IEICE TRANS. INF. & SYST., VOL.E96–D, NO.7 JULY 2013

LETTER

Characterizing Web APIs Combining Supervised Topic Model with Ontology*

Yuanbin HAN[†], ***Student Member***, **Shizhan CHEN**[†a)], ***and*** **Zhiyong FENG**[†], ***Nonmembers***

SUMMARY This paper presents a novel topic modeling (TM) approach for discovering meaningful topics for Web APIs, which is a potential dimensionality reduction way for efficient and effective classification, retrieval, organization, and management of numerous APIs. We exploit the possibility of conducting TM on multi-labeled APIs by combining a supervised TM (known as Labeled LDA) with ontology. Experiments conducting on real-world API data set show that the proposed method outperforms standard Labeled LDA with an average gain of 7.0% in measuring quality of the generated topics. In addition, we also evaluate the similarity matching between topics generated by our method and standard Labeled LDA, which demonstrates the significance of incorporating ontology.

key words: *Web API, supervised topic model, topic coherence, ontology, multi-labeled*

1. Introduction

Web API has become an increasingly novel data-oriented form of web service due to its flexibility and programmability. According to ProgrammableWeb [1], there have been over 8000 APIs and nearly 7000 Mashup applications registered by public providers until Jan 2013. With the fast growth of Web APIs online, the development of effective and efficient tasks like classification, retrieval, organization, and management is needed.

One possible solution to above issues can be topic modeling technique known as Latent Dirichlet Allocation (LDA) [2], which models large document corpus into a three-level structure: each document is represented by a multinomial distribution over latent topics, and each topic corresponds to a multinomial distribution over words. Due to the powerful statistical ability of automatically discovering low-dimensional sub-features behind large document corpus, LDA has been vastly applied in various objects [3]. Recently, Labeled LDA [4] was proposed to extend LDA into supervised model, where the number of topics is supervised by labels, and topics are no longer latent and associated with unique and interpretable names–labels.

This paper focuses on discovering meaningful topics in Web APIs by leveraging an ontology extended Labeled LDA. As Labeled LDA constrains a one-to-one correspondence between topics and tags, each extracted topic can be associated with a unique label. However, as tags are user-generated, they are naturally coarse in semantics. Several tags may represent the same semantic (e.g., "*blog*", "*blogging*" and "*microblogging*" imply a single semantic). In addition, when saying a topic, it generally represents a high-level abstracting semantic, however, many tags are too fine-grained to be referred as topics, e.g., "*mapping*", "*viewer*", "*GPS*" and "*display*", they are actually forming a topic, or part of a topic (e.g., "*map*").

Considering these shortcomings, Labeled LDA could generate poor quality or redundant topics. We propose to pre-aggregate noisy tags by leveraging the hierarchy of ontology, where synonymy tags can be annotated into unique concepts, and fine-grained tags can be mapped into suitable high-level abstracting concepts. In doing so, we semi-automatically build a small ontology based on the co-occurrences of tags, and project the constructed ontology into Labeled LDA. Experiments on real-world API data set show that the proposed method outperforms standard Labeled LDA, and also demonstrate that our co-occurrence-based ontology is reasonable for pre-aggregating noisy tags. In addition, our method also suggests a series of potential extension of supervised TM by incorporating ontology and linked open data (LOD).

2. Methodology

This paper aims to discover meaningful topics in Web APIs by combining Labeled LDA with ontology, as summarized in Fig. 1. We describe the key components as following.

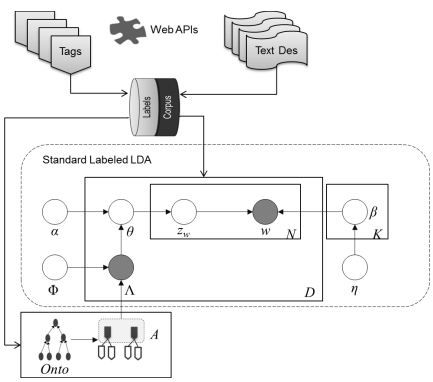

Fig. 1 Overview of the proposed method.

Manuscript received January 17, 2013.

[†]The authors are with the School of Computer Science and Technology, Tianjin University, Tianjin, 300072, China.

*This work was partially funded by the National Science Foundation of China grant No.61173155 and No.61070202.

a) E-mail: shizhan@tju.edu.cn (Corresponding author)

DOI: 10.1587/transinf.E96.D.1548

2.1 Ontology Extended Labeled LDA

Graphical model of **S**tandard **L**abeled LDA (SL-LDA) is shown in Fig. 1 (as the dotted box filed locates). Briefly, according to [4], each topic k is specified to be a distribution over vocabulary β_k from a Dirichlet prior η, then each document d is specified to be a multinomial mixture distribution $\theta^{(d)}$ over K topics from a Dirichlet prior α. These two steps exactly correspond to traditional LDA, while the most fundamental contribution of SL-LDA is to restrict $\theta^{(d)}$ to enable topics in each document d to exactly correspond to the document's labels set $\Lambda^{(d)}$, details of projection labels set can be referred to [4] (see page 3, the variables in the Graphical model are the same as [4]).

The proposed **O**ntology **E**xtended **L**abeled LDA (OEL-LDA) can be viewed as an ontology-based supervised version of SL-LDA. For the original label set $L = \{l_1, l_2, \ldots, l_s\}$, we first map labels in L into a concept set $C = \{c_{l_1}, c_{l_2}, \ldots, c_{l_s}\}$ based on ontology *Onto*. Then we aggregate each concept in C into its highest ancestor based on hierarchy of *Onto*, where the highest ancestor must be also a concept in C. Finally, We obtain an ancestor concept set $A = \{a_1, a_2, \ldots, a_t\}$, where $t < s$. We detail this process in Algorithm 1. To this end, our projection to SL-LDA is that we use ancestor concept set A as input of Λ rather than label set L, as shown in the bottom of Fig. 1.

2.2 Learning an Appropriate Ontology

Constructing a suitable ontology that can be used for pre-aggregating noisy tags is a challenge here. In this section, by investigating the associated tag data set of APIs, we propose a simple yet reasonable way to semi-automatically build an ontology based on the co-occurrences of tags.

We first model the multi-labeled APIs into an API-Label matrix as $AL = (al_{i,j})_{n\times s}$, where n denotes number of APIs, s denotes number of labels. By computing $LL = AL^T \cdot AL = (co_{i,j})_{(s\times s)}$, we can obtain a Label-Label matrix which implies the co-occurrences between tags, where value

Algorithm 1: Pre-aggregating noisy labels.

Input: Original label set L, ontology *Onto*
Output: A, and set $A.index$ denotes mapping indexes from L to A

```
C ← annotation(L, Onto); A ← ∅; A.index ← ∅
for c ∈ C do
    tmp_c ← c
    while get_father(tmp_c, Onto) ≠ "#thing" do
        tmp_c ← get_father(tmp_c, Onto)
                /* for semantic relations such as
        B kind−of A, B attribute−of A alike, we
        define A to be father of B.  */
        if tmp_c ∈ C then
            c.tmp_ancestor ← tmp_c
    c.ancestor ← c.tmp_ancestor
A ← A ∪ {c.ancestor}; A.index ← A.index ∪ {(c, c.ancestor)}
return A, A.index
```

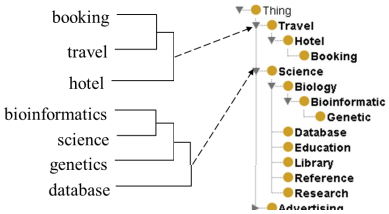

Fig. 2 Learning ontology based on the co-occurrence of labels, where left shows an illustrate of hierarchical clustering on *LL*, right presents the final ontology we extracted.

of $co_{i,j}$ implies the co-occurrence of labels i and j. We then perform a hierarchical clustering on *LL*.

As can be intuitively seen from hierarchical clustering on *LL*, labels that tend to group into the same clusters are generally representing common semantics (Fig. 2, left). Inspired by this observation, we start to semi-automatically build an ontology based on hierarchical clustering tree. For specifying the hierarchy of ontology (to define the semantic relations like *subclass-of*, *kind-of*), we generally consult some existed ontologies such as SUMO [5], OpenCyc [6], DBPedia [7] as well as Protege Ontology Library [8]. Finally, we adjust some concepts based on the maximum co-occurring label pair in *LL*.

3. Evaluation

3.1 Data Set and Experiment Setup

We evaluate our proposed method over a large set of multi-labeled APIs we crawled in August 2012. Before training TM, we perform following preprocessing to the corpus: 1) filtering non-English terms, stop words and punctuation; 2) selecting APIs with text description at least 30 terms; 3) removing terms occurring quite frequently/sparsely (terms occur too frequently/sparsely generally lose their semantics to the corpus. In this experiment, we remove terms occur less than 4 APIs and top 30 frequently used (like "*api*")); 4) selecting tags occurring at least 20 APIs, and removing meaningless tags like "*deadpool*", "*application*".

To this end, we obtained a final multi-labeled corpus consisting of 4815 APIs associated with 133 unique tags. By performing Algorithm 1, we extract a set consisting of 26 ancestor concepts from these 133 unique tags. We employ an efficient zeroth-order collapsed variational Bayes approximation algorithm (CVB0) [9] for training both SL-LDA and OEL-LDA with 1000 iterations. Table 1 summarizes some topics extracted by OEL-LDA and SL-LDA.

3.2 Topic Coherent

Traditional work usually evaluated TMs with quantitative intrinsic methods, e.g., evaluating the probability of held-out documents [10], However, [11] pointed out that such methods were insufficient for evaluating TMs since they did not take account of measuring the "coherent" of topics. Recently, Mimno *et al.* [12] introduced a new topic coherence

Table 1 Exampled topics extracted by OEL-LDA and SL-LDA. Note the row colored grey is an OEL-LDA trained topic, the rest three rows are SL-LDA trained topics. Each topic is associated with a unique name (label). Note topics presented here, the name of OEL-LDA trained topic is the ancestor concept of the rest three SL-LDA trained topics in ***A.index***.

Topics	Top 20 terms
`Travel`	travel, booking, documentation, hotel, reservations, offers, flight, reservation, sites, hotels, trip, system, book, support, flights, website, trips, location, world, based
`travel`	travel, booking, system, partners, documentation, amadeus, software, realtime, provider, agencies, providers, publicly, book, pricing, engine, reservations, direct, systems, interface, integration
`hotel`	hotel, reservation, reservations, booking, support, property, accommodations, properties, guest, management, records, functions, options, tools, payment, selection, request, including, include, additional
`booking`	travel, flight, trip, flights, sites, world, documentation, hotels, based, website, trips, location, offers, traffic, activities, apis, booking, websites, protocol, rental

score, namely *topic coherence*, which corresponds well with human annotated coherence judgments and can be used for better measuring the quality of topics than evaluating the probability of held-out documents. Generally, *topic coherence* measures each topic k by computing co-occurrence of its associated most probable m terms (see [12], page 4):

$$coherence(k) = \sum_{i=2}^{m} \sum_{j=1}^{i-1} \log \frac{C(t_i, t_j) + 1}{C(t_j)} \tag{1}$$

where $C(t_i, t_j)$ is the number of documents that terms t_i, t_j co-occur, $C(t_j)$ is the number of documents that term t_j appears. For evaluating our proposed method, we present a simple way of computing *topic coherence* as follows:

Step 1. Building a Document-Term Matrix from API corpus as $DTM = (\chi_{i,j})_{n \times w}$, where n denotes number of APIs, w denotes number of terms. $\chi_{i,j}$ denotes frequency that term j occurs in API i.

Step 2. Making words occur less than 2 times in one API:

$$\chi_{i,j} = \begin{cases} 1, & \text{for } \chi_{i,j} > 1 \quad (2a) \\ \chi_{i,j}, & \text{otherwise} \quad (2b) \end{cases}$$

Step 3. Generating a derived Term-Term Matrix:

$$TTM = DTM^T \cdot DTM = (\tau_{i,j})_{(w \times w)} \tag{3}$$

where $\tau_{i,j}$ is the number of APIs that terms i and j co-occur, $\tau_{j,j}$ corresponds to the number of APIs that term j appears. Therefore, we can easily compute *topic coherence* using the following equations:

$$C(t_i, t_j) = \tau_{t_i,t_j}, C(t_j) = \tau_{t_j,t_j} \tag{4}$$

We then evaluate the topics generated in Sect. 3.1 based on above steps, as shown in Fig. 3. Note that OEL-LDA extracted 26 topics corresponding to 26 ancestor concepts, while SL-LDA extracted 133 topics corresponding to 133 labels. The result demonstrates that our OEL-LDA outperforms SL-LDA with an average gain of 7.0% in topic quality (lager coherence implies better topic).

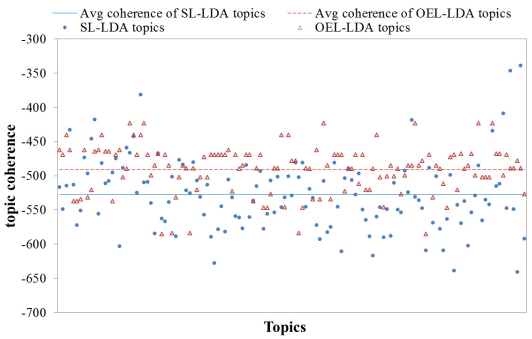

Fig. 3 Comparison of *topic coherence* of OEL-LDA and SL-LDA. Note that for comparing purpose, we plot 133 points for OEL-LDA topics using ***A.index*** mapping instead of 26 topics, which means some OEL-LDA topics are duplicated plotted (e.g., as the topics illustrated in Table 1, OEL-LDA topic "*Travel*" is plotted as 3 points here, for comparing the coherence with its related sub-concepts SL-LDA topics.).

3.3 Similarity Matching of Topics learned by SL-LDA and OEL-LDA

We attempt to evaluate relation of topics learned by SL-LDA and OEL-LDA. As can be seen from Table 1, based on the observation of the associated top 20 terms of each topic, it is obviously that the OEL-LDA topic "*Travel*" is semantically similar to topics "*hotel*", "*booking*" and "*travel*"(where the three SL-LDA topics are annotated to be sub-concepts of "*Travel*" according ***A.index***.). Thus we argue that topics extracted by SL-LDA could be most similarly matched to their annotated ancestor topics extracted by OEL-LDA.

To verify this assumption, we compute cosine similarities between all OEL-LDA topics and SL-LDA topics based on their associated top 20 terms and probability of each term contributed to each topic. We plot the results in Fig. 4 with heat maps. Each heat map corresponds to labels with one ancestor concept based on ***A.index***.

We emphasis Fig. 4 empirically demonstrates that the constructed ontology is well founded, where most SL-LDA topics capture larger similarities to their annotated ancestor OEL-LDA topics than others[†]. Note that our used ontology was constructed essentially by the co-occurrences of tags (before TM), and the result of Fig. 4 was computed by cosine similarity of TMs. Hence, this result empirically demonstrates that the constructed ontology was well founded, and performed well on pre-aggregating noisy tags. Besides, the result also implies that using ontology to pre-aggregating tags is necessary for merging similar/redundant topics extracted by SL-LDA. Furthermore, the results also suggest a new way of extracting a meaningful ontology since each tags are rich annotated with 20 terms.

[†]In Fig. 4, A few SL-LDA topics were not best matched (with high similarity) to their corresponding OEL-LDA topics, which could be related to ontology. However, this influenced little on the argument that our ontology was well founded.

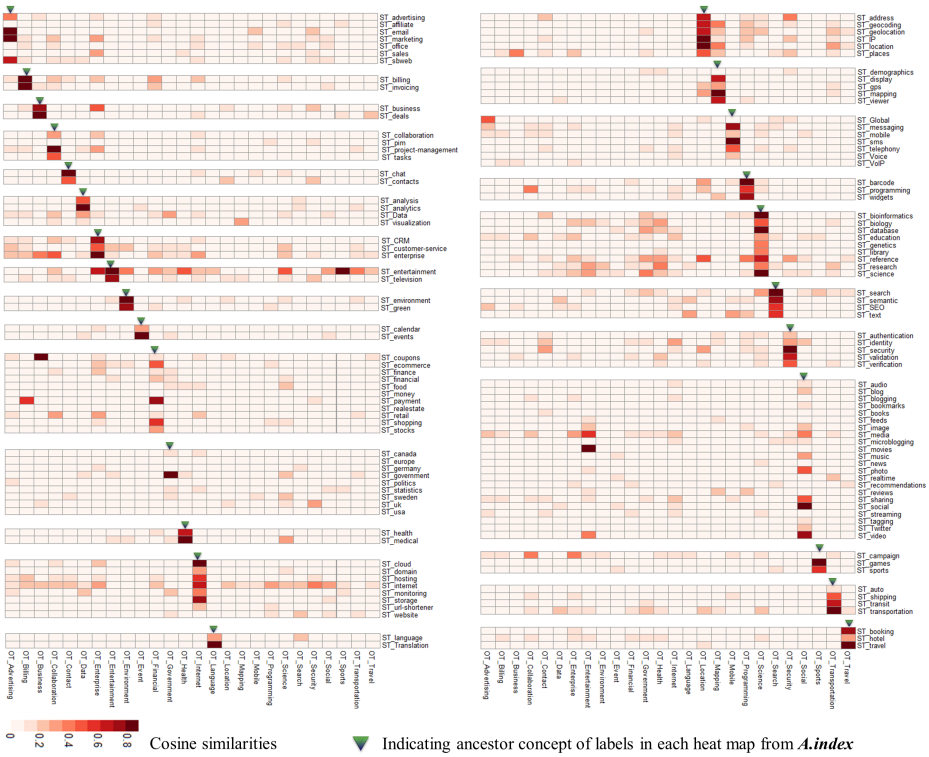

Fig. 4 Similarity matching of topics learned by SL-LDA and OEL-LDA using heat maps, where rows denote SL-LDA topics (beginning with "ST_"), columns denote OEL-LDA topics (beginning with "OT_"), colors approaching red imply better similarity matching between rows and columns. SL-LDA topics in each heat map correspond to an OEL-LDA topic (as the triangles indicate), which is the ancestor concept of the SL-LDA topics specified by ***A.index***. Thus, this figure contains 26 heat maps.

4. Conclusion

In this paper, we have shown the proposed OEL-LDA significantly refined noisy tags into unique and suitable high-level abstracting concepts, which were reasonable for representing the underlying semantics beyond "topics". We argue that our method differs from most existed enhanced LDA-like methods by using a simple yet helpful external pre-processing, rather than projecting sophisticated internal variables. We also point out that another underlying reason of leveraging ontology in OEL-LDA is that we still keep the power of SL-LDA of "supervised TM" and "naming topics" (beyond traditional LDA). We evaluated our method on real-world Web API data, and showed OEL-LDA improved SL-LDA in terms of topic quality. We also provided evidence that our co-occurrence based ontology was well founded.

We highlight that this work is the first step to our goals of classification, retrieval, organization, and management for the ever-increasing APIs. We also believe that this work can be used for TM on other multi-labeled corpus. More significantly, a series of potential prospects can be drawn from our work. Firstly, we have explored leveraging ontology for TM in this paper, while meaningful ontology can be also learned from TM. Secondly, our proposed method has good scalability for both TM and APIs, where ontology-based TM actually provided a flexible manner for generating different semantic granular/level topics, and since OEL-LDA essentially chained APIs, topics and concepts, we can possibly build linked APIs by incorporating LOD in OEL-LDA.

References

[1] http://www.programmableweb.com/
[2] D.M. Blei, A.Y. Ng, and M.I. Jordan, "Latent dirichlet allocation," J. Machine Learning Research, vol.3, pp.993–1022, 2003.
[3] D.M. Blei, "Probabilistic topic models," Commun. ACM, vol.55, no.4, pp.77–84, 2012.
[4] D. Ramage, D. Hall, R. Nallapati, and C.D. Manning, "Labeled LDA: A supervised topic model for credit attribution in multi-label corpora," Proc. EMNLP, pp.248–256, 2009.
[5] http://www.ontologyportal.org/
[6] http://www.opencyc.org/
[7] http://wiki.dbpedia.org/Ontology
[8] http://protegewiki.stanford.edu/wiki/Protege_Ontology_Library
[9] A.U. Asuncion, M. Welling, P. Smyth, and Y.W. Teh, "On Smoothing and Inference for topic models," Proc. UAI, pp.27–34, 2009.
[10] H.M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno, "Evaluation methods for topic models," Proc. ICML, pp.1105–1112, 2009.
[11] J. Chang, J.L. Boyd-Graber, S. Gerrish, C. Wang, and D.M. Blei, "Reading tea leaves: How humans interpret topic models," Proc. NIPS, pp.288–296, 2009.
[12] D.M. Mimno, H.M. Wallach, E.M. Talley, M. Leenders, and A. McCallum, "Optimizing semantic coherence in topic models," Proc. EMNLP, pp.262–272, 2011.