Discovering and Learning Sensational Episodes of News Events

Xiang Ao[1,3], Ping Luo[1], Chengkai Li[2], Fuzhen Zhuang[1], Qing He[1], Zhongzhi Shi[1]
[1]Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS),
Institute of Computing Technology, CAS, Beijing, China
[2]University of Texas at Arlington, USA
[3]University of Chinese Academy of Sciences, Beijing, China
{aox,zhuangfz,heq,shizz}@ics.ict.ac.cn, ping.luoul@gmail.com, cli@uta.edu

ABSTRACT

This paper studies the problem of discovering and learning sensational 2-episodes, i.e., pairs of co-occurring news events. To find all frequent episodes, we propose an efficient algorithm, MEELO, which significantly outperforms conventional methods. Given many frequent episodes, we rank them by their sensational effect. Instead of limiting ourselves to any individual subjective measure of sensational effect, we propose a learning-to-rank approach that exploits multiple features to capture the sensational effect of an episode from various aspects. An experimental study on real data verified our approach's efficiency and effectiveness.

Categories and Subject Descriptors

H.2.8 [**Information Systems**]: DATABASE MANAGEMENT—*Database applications*

Keywords

Frequent episode mining; Learning to rank; News event

1. INTRODUCTION

Since May 2011 to February 2012, a *Ramsey's Curse* had gained much sensation among soccer reporters and fans in the world. When the Welsh soccer player Aaron Ramsey scored in a match, an international figure in politics or culture may die within several days. Osama bin Laden, Muammar Gaddafi, Steve Jobs and Whitney Houston died within 21−72 hours after Ramsey scored in games. It created the superstition that Ramsey's goals triggered the deaths of famous people. Such pairs of co-occurring news events within short period, while not necessarily bearing causal relationship between each other, create tabloid fodder that media likes to capitalize on. We call such a pair of events a *sensational episode*. A sensational episode refers to 2-episode in frequent episode mining [4] (FEM for short)–a serial episode consisting of 2 events, which is in the form of $lhs \rightarrow rhs$.

It is accomplished by two steps to discover sensational episodes of news events. The first task is to find all *frequent 2-episodes*. For this, we designed an efficient algorithm—MEELO. The frequency of an episode is defined by its minimal occurrences [4]. Most conventional algorithms for frequent episode mining need to check, by post-processing, whether a detected occurrence is minimal or not. MEELO fundamentally differs from previous algorithms in that it does not need post-processing in finding minimal occurrences. There can be a large number of frequent 2-episodes of news events that satisfy a given frequency threshold. The second task is to rank all such candidate episodes by their *sensational effect* from the perspectives of news audience. Instead of limiting ourselves to any individual subjective measure of sensational effect, we propose a learning-to-rank approach that exploits multiple features to capture the sensational effect of an episode from different aspects. Given a set of manually labeled frequent 2-episodes and their features, we trained a random forests ranker for the second task.

WWW'14 Companion, April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04.
http://dx.doi.org/10.1145/2567948.2577290.

2. DISCOVERING AND LEARNING SENSATIONAL EPISODES

Mining frequent 2-episodes: We define an *event* e as a tuple $\langle s, v, o\rangle$, where s denotes a *subject entity*, v denotes a *verb predicate* and o denotes an *object entity*. Either s or o (but not both) can be absent (denoted as NULL). The set of all distinct events, denoted $\mathcal{E}$, is called the *universe event set*. Let $\vec{S} = (E_1, T_1), \ldots, (E_n, T_n)$ be an *event occurrence sequence*, which is an ordered sequence of event sets, where each $E_i \subseteq \mathcal{E}$ consists of all events at time T_i. The event sets are chronologically ordered by their timestamps, i.e., $T_i < T_j$ for all $1 \leq i < j \leq n$. A *2-episode* α consists of two events $e_1 \rightarrow e_2$ where $e_1 \neq e_2$, e_1 is the *antecedent event* and e_2 is the *consequent event*. Given a 2-episode $\alpha = e_1 \rightarrow e_2$, the time interval $[T_1, T_2]$ is an *occurrence* of α if (1) e_1 occurs at time T_1; (2) e_2 occurs at time T_2; and (3) $0 < T_2 - T_1 < max\_intvl$, where max_intvl is a user-specified threshold called the *maximum time interval*. An occurrence of 2-episode α, $[T_1, T_2]$, is a *minimal occurrence* of α if no other occurrence $[T_1', T_2']$ of α is a subinterval of $[T_1, T_2]$. Given a 2-episode α, let the *support* of α be the number of its *minimal occurrences*. A 2-episode is *frequent* if and only if its support is no less than min_sup—a user-specified *minimum support threshold*.

Based on the definitions, we propose an algorithm MEELO (Mining frEquent Episode via Last Occurrence) to find all frequent 2-episodes in an event occurrence sequence $\vec{S}$, given two thresholds min_sup and max_intvl. MEELO finds all frequent 2-episodes by two passes of sequential scan over $\vec{S}$. In the first pass, it removes from $\vec{S}$ all occurrences of infrequent events (whose frequencies are less than min_sup). In the second pass, MEELO finds frequent 2-episodes by another sequential scan over the resulting subsequence from the first pass. It relies on a data structure, *last occurrence list* (LO-list for short, denoted $\mathcal{L}$), to find all minimal occurrences of 2-episodes. Each node $q = q.event{:}q.time$ in $\mathcal{L}$ consists of two fields—$q.event$ and $q.time$, which record an event and an occurrence time of the event, respectively. MEELO scans frequent event occurrence sequence $\vec{S}$ from beginning (T_1) to end (T_n). At

every timestamp T_i, the following invariants on the content of LO-list $\mathcal{L}$ are guaranteed by the algorithm:

Invariant 1 For every node $q = q.event{:}q.time$ in $\mathcal{L}$, $q.time$ is the time of the latest occurrence of $q.event$ during time window $[T_p, T_i]$, where

$$T_p = \max(T_1, T_i - max\_intvl + 1) \quad (1)$$

Invariant 2 The nodes in $\mathcal{L}$ from head to tail are in reverse chronological order of their timestamps.

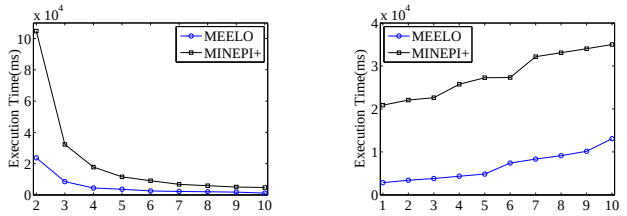
Learning sensational relationship: We attempt to learn sensational effect through labeled examples of frequent episodes which are represented by several developed features.Utilizing entity taxonomy DBPedia Ontology, we define a *super event* e_s as an event, which includes at least one entity in an abstraction form. With super events we can build a taxonomy of events and aim to identify the sensational episodes with both the basic and super events. Further, we define the depth of an episode $\alpha = e{\rightarrow}e'$, denoted by $\mathrm{depth}(\alpha)$, as the arithmetic mean of the *depth* of the event e and e', and the *depth* of events can be calculated based on the entity taxonomy. Based on the taxonomy of events, we propose two features to measure sensational effect of frequent 2-episodes. One is inspired by [3], denoted by $\mathrm{HSC}(\alpha)$, which measures the semantic correlation of two events. The other is based on the lift of two events, denoted by $\mathrm{HLift}(\alpha)$, which measures the correlation in probability of two events. $\mathrm{depth}(\alpha)$ is used as a multiplier which intents to balance the bias brought by the entity taxonomy.

Second, we adopt the *bombshell value* for each occurrence of an event. The bombshell value of an event is actually its sensational effect. Since all the events are extracted from the tweets in this study, we use the re-tweet number of the tweet, which includes the event, as the bombshell value for this event occurrence. For a super event, to acquire its bombshell, we can sum the bombshell values of all its successors occurring at the same time. Based on the event bombshell, we define the difference between bombshell values of the occurrences of the consequent event and those of the antecedent event as a feature related to the sensational effect. Furthermore, for an episode $\alpha = e{\rightarrow}e'$, we adopt only the bombshell values of e' to obtain a bombshell of α, and take the Kulczynsky correlation [2] as a weight, and we denote this feature as $\mathrm{WBV}(\alpha)$. Some traditional measures are also utilized here to measure the sensational effect, including the Granger value and the support of α. With all these aforementioned features, we can generate a feature vector $\vec{F}(\alpha)$. By training a random forests ranker implemented in RankLib on labeled examples with $\vec{F}$, we can give a predicted rankings for frequent episodes by their sensational effect.

3. EXPERIMENTS

We gathered $65,298$ tweets of twelve news agency accounts in Twitter range from February 21, 2012 to March 17, 2013. After several preprocessing, we finally obtain the universe event set $\mathcal{E}$ including $23,692$ basic events and $47,358$ super events.

All experiments, implemented in Java, are performed on a computer with a 3.30 GHz Intel Core i5 Processor with 8G memory. For efficiency comparison, we compare the impact of two parameters (min_sup and max_intvl) on the efficiency of MEELO in mining frequent 2-episodes. The baseline algorithm we compare with is MINEPI+ [1] which is a state-of-art in FEM on *complex sequence* containing multiple events in the same timestamp (same with the event occurrence sequence). For effectiveness investigation, we focus on the impact of different proposed features. $2,056$ frequent episodes are labeled by news audience with popular five-graded absolute relevance judgement. We divide the labeled examples into three parts and perform 3-cross validation on this labeled data set. For this comparison, the feature $\mathrm{HSC}(\alpha)$, $\mathrm{Hlift}(\alpha)$, and



(a) Minimum support (b) Maximum time interval

Figure 1: The execution time under different settings

Table 1: Feature set effectiveness over labeled data set

Feature Set	NDCG@5	NDCG@10	ERR@10
BF	$0.5499_{(\pm 0.0706)}$	$0.4777_{(\pm 0.053)}$	$0.9469_{(\pm 0.0068)}$
BF+HSC	$0.5687_{(\pm 0.0788)}$	$0.5423_{(\pm 0.0513)}$	$0.9506_{(\pm 0.0037)}$
BF+HSC +HLift	$0.5861_{(\pm \mathbf{0.0167})}$	$0.5326_{(\pm \mathbf{0.0182})}$	$0.952_{(\pm \mathbf{0.0012})}$
BF+HSC+ HLift+WBV	$\mathbf{0.6477}_{(\pm 0.1043)}$	$\mathbf{0.58}_{(\pm 0.0736)}$	$\mathbf{0.9544}_{(\pm 0.0031)}$

$\mathrm{WBV}(\alpha)$ will be added gradually to the feature set, and they are denoted as HSC, HLift and WBV, respectively. The other features are considered as baseline feature set, which is denoted as BF. The parameters of random forests ranker are set as default values.

Figure 1(a) shows the execution time on the data set under varied minimum support when the maximum time interval is set to seven. Figure 1(b) shows the execution time on the data set under varied maximum time interval when the minimum support is set to three. From the figures, we can see MEELO always has a lower execution time compared with MINEPI+. Since MEELO avoids the post-processing on checking minimal occurrences of generated episodes, it dominates a better performance in efficiency. Table 1 shows the mean with the standard deviation of each metric on different feature sets, and the highest value of each metric is marked in boldface. From the table, we can see HSC is useful to detect sensational episodes since the improvements are significant on NDCG@10 and ERR@10, HLift is useful to acquire a stable performance because the standard deviations between different folds are significantly reduced compared with the feature sets without it, and WBV is an essential factor for giving sensational episodes higher rankings since the improvements on NDCG@5 are significant. Last, we exhibit some examples of interesting sensational episodes discovered in the data set: 1) The United States stock, e.g. Dow Jones Industrial, would slip when Barack Obama lead his opponent Mitt Romney in poll or in the final election. We can find some implicit causality for it. Since different background between the two candidates, the financial community support Romney than Obama during the period of election 2012. 2) There would be politicians in the United States resign at most in four days after the movie "Argo" won an award in film festivals. The resigned politicians include Ken Salazar, Hillary Clinton and Steven Chu.

Acknowledgements This work is supported by the National Natural Science Foundation of China (61175052, 61203297, 61035003), National High-tech R&D Program of China (863 Program) (No. 2014AA012205, 2013AA01A606, 2012AA011003). Li is partially supported by NSF grants 1018865, 1117369, and 2011, 2012 HP Labs IRP Awards. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the funding agencies.

4. REFERENCES

[1] Kuo-Yu Huang et al. Efficient mining of frequent episodes from complex sequences. *Information Systems*, 33(1):96–114, 2008.

[2] Tianyi Wu et al. Re-examination of interestingness measures in pattern mining: a unified framework. *DMKD*, 21(3):371–397, 2010.

[3] Wei Shen et al. Linden: linking named entities with knowledge base via semantic knowledge. In *WWW12*.

[4] Heikki Mannila and Hannu Toivonen. Discovering generalized episodes using minimal occurrences. In *KDD96*.