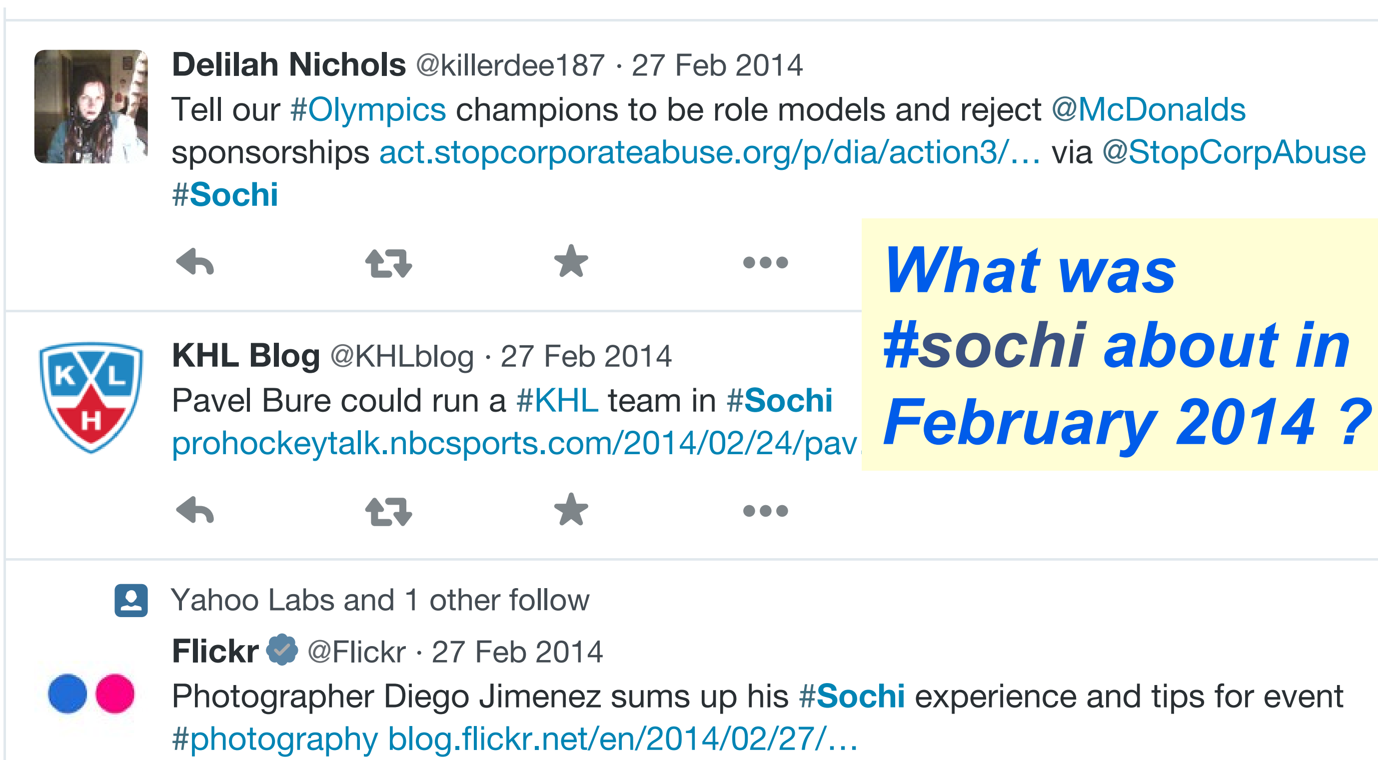
Semantic Annotation of Microblog Topics Using Wikipedia Temporal Information

Tuan Tran

2nd International Alexandria Workshop, Hannover, Germany

Research Goal

Understanding meanings of past trending hashtags



What was #sochi about in February 2014 ?

Research Goal

Understanding meanings of past trending hashtags

- Annotation is a crucial step !
- Goal: Mapping hashtag to Wikipedia pages

What was #sochi about in February 2014 ?

Research Goal

Semantic annotation for hash tag is not easy:

- Hashtags are peculiar

#sochi	vs.	**2014_Winter_Olympics**
#jan25	vs.	**Egypt_Revolution_of_2011**
#mh370	vs.	**Malaysia_Airlines_Flight_370**
#crimea	vs.	**Ukrainian_crisis**

- Time matters

#germany :

Greek_government-debt_crisis

2014_FiFA_World_Cup

European_migrant_crisis

July 2014

July 2015

Sep 2015

Semantic Annotations in Twitter

Mostly in tweet level:

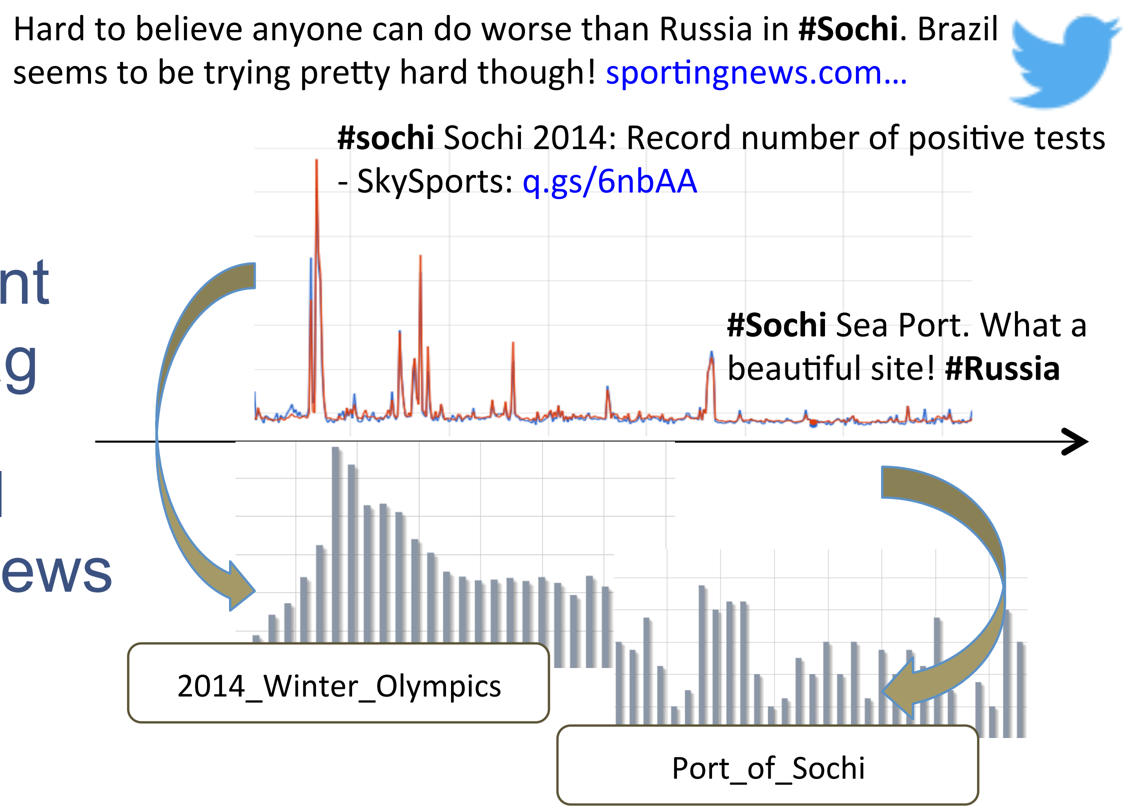
- Tweak the similarities (*TAGME*, CIKM'10; Liu, ACL'13)
- Employ Twitter-specific features with human supervision (Meij, WSDM'12; Guo, NAACL'13)
- Expand context to users (Cassidy, COLING'12; *KAURI*; KDD'13), time (Fang, TACL'14; Hua, SIGMOD'15)

Annotating hashtags is limited to general topic models (Ma, CIKM'14; Bansal, ECIR'15)

Main Idea

Align contexts from *both* sides:

- Twitter: All constituent tweets of the hashtag
- Wikipedia: Temporal signals from page views and edit history



Framework Outline

Problem: Given a *trending* hashtag, its burst time period T, identify top-k most prominent entities to describe the hashtag in T.

Three steps:

1. Candidate Entities Identification
2. Entity – Hashtag Similarities
3. Entity Prominence Ranking

Candidate Entities Identification

Mine from tweets contents via lexical matching.

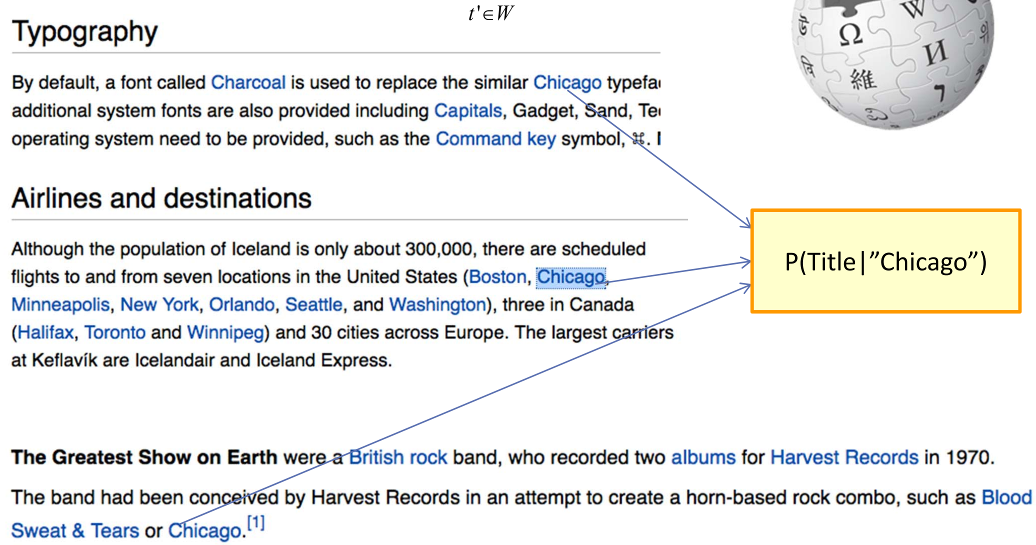
- Twitter side: Extract *n*-grams from tweets ($n \leq 5$)
- Wikipedia side: Build a lexicon for each entity: Anchors of incoming links, Redirects, Titles, Disambiguation pages
- Start with sample text, expand via links to increase recall

Entity – Hashtag Similarities: Link-based

- Build upon phase – entity similarities
- Use *Commonness*
- Aggregate linearly to hashtag – entity similarity

P(t|m): "Commonness"

$$Commonness(m \Rightarrow t) = \frac{count(m \rightarrow t)}{\sum_{t' \in W} count(m \rightarrow t')}$$



(Slide from Dan Roth: *"Wikification and Beyond: The Challenges of Entity and Concept Grounding".* Tutorial at ACL 2014)

Entity – Hashtag Similarities: Text-based

- Compare the distributions of words over hashtags and texts of the entity
- Consider both entity's static text and *temporal edits*:

$$\hat{P}(w|e) = \lambda \hat{P}(w|M_{C_T(e)}) + (1-\lambda)\hat{P}(w|M_{C(e)})$$

Language model of e's edited text during T

Language model of e's latest text

Entity – Hashtag Similarities: Collective Attention-based

Compare the temporal correlation of collective attention between the hashtag and the entity:

$$f_t(e,h) = \min_{q,\delta} \frac{\|TS_h - \delta d_q(TS_e)\|}{\|TS_h\|}$$

time series shifted from TS_e by q units

[1] *Jaewon Yang, Jure Leskovec. "Patterns of Temporal Variation in Online Media". WSDM 2011*

Entity Prominence Ranking

- Rank by the unified similarity score:

$$f(e,h) = \alpha f_m(e,h) + \beta f_c(e,h) + \gamma f_t(e,h)$$
$$\alpha + \beta + \gamma = 1$$

- To learn the model $\omega = (\alpha, \beta, \gamma)$, we need a *premise* of what makes an entity prominent !
- The coherence premise[2] is not applicable at topic level

[2] *Ratinov et al. "Local and Global Algorithms for Disambiguation to Wikipedia". ACL 2012*

Influence Maximization

Observation: Prominent pages are first created / updated with texts, then linked to other pages

- Reflect the shifting attentions of users in social events[3]

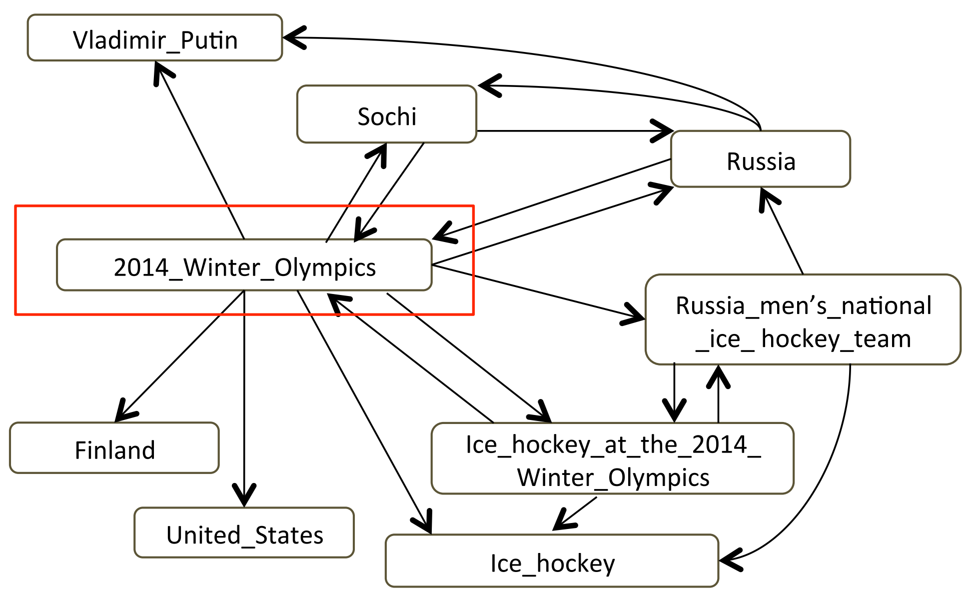
***#love #Sochi** 2014: Russia's ice hockey dream ends as Vladimir Putin watches on …*

***#sochi** Sochi: Team USA takes 3 more medals, tops leaderboard | http://abc7.com http://adf.ly/dp8Hn*

#Sochi** bear after **#Russia**'s hockey team eliminated with loss to **#Finland

*I'm still happy because Finland won. Is that too stupid..? **#Hockey #Sochi***

…



[3] *Keegan et al. "Hot off the wiki: Dynamics, Practices, and structures in Wikipedia.." WikiSym 2011*

Influence Maximization

Premise: Rank top-k entities so as to maximize the spreading to all other candidates.

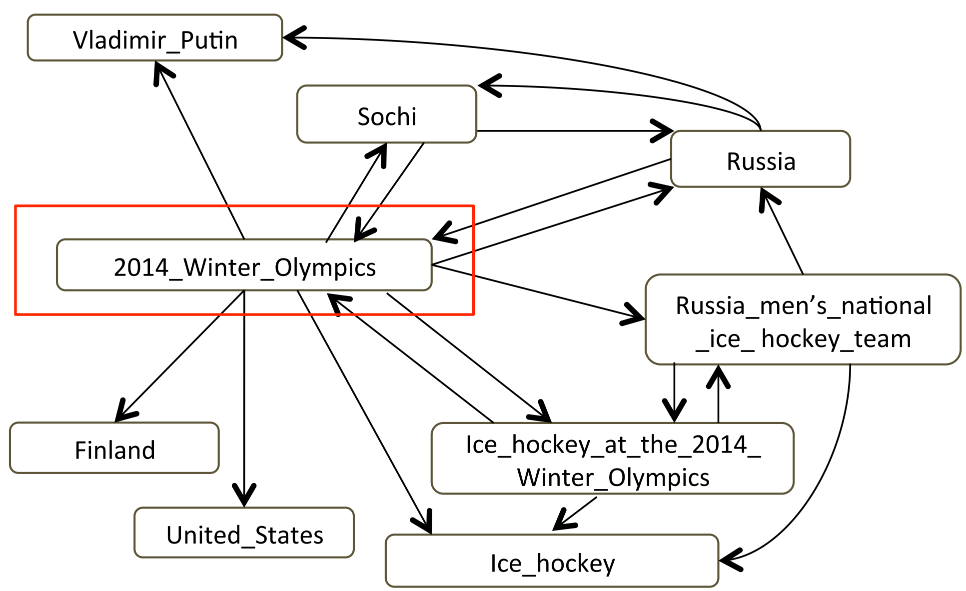
#love #Sochi *2014: Russia's ice hockey dream ends as Vladimir Putin watches on …*

#sochi *Sochi: Team USA takes 3 more medals, tops leaderboard | http://abc7.com http://adf.ly/dp8Hn*

#Sochi *bear after* ***#Russia****'s hockey team eliminated with loss to* ***#Finland***

I'm still happy because Finland won. Is that too stupid..? ***#Hockey #Sochi***

…



Experiments

Data:

- Collect 500 million tweets for 4 months (Jan-Apr 2014) via Streaming API.
- Process and sample distinct trending hashtags
  - Several heuristics + clustering methods[4] used → 2444 trendings
  - 3 inspectors chose 30 meaningful trending hashtags

[4] *Lehmann et al. "Dynamical Classes of Collective Attention in Twitter". WWW 2012*

Experiments

Baselines:

- Wikiminer (Milne & Witten, CIKM 2008)
- Tagme (Ferragina et al., IEEE Software 2012)
- KAURI (Shen et al., KDD 2013)
- Meij method (Meij et al., WSDM 2012)
- Individual similarities : M (link), C (text), T (temporal)

Evaluation: 6,965 entity-hashtag pairs are evaluated from 0-1-2 scales (5 evaluators, inter-agreement 0.6)

Experiments

	Tagme	Wikiminer	Meij	Kauri	M	C	T	IPL
P@5	0.284	0.253	0.500	0.305	0.453	0.263	0.474	**0.642**
P@15	0.253	0.147	**0.670**	0.319	0.312	0.245	0.378	0.495
MAP	0.148	0.096	0.375	0.162	0.211	0.140	0.291	**0.439**

Better in general

Non-verbal signal is good

Experiments

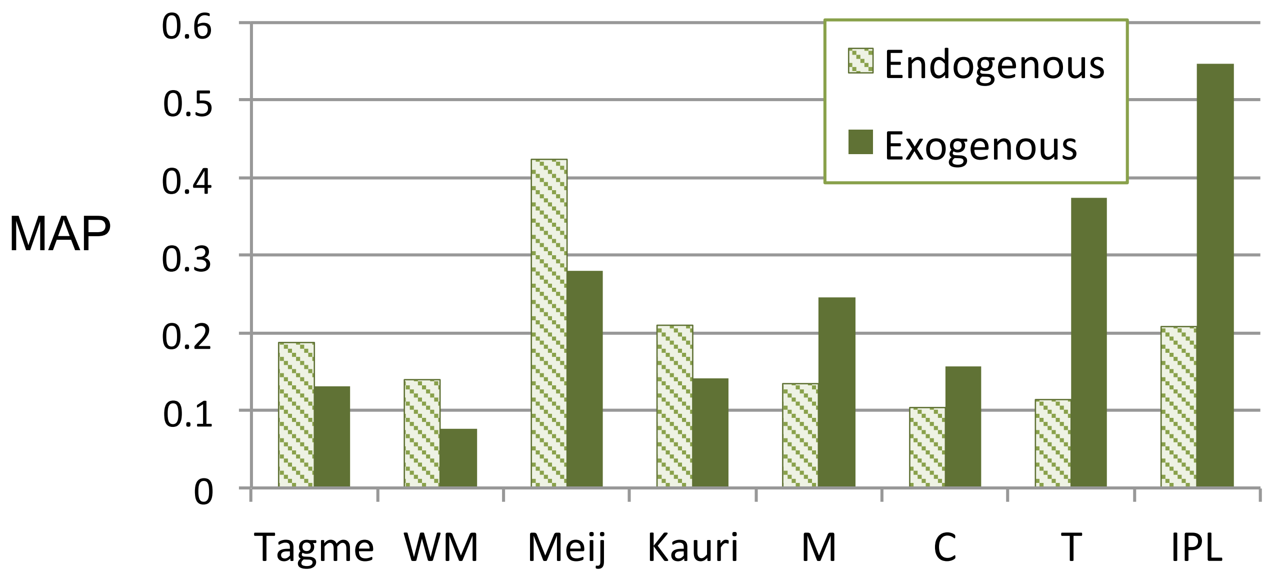
Better at top

	Tagme	Wikiminer	Meij	Kauri	M	C	T	IPL
P@5	0.284	0.253	0.500	0.305	0.453	0.263	0.474	**0.642**
P@15	0.253	0.147	**0.670**	0.319	0.312	0.245	0.378	0.495
MAP	0.148	0.096	0.375	0.162	0.211	0.140	0.291	**0.439**

Better when
including
low-ranked entities.

Experiments

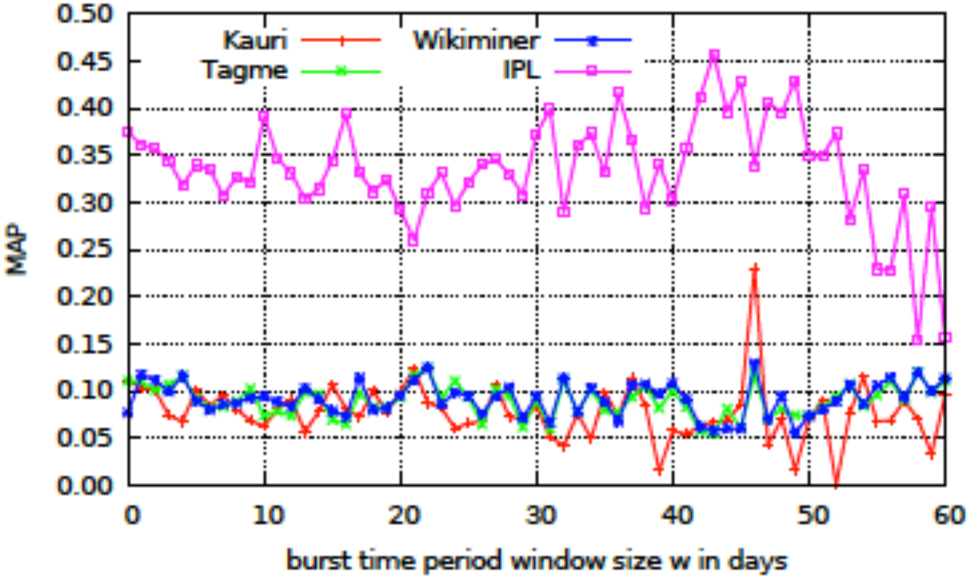
Manually class events (hash tags' peaks) to endogenous / exogenous via checking tweets' contents



Experiments

Influence of size of burst time period:

- Larger window → more noisy introduced



Conclusions

- Semantic Annotation in topic level is difficult
- Lesson learnt: Can be improved by exploiting temporal and contexts from both sides (non verbal evidences are promising)
- Future direction: Improve efficiency, text-based similarities

Thank you ☺

Question ?

Terminology

Trending hashtag: There are peaks[1] from daily tweet count time series:

- Variance score > 900
- Highest peak > 15 x median of 2-month window sample

Burst time periods: w-window around one peak

Entities: Wikipedia pages, no redirects, disambiguation, lists

- Entity text, view count per day, edits during T

[1] *Lehman et al. "Dynamical classes of collective attention in Twitter". WWW 2012*

Candidate Entities Identification

Mine from tweets contents, via lexical matching.

- Twitter side: Extract *n*-grams from tweets ($n \leq 5$)
  - Parse POS tags for tokens, filter patterns using rules
- Wikipedia side: Build a lexicon (anchors, redirects, titles, disambiguation pages)
- Practical issue:
  - Start from sample tweets
  - Expand to incoming / outgoing linked entities

Influence Maximization-based Learning

Measure the observed spreading activities via entities *influence scores*

- Learn ω to minimize the loss w.r.t. influence score r:

$$\omega = \arg\min \sum_{E(h,k)} L(f(e,h), r(e,h))$$

- Influence score is estimated via random walks:

$$\mathbf{r_h} := \tau \mathbf{B} \mathbf{r_h} + (1-\tau)\mathbf{s_h}$$

- $r(e,h)$ and $f(e,h)$ is jointly learnt via gradient descent method

Entity – Hashtag Similarities: Link-based

Built upon direct similarities of tweets – entities:

- Based on commonness (Meij, WSDM12; Fang, TACL14)

$$LP(e|m) = \frac{|l_m(e)|}{\sum_{m'} |l_{m'}(e)|}$$

No. of incoming links to e with anchor m

- Aggregate to hashtag level, weighted by the frequency:

$$f_m(e,h) = \sum_m (LP(e|m) \cdot q(m))$$

No. of times m appears in h's tweets

Influence Graph

- A link from a to b indicates an "influence endorsement" from b to a
- Level of endorsement is proportional to the relation weight:

$$MW(e_1, e_2) = 1 - \frac{\log(\max(|I_1|,|I_2|) - \log(|I_1 \cap I_2|)))}{\log(|E|) - \log(\min(|I_1|,|I_2|))}$$

- Normalize to have influence matrix:

$$b_{i,j} = \frac{MW(e_i, e_j)}{\sum_{(e_i,e_k)\in V} MW(e_i, e_k)}$$

Iterative Influence-Propagation Learning

Input : $h, T, D_T(h), \mathbf{B}, k$, learning rate μ, threshold ϵ
Output: ω, top-k most prominent entities.

Initialize: $\omega := \omega^{(0)}$
Calculate $\mathbf{f}_m, \mathbf{f}_c, \mathbf{f}_t, \mathbf{f}_\omega := \mathbf{f}_{\omega^{(0)}}$
while *true* **do**
    $\hat{\mathbf{f}}_\omega :=$ normalize $\mathbf{f}_\omega$
    Set $\mathbf{s_h} := \hat{\mathbf{f}}_\omega$, calculate $\mathbf{r_h}$
    Sort $\mathbf{r_h}$, get the top-k entities $E(h, k)$
    if $\sum_{e \in E(h,k)} L(f(e,h), r(e,h)) < \epsilon$ **then**
        Stop
    end
    $\omega := \omega - \mu \sum_{e \in E(h,k)} \nabla L(f(e,h), r(e,h))$
end
return ω, $E(h, k)$

Hashtag Sampling

- Calculate for each peak, the vector (f_a,f_b,f_c) of portion of tweets before, during, and after the peak time.
- Clustering with EM, choose 4 most plausible clusters.
- Sample separately from each cluster