A Framework to Represent and Mine Knowledge Evolution from Wikipedia Revisions

Xian Wu[†‡],Wei Fan[*], Meilun Sheng[†], Li Zhang[‡],
Xiaoxiao Shi[#], Zhong Su[‡] and Yong Yu[†]
[†]Shanghai Jiao Tong University [‡]IBM Research - China
[*]IBM T.J.Watson Research Center [#]University of Illinois at Chicago
{wuxian,meilunsheng,yyu}@apex.sjtu.edu.cn, weifan@us.ibm.com,
{lizhang,suzhong}@cn.ibm.com,xiaoxiao@cs.uic.edu

ABSTRACT

State-of-the-art knowledge representation in semantic web employs a triple format (subject-relation-object). The limitation is that it can only represent static information, but cannot easily encode revisions of semantic web and knowledge evolution. In reality, knowledge does not stay still but evolves over time. In this paper, we first introduce the concept of "quintuple representation" by adding two new fields, *state* and *time*, where *state* has two values, either *in* or *out*, to denote that the referred knowledge takes effective or becomes expired at the given *time*. We then discuss a two-step statistical framework to mine knowledge evolution into the proposed quintuple representation. Utilizing extracted quintuple properly, it not only can reveal knowledge changing history but also detect expired information. We evaluate the proposed framework on Wikipedia revisions, as well as, common web pages currently not in semantic web format.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

Knowledge Evolution Extraction, Wikipedia Revision, Expired Data Detection

1. INTRODUCTION

Knowledge does not stay still but changes over time, typical knowledge representations like semantic triples only encodes the static knowledge, but can not easily describe how knowledge evolves. To solve this problem, we focus on mining the process of knowledge evolution rather than at a particular state. To define an update in knowledge, we extend the typical triple representation with two more fields and introduce a representation as below:

Definition 1: *Knowledge Update Quintuple: (subject-relation-object-state-time) where the "subject-relation-object" triple is used to represent a piece of knowledge; "state" is set either "in" or "out" to denote that this referred knowledge takes effect or becomes expired; "time" records when the knowledge update occurs.*

WWW 2012 Companion, April 16–20, 2012, Lyon, France.
ACM 978-1-4503-1230-1/12/04.

A naive approach to acquire quintuples would be: First apply state-of-art triple extraction methods [1] to each version of the article and then compare the sets of triples between every two adjacent versions. For each triple variation in comparison, a quintuple is generated. However, in practise, five types of revisions could cause triple variations including writing polishing, structure reorganization, vandalism, editing war and knowledge update. Among these five types of revisions, vandalism and edit wars provide false or subjective knowledge updates, in the same time, writing polish and structure re-organization only alter the representation without contributing informative updates.

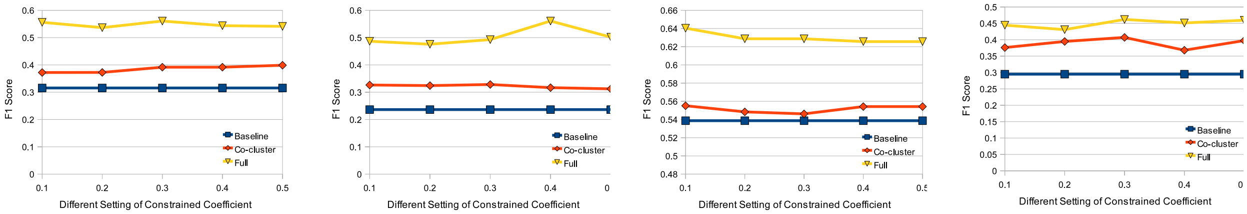
Table 1: Six Quintuples on the Subject "Juventus" and Object "Ciro Ferrara"

Subject	Relation	Object	State	Time
Juventus	Greatest_player	Ferrara	out	07-14
Juventus	Team_members/ Noted_former_player	Ferrara	in	07-14
Juventus	Team_members/ Noted_former_player	Ferrara	out	08-22
Juventus	Noted_former_players	Ferrara	in	08-22
Juventus	Noted_former_players	Ferrara	out	11-14
Juventus	Notable_former_players	Ferrara	in	11-14

For example, Table 1 lists six quintuples detected between the subject "Juventus" and the object "Ciro Ferrara" by the naive direct comparison method. However, none of them are actual knowledge updates. This is because the four relations "Greatest_players", "Team_members/Noted former players", "Noted former players" and "Notable former players" are actually identical in semantics. In this manner, the six quintuples can cancel each other out. In this paper, we employ a two-stage statistical framework to identify the quintuples, as a result of knowledge updates and remove those caused by other four types of revisions.

2. THE PROPOSED FRAMEWORK

Given a web page w, let $\{(r_1, t_1), (r_2, t_2), ..., (r_N, t_N)\}$ denote its N historical revisions and (r_i, t_i) denote the revision r_i conducted at time t_i; Let $V = \{v_0, v_1, ..., v_N\}$ denote the $N + 1$ historical versions where v_i is modified from v_{i-1} by the revision r_i. After performing triple extraction on each version, a naive approach to acquire quintuples is formulated in (1). However, the complication is that a quintuple obtained in this way does not always equal to an update in



(Juventus) (Apple) (Movie) (Home Page)

Figure 1: F1 Score of Three Models on Four Data Sets with Different Constrained Co-clustering coefficient δ

knowledge. We introduce the following algorithm.

$$Q_i = \{T_{i-1} - T_i, out, t_i\} \cup \{T_i - T_{i-1}, in, t_i\} \qquad (1)$$

where T_{i-1} and T_i denote the sets of triples extracted from the version v_{i-1} and v_i respectively. Thus $T_i - T_{i-1}$ denotes the triples that are added by the version r_i and $T_{i-1} - T_i$ denotes the triples that are detected from the version r_{i-1}.

Algorithm 1: Detecting Knowledge Updates from Revision Data

```
Input  : $V$: all the historical versions of a web page $w$.
Output: $U$: the set of detected knowledge updates.
Initialize the set of quintuples $Q = \emptyset$.
foreach Version $v_i$ in the set $V$ do
    Obtain $Q_i$ according to Eq. (1).
    AddToSet($Q_i, Q$).
Grouping the semantic identical quintuples via
  $G$ = Constrained-Cocluster($Q$).
Perform Kalman Filter to remove the noise and obtain
  the knowledge updates, $U$ = KalmanFilter(G).
```

Algorithm 1 contains four steps: (1) Extract triples from each historical version of an article (2) Compare sets of triples between adjacent versions and generate a raw set of quintuples. (3) Unify the semantics of subjects, relations and objects fields of quintuples with a constrained spectral co-clustering model [2], and then merge the quintuples caused by inconsistent knowledge representations. As a result, the inconsistent knowledge representation before and after writing polish and structure re-organization is removed. (4) Considering the lasting time as a filtering criterion, a Kalman filter is introduced to cleanse the false updates caused by vandalism and edit wars.

3. EMPIRICAL EVALUATION

We evaluate the proposed framework on revision history of three Wikipedia article: Juventus F.C, Apple Inc., List of Highest-grossing Films and a data set of home pages of PhD Students. After obtaining the above four data sets, we apply the methods to extract triples from both structured and unstructured content of each revision. Then we generate quintuples by comparing adjacent triples sets. We invite annotators to label the actual knowledge updates from them. They are required to label each quintuple whether it is a knowledge update or not.

Baseline Model: Instead of directly performing (1) on all the revisions, we select the ones that last for more than one hour.

Co-cluster Model: Compared to the complete approach in Algorithm 1, this model only consists of the constrained co-clustering step, the Kalman Filter is not included.

Full Model: The full model consists of both constrained co-clustering and the Kalman Filter.

Figure 1 summarizes the comparison results of above three models with different settings of the constrained co-clustering co-efficient δ, in which, δ is used to balance between the co-occurrence effects and the string similarity. A larger δ tends to group lexicon similar items, while a lower δ mainly considers the co-occurrence information. As shown in Figure 1, with different δ, the *Co-cluster Model* consistently outperforms the baseline model and the *Full Model* outperforms the other two. For the Wikipedia data sets, as there are many cases of vandalism and edit wars, the addition of Kalman Filter can remove these noise and improve the performance significantly. As to the home page data set, since a web page is maintained by a single person, most of non-informative revisions are caused by structure re-organizations and writing polishes, therefore Constrained co-clustering dominates the performance improvement.

4. CONCLUSIONS

In this paper, we first discuss the knowledge evolution mining problem and introduce quintuples to represent knowledge evolution. Compared to typical triple representation in Semantic Web, the new quintuple representation helps record the effective and expiration period of knowledge. To derive such quintuples, we then propose a two stage statistical framework to mine revision data. We evaluate the proposed approach using Wikipedia revision data, as well as the home pages of PhD students. From the experimental results, the proposed approach outperforms both the baseline straight-forward merging model and the co-clustering model. Due to the coverage and diversity of Wikipedia data, the acquired knowledge updates can be used as a repository to detect expired web information and derive the complete knowledge evolution road map. Therefore, the web user will not be misled, but can view and understand knowledge from an evolving perspective.

5. REFERENCES

[1] O. Etzioni, M. Banko, S. Soderland, and D. S. Weld. Open information extraction from the web. *Commun. ACM*, 51:68–74, December 2008.

[2] X. Shi, W. Fan, and P. S. Yu. Efficient semi-supervised spectral co-clustering with constraints. *IEEE International Conference on Data Mining*, 0:1043–1048, 2010.