countries a global emergency after the number of cases spiked more than tenfold in a week including the highest death toll in a 24-hour period reported Friday China counted 9 692 NEW YORK Health officials Thursday reported the first U S case of person-to-person spread of the new virus from China The man is married to the Chicago woman who got sick from the virus after she returned from a trip to Wuhan China the epicenter of the outbreak There have been cases reported of The World Health Organization declared the outbreak sparked by a new virus in China that has spread to more than a dozen countries a global emergency after the number of cases spiked more than tenfold in a week including the record number of deaths in 24 hours reported Friday	2020-03-11 impact 185.81
police, black, minneapolis, lives, floyd, matter, protests, george, officer, death, white, shot	☑ ☑ ☑	MINNEAPOLIS The Minneapolis police officer charged with second-degree murder in George Floyds death was scheduled to make his first court appearance Monday Derek Chauvin 44 is also charged with third-degree murder and second-degree manslaughter in Floyds May 25 death Floyd a handcuffed black man died after the white police officer pressed his knee The Latest on the May 25 death in Minneapolis of George Floyd a handcuffed black man who pleaded for air as a white police officer pressed a knee on his neck: TOP OF THE HOUR: - Los Angeles police have arrested more than 2 700 during protests - Minnesota files civil One of the four former white Minneapolis police officers who were charged over the death of George Floyd 46 a black man whose death in custody set off protests for police reform and racial justice was released on bail on Wednesday	

Figure 2: Table of events with event terms, proposed descriptions and impact graphs. The table is updated according to changes the user made to the various parameters available.

Then, we ran a single *"Baseline"* computation, storing its results in an initially empty cache, using the following parameters: *tsl* of **1440** minutes (24 hours), *ec* of **10** events, *kwc* of **10** words. The run labeled *"Baseline (cached)"* shows that running the same computation again, using the cache, is very fast.

We then performed three computations, labeled *"With cache"*, each run with a cache containing only the results of the **Baseline** run, potentially reusing intermediary results, each one can be compared to its corresponding *"From scratch"* run duration. The runs labeled *"—"* show the speed gains when running with other parameter values. The values of parameters are shown in columns: *tsl*, *ec* and *kwc*.

With this cache system, computations could be performed *off-line*, allowing users to change parameters while benefiting form cached results.

2.3.2. Reduced number of file operations

Another optimization is the reduction of the number of file openings/closures during the discretization step. Time-slice files were opened and closed *for each document*, but, after implementing this optimization, the files are kept open as long as possible thanks to a circular buffer system (at most 512 files are kept open by the buffer to comply to the limitations imposed

Table 1
The changed parameter is highligthed. *"Duration"* is the runtime of the request with the given parameters.

Computation	*tsl*	*ec*	*kwc*	**Duration**
Baseline	1440	10	10	5m 5s
Baseline (cached)	1440	10	10	191ms
From scratch	**28 800**	10	10	4m 16s
Using cache	**28 800**	10	10	4m 5s
—	**2880**	10	10	4m 29s
—	**1439**	10	10	4m 53s
From scratch	1440	**30**	10	6m 22s
Using cache	1440	**30**	10	2m 19s
—	1440	**9**	10	48s
—	1440	**5**	10	30s
From scratch	1440	10	**30**	5m 23s
Using cache	1440	10	**30**	1m 9s
—	1440	10	**9**	50s
—	1440	10	**5**	48s

by the operating system, see `ulimit open files`). On a database composed of 20 000 documents, where 14 time-slices were needed, those files are opened/closed only once, at the beginning/end of the discretization step, saving 1.5 seconds on the total runtime.

2.3.3. Various other improvements

- Multi-threading to speed up compatible computations
- Automatic CSV format detection (separator, date/time format, column names)
- Compatibility with date-only or date-time timestamps

2.4. Labeling methods

We noticed that the keywords produced by the original MABED implementation to describe an event were often difficult to understand, as they covered several contexts. Therefore, we decided to label events with full-text excerpts (short descriptions of existing press articles) in order to enable an easier understanding by users. In order to do this we implemented a keyword based document search method to find the best matching texts according to the keywords generated to describe the event.

On the dataset were run 4 different methods to find descriptions for the detected events, used to query and rank potential descriptions (documents taken from the main dataset, for instance) and pick the best one(s) to describe a specific event.

2.4.1. SKC: Simple keyword counting

With this method, the score of a document is simply the number of event terms that appear at least once in the document. Repetitions of a word do not increase the score of a document. Let $\mathbf{T}_{e,d}$ the set of main and secondary terms of the event e ($t \in \mathbf{T}_e$) that are also present in the document ($t \in \mathbf{W}_d$). The score of a document is:

$$s_{d,e} = \frac{|\mathbf{T}_{e,d}|}{|\mathbf{T}_e|} \in [0;1]$$

2.4.2. OKC: Occurrence-based keyword counting

With this method, the score of a document is the weighted sum of the number of event terms that appear at least once in the document. Repetitions of a word do increase the score of a document, but this effect is mitigated by the use of a logarithmic factor. Let $C_{t,d}$ be the count of occurrences of the event term t in the document d. The weight of a term is 1 for the main terms, and their normalized magnitude for the secondary terms. The score of a document is:

$$s_{d,e} = \sum_{t \in \mathbf{T}_{e,d}} w_t \cdot (1 + \log(C_{t,d}))$$

2.4.3. BM25: Okapi BM25

BM25 is a ranking function developed in the 1970s used to estimate the relevance of documents to a given search query. Despite its old age, variants of BM25 are still considered to be state-of-the-art. For our experiment, we chose to use the ATIRE BM25 variant, which is often considered a baseline implementation. This variant is biased towards shorter documents, a drawback which becomes a desired property when we want concise event descriptions.

2.4.4. TE: Text Embedding based ranking

Using the spaCy Python Natural Language Processing framework, we produced an average vector embedding for each event based on documents that could describe it. With this average vector, we can rank each document by its distance to this average embedding (cosine similarity).

Let $\mathbf{V}_d$ the vector embedding of document d. To construct document embeddings, spaCy computes the average of all the embeddings of the words of the document. Let $\overline{\mathbf{V}}_e$ the mean

Table 2
Notations used

Symbol	Description
$\mathbf{T}_e$	All terms of an event e
w_t	The weight of a term ($0 \leq w_t \leq 1$)
$\mathcal{D}_e$	Documents of the dataset containing the term e.
$s_{d,e}$	Score of a document d w.r.t the event e
$\mathbf{W}_d$	The list of words in a document

Table 3
Results showing various metrics of our 4 event labeling methods.

Method	1st correct	1st best	All 3 correct	All 3 best
1. SKC	95 %	88 %	84 %	72 %
2. OKC	97 %	72 %	94 %	69 %
3. BM25	89 %	81 %	69 %	66 %
4. TE	22 %	9 %	16 %	3 %

vector embedding of the set of all documents $d \in \mathcal{D}_e$ that include at least **one** of the terms of the event e, *i.e.* documents that could describe the event. The score formula is:

$$s_{d,e} = \cos\left(\overline{\mathbf{V}}_e, \mathbf{V}_d\right) = \frac{\overline{\mathbf{V}}_e \cdot \mathbf{V}_d}{\|\overline{\mathbf{V}}_e\| \, \|\mathbf{V}_d\|}$$

3. Event labeling evaluation

3.1. Experimental setup

The goal of our experiment is to evaluate the accuracy of each of the 4 labeling methods we described in the Section 2.4. Participants ($n = 2$) were provided with the interface depicted in Figure 2. They were given a list of 32 events, with 3 proposed descriptions for each. They were asked, for each full-text description, to indicate whether the proposed text is a correct description of the event, in their opinion, and rank the description according to its perceived correctness.

3.2. Experimental Results

Based on the human annotators' answers, we arrive at the results shown in Table 3. The first column is the name of the event description ranking method being evaluated. The following four columns present the percentage of correct proposals for which: the first ranked text is valid, the first ranked text is the best among the 3 presented texts, the top three proposed texts are valid, and finally the percentage of events for which all 3 proposed text are valid and their order is correct (the first description being the best, and the third being the least representative).

The most relevant metrics for our use case are the first two columns (1st correct, 1st best). Indeed, they indicate the quality of the first description produced by each method, which would be the only description shown to end users. As we can see, the best results are given by the *SKC* **Simple Keyword Counting** method, which is also the fastest to run. The Term Embedding method might need thorougher work to be on par with the other methods.

4. Discussion and Conclusion

In this short paper, we presented an ongoing work on event detection optimization and event labeling [3]. We optimized an existing method to be usable in an interactive event visualization context. We have also tested several event label search methods and were able to select a simple yet efficient one, that selects an event description taken from the input texts in order to complete the keywords provided by the baseline method.

We are currently working on the automatic estimation of the number of events present in the dataset, as well as on the dynamic estimation of the best value for the time-slice length. The creation of a larger event detection dataset and a larger scale user evaluation of user proposed labels are also in our future plans.

References

[1] Y. Chen, Z. Ding, Q. Zheng, Y. Qin, R. Huang, N. Shah, A history and theory of textual event detection and recognition, IEEE Access 8 (2020) 201371–201392. doi:10.1109/ACCESS.2020.3034907.

[2] K. Morabia, N. L. Bhanu Murthy, A. Malapati, S. Samant, SEDTWik: Segmentation-based event detection from tweets using Wikipedia, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop, Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 77–85. URL: https://aclanthology.org/N19-3011. doi:10.18653/v1/N19-3011.

[3] G. Jacobs, V. Hoste, Extracting fine-grained economic events from business news, in: Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation, COLING, Barcelona, Spain (Online), 2020, pp. 235–245. URL: https://aclanthology.org/2020.fnp-1.36.

[4] G. A. Al-Sultany, H. J. Aleqabie, Events Tagging in Twitter Using Twitter Latent Dirichlet Allocation, International Journal of Engineering & Technology 7 (2018) 884–888.

[5] D. Kilroy, S. Caton, G. Healy, Finding short lived events on social media, in: L. Longo, L. Rizzo, E. Hunter, A. Pakrashi (Eds.), Proceedings of The 28th Irish Conference on Artificial Intelligence and Cognitive Science, Dublin, Republic of Ireland, December 7-8, 2020, volume 2771 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 49–60. URL: http://ceur-ws.org/Vol-2771/AICS2020_paper_19.pdf.

[6] D. Gao, W. Li, X. Cai, R. Zhang, Y. Ouyang, Sequential Summarization: A Full View of Twitter Trending Topics, IEEE/ACM Transactions on Audio, Speech, and Language Processing 22 (2014) 293–302. URL: http://dl.acm.org/citation.cfm?id=2584479.2584480. doi:10.1109/TASL.2013.2282191.

[7] A. Guille, C. Favre, Event Detection, Tracking and Visualization in Twitter: A Mention-anomaly-based Approach, Springer Social Network Analysis and Mining 5 (2015) 18:1–18:18. doi:10.1007/s13278-015-0258-0.

[3] The source code of our work is available online at https://github.com/CorentinForler/mabed-interactive-labeling