WiiCluster: a Platform for Wikipedia Infobox Generation

Kezun Zhang[§], Yanghua Xiao[§*], Hanghang Tong[‡], Haixun Wang[†], Wei Wang[§]
[†]School of Computer Science, Shanghai Key Laboratory of Data Science, Fudan University, Shanghai, China
[§]{kzhang12, shawyh, weiwang1}@fudan.edu.cn

[‡] Arizona State University, USA
[‡]hanghang.tong@asu.edu

[†]Google Research, USA
[†]haixun@google.com

ABSTRACT

Wikipedia has become one of the best sources for creating and sharing a massive volume of human knowledge. Much effort has been devoted to generating and enriching the structured data by *automatic* information extraction from unstructured text in Wikipedia. Most, if not all, of the existing work share the same paradigm, that is, starting with information extraction over the unstructured text data, followed by supervised machine learning. Although remarkable progresses have been made, this paradigm has its own limitations in terms of effectiveness, scalability as well as the high labeling cost.

We present WiiCluster, a scalable platform for automatically generating infobox for articles in Wikipedia. The heart of our system is an effective *cluster-then-label* algorithm over a rich set of semi-structured data in Wikipedia articles: *linked entities*. It is totally unsupervised and thus does not require any human label. It is effective in generating semantically meaningful summarization for Wikipedia articles. We further propose a cluster-reuse algorithm to scale up our system. Overall, our WiiCluster is able to generate nearly 10 million new facts. We also develop a web-based platform to demonstrate WiiCluster, which enables the users to access and browse the generated knowledge.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Clustering; Information filtering

Keywords

Knowledge Extraction; Summarization; Cluster Visualization

1. INTRODUCTION

Wikipedia[1] has become one of the best sources for creating and sharing a massive volume of human knowledge. Among others, an important reason that makes it extremely valuable is that part of its data is *structured*, and hence machine processible. Usually, a Wikipedia article is about an entity. Many Wikipedia articles contain structured information such as *table*, *text*, *hyper link*, etc., all of which are the targets of information extraction. More importantly, many entities are associated with an *infobox* which consists of a set of (*property*, *value*) pairs about the entities. Such structured information is the core building block behind many applications, including search engines, for answering user questions about these entities, etc. But the current infobox in Wikipedia is often *incomplete* [3].

In this paper, we present WiiCluster, a scalable platform for automatically generating structural information to supply knowledge for infobox in Wikipedia. Instead of performing information extraction over unstructured natural language text directly, we focus on a rich set of semi-structured data in Wikipedia articles: *linked entities*. A Wikipedia article typically consists of many links to other Wikipedia articles. Intuitively, the author of the article, in describing a Wikipedia entity, refers the reader to many other entities that are important or related to the entity. The key idea of this paper is the following: *If we can summarize the relationship between the entity and its linked entities, we immediately harvest some of the most important information about the entity.*

In order to convert such semi-structured data (i.e., linked entities) to the structure infobox, we propose an effective *cluster-then-label* algorithm to map the *(cluster-label, cluster)* to *(property, value)* pairs. For example, the article *"Shanghai"* in Wikipedia has the linked entity like *"The Bund"*,*"Oriental Pearl Tower"*,*"Fudan University"*,*"Shanghai Jiao Tong University"*. We group *"The Bund"*,*"Oriental Pearl Tower"* together by clustering and assign a label (such as *"Visitor Attractions in Shanghai"*) to the cluster. The cluster label explains how the entities in the cluster are related to *"Shanghai"* and can be considered as a property of *"Shanghai"*. Similarly we can group *"Fudan University"*,*"Shanghai Jiao Tong University"* together and generate a corresponding label *"Universities in Shanghai"*.

Recently, extensive effort has focused on expanding and enriching the structured data by *automatic* information extraction from unstructured text in Wikipedia [3, 2] to complete infobox. Although remarkable progresses have been made, its effectiveness and scalability are still somewhat limited (related work for detail).

Our approach outperforms above methods in scale and efficiency, because we harvest knowledge from all linked entities instead of limiting "property" in the infobox template (scale guaranteed), and knowledge are harvested by summarizing instead of pairwise relationship extracting (efficiency guaranteed). Even if the quality of our method might not be as good as those by dedicated human labeling, since our "cluster-label" is not constrained as "property" in infobox template, the infobox we generate could still represent knowledge in human sense and provide a starting point for further manual editing by human (which might save part of their labeling/editing cost).

[*]Correspondence author. This work was partially supported by the National NSFC(No.61472085, 61171132, 61033010), by National Key Basic Research Program of China under No.2015CB358800, by Shanghai STCF under No.13511505302, by NSF of Jiangsu Prov. under No. BK2010280, by the NSF under Grant No. IIS1017415, by the Army Research Laboratory under Cooperative Agreement No. W911NF-09-2-0053, by Defense Advanced Research Projects Agency (DARPA) under Contract No. W911NF-11-C-0200 and W911NF-12-C-0028, and by Region II University Transportation Center under the project No. 49997-33 25.

CIKM'14, November 3–7, 2014, Shanghai, China.
ACM 978-1-4503-2598-1/14/11.
http://dx.doi.org/10.1145/2661829.2661840 .

[1] http://www.wikipedia.org

Our WiiCluster enjoys three key advantages. First, it is *effective* in generating semantically meaningful summarization for Wikipedia articles. Second, it is totally *unsupervised* and thus does not require any human label. Third, it is *scalable* by adopting an efficient cluster-reuse algorithm. Overall, our WiiCluster is able to generate nearly 10 million new facts. We also develop a web-based platform to demonstrate WiiCluster, which enables the users to access and browse the generated knowledge.

2. DEMONSTRATING WIICLUSTER

In this section, we demonstrate the main functionalities of WiiCluster. We implement the WiiCluster in Java. The knowledge is stored in MySql database and can be accessed on our web platform `http://gdm.fudan.edu.cn/WiiCluster`. The platform receives a user's search request and retrieves its infobox generated by WiiCluster from the database. It further supports the browsing and visualization of the intrinsic knowledge of a given infobox.

Scale of WiiCluster. We run the experiments on Wikipedia (released in January 1, 2013), which has 3.2 million English articles in total. After removing the noisy, unrelated linked entities and filtering the linked entities without feature for clustering, we have 1.95 million valid articles in total. Our WiiCluster finds 9.8M clusters for these 1.95M articles. On average, we find 5 clusters for each article; and 3.3 entities for each cluster. If we treat the *(article entity, property, an entity in a cluster)* as a single fact, WiiCluster generates 32M such facts in total.

Display Platform. Figure 1 shows the main interface of WiiCluster. It has a search box on the top. If the user types in an entity name (e.g., *Shanghai*), WiiCluster retrieves its infobox from the database and displays it in the main screen. There are three main parts/views to browse a given infobox. Let us use the example of *"Shanghai"* to illustrate these views. In Figure 1, part A lists the primary property index in the alphabetical order (e.g., *"airports", "attractions"*). A user can be further navigated to browse the corresponding dependent clusters in part B. Part B shows all the primary properties and their dependent clusters, each of which is composed of a secondary property and one or more linked entities. For example, the two primary properties *"attractions"* and *"universities"*, each of which has a dependent cluster. Furthermore, secondary property of *"attractions"* is *"visitor attractions in Shanghai"* and it contains 18 entities such as *"The Bund", "Oriental Pearl Tower"* etc. famous tourist attractions in *Shanghai.* The secondary property represents the label of a group of the linked entities in the same cluster. Each entity can be used to navigate the user to browse the information of the corresponding article. In part C, we use prefuse (an open source graph view package)[2] to visualize the intrinsic knowledge of a given infobox. For example, if the user clicks the primary property *"districts"*, its dependent cluster will be shown in an aggregated area.

We present two different models to visualize the clusters: (a) *aggregate graph model* and (b) *tree model.* For example, If the user clicks the entry *Graph* in part C, the aggregate graph viewer will show. This visualization model displays the properties and values in the form of a graph, where all the properties and entities are represented as nodes and their dependency are represented as the directed edges. The nodes in one cluster are wrapped within an aggregated area. In the display panel, the central node is the article (*"Shanghai"*), the red node represents the primary property of the article, the linked aggregated areas represent their inner clusters. Each aggregated area composes of a green node and a set of white nodes. The green node represents the secondary property, and it is dependent on the primary property by which it is linked. And the linked white nodes represent the linked entities, which depend on the secondary property. By default, we display the article and its primary properties in the panel only, and all the inner clusters are hidden. A user can click the primary property to display or hide its inner clusters.

The generated summarization WiiCluster is meaningful and sensible. In the *"Shanghai"* example, we generate a number of clusters of entities with the primary and secondary properties. Each cluster and the corresponding properties are good supplement for the infobox of the article.

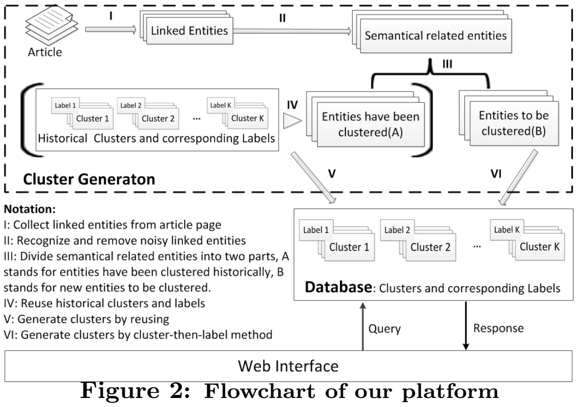

Article
I
Linked Entities
II
Semantical related entities
III
Label 1
Cluster 1
Label 2
Cluster 2
...
Label K
Cluster K
Historical Clusters and corresponding Labels
IV
Entities have been clustered(A)
Entities to be clustered(B)
Cluster Generaton
V
VI
Notation:
I: Collect linked entities from article page
II: Recognize and remove noisy linked entities
III: Divide semantical related entities into two parts, A stands for entities have been clustered historically, B stands for new entities to be clustered.
IV: Reuse historical clusters and labels
V: Generate clusters by reusing
VI: Generate clusters by cluster-then-label method
Database: Clusters and corresponding Labels
Query
Response
Web Interface

Figure 2: Flowchart of our platform

3. TECHNICAL DETAILS

In this section, we present the algorithmic details. Figure 2 shows the overall flowchart of WiiCluster. From algorithmic perspective, there are two challenges, including *C1. how to accurately summarize linked entities?*, and *C2. how to efficiently extract knowledge for all articles?*. Next, we describe an effective *cluster-then-label* algorithm and a *cluster reuse* strategy to address these two challenges, respectively. Details of the algorithms and quantitative comparison can be found in [4].

3.1 Cluster-then-Label

In order to convert the semi-structured linked entities into structured *(property, value)* pairs as infobox in Wikipedia, we need to group the similar linked entities (i.e., values) together as well as assign a label (i.e., property) for each group.

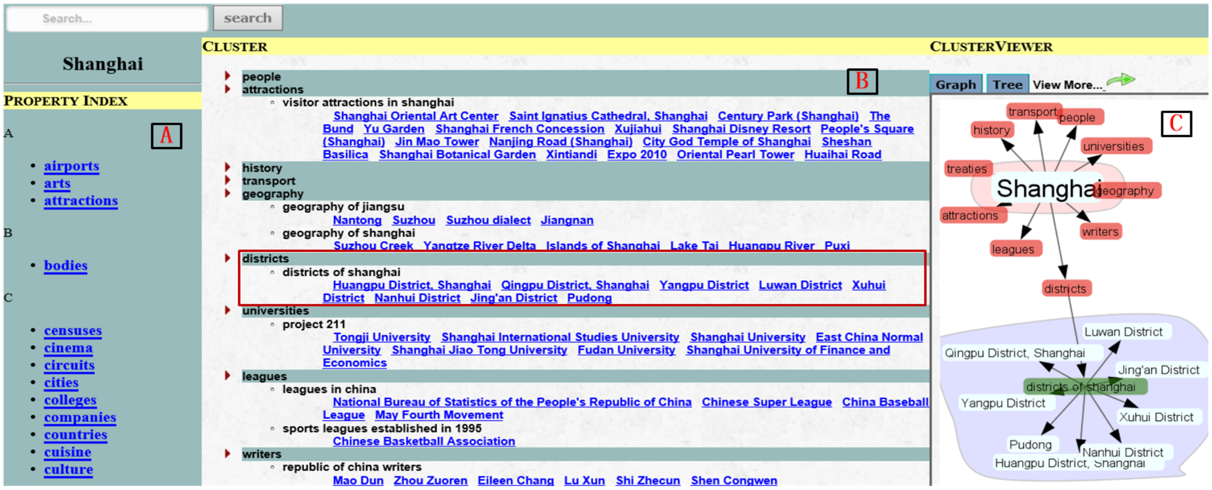
We thus propose a "cluster-then-label" approach: we cluster linked entities into different semantic groups, and then assign each group a semantic label (a property). More specifically, we use a G-means based clustering algorithm to cluster the linked entities into different semantic groups. And further propose a LCA (Least Common Ancestor) based label generating algorithm to assign a label for each group. Each labeled group is eventually a candidate (*property*, *value*) pair for the infobox.

A. Clustering. In Wikipedia, an entity is typically associated with one or more categories by editors. A category is widely used to characterize the concept of an entity. Hence, we use the categories to construct the feature vector for the entity.

However, the clustering performance is poor if we use the categories directly as the feature vector, mainly due to the following two limitations. First, some categories are not hypernyms of the corresponding entities, which may lead to mis-clustering. Second, direct categories of article are usually too specific, which leads to the small clusters with the limited number of linked entities. To address these two limitations, we propose a *feature expansion-and-weighting* procedure to construct the feature vectors for the entities. To be specific, for a given entity e, we recursively extend its feature set from its direct categories to the higher level categories. For each expanded category c, we define its weight $p(c|e)$ as the probability of category c being a hypernym of the entity e. The higher the weight $p(c|e)$ is, the more the clustering algorithm will reply on this expanded feature c. Once we have constructed the feature vectors for all the linked entities, many off-the-shelf clustering algorithms can be plugged in. In our current implementation, we use G-means [1] algorithm to cluster the linked entities with the cosine distance that is defined on the expanded and weighted feature vectors.

B. Labeling. Next, we assign a semantic label for each group/cluster. In this way, we could explain why the group of entities are linked by the article entity. The semantic label as well as the group of entities thus becomes a property of the article entity and its corresponding value. This information might provide a good supplement of the current infobox. In WiiCluster, we design the following two labeling methods, including *Category as Label* and *Word as Label.*

[2] `http://prefuse.org`



Shanghai

PROPERTY INDEX

A

- airports
- arts
- attractions

B

- bodies

C

- censuses
- cinema
- circuits
- cities
- colleges
- companies
- countries
- cuisine
- culture

CLUSTER

people

attractions

- visitor attractions in shanghai

  Shanghai Oriental Art Center Saint Ignatius Cathedral, Shanghai Century Park (Shanghai) The Bund Yu Garden Shanghai French Concession Xujiahui Shanghai Disney Resort People's Square (Shanghai) Jin Mao Tower Nanjing Road (Shanghai) City God Temple of Shanghai Sheshan Basilica Shanghai Botanical Garden Xintiandi Expo 2010 Oriental Pearl Tower Huaihai Road

history

transport

geography

- geography of jiangsu

  Nantong Suzhou Suzhou dialect Jiangnan

- geography of shanghai

  Suzhou Creek Yangtze River Delta Islands of Shanghai Lake Tai Huangpu River Puxi

districts

- districts of shanghai

  Huangpu District, Shanghai Qingpu District, Shanghai Yangpu District Luwan District Xuhui District Nanhui District Jing'an District Pudong

universities

- project 211

  Tongji University Shanghai International Studies University Shanghai University East China Normal University Shanghai Jiao Tong University Fudan University Shanghai University of Finance and Economics

leagues

- leagues in china

  National Bureau of Statistics of the People's Republic of China Chinese Super League China Baseball League May Fourth Movement

- sports leagues established in 1995

  Chinese Basketball Association

writers

- republic of china writers

  Mao Dun Zhou Zuoren Eileen Chang Lu Xun Shi Zhecun Shen Congwen

CLUSTERVIEWER

Figure 1: Screenshot of WiiCluster: searching entity *"Shanghai"* and clicking primary property *"districts"*.

Category as Label. Generally speaking, a good cluster label should capture the common theme of entities within each cluster *completeness* and in the meanwhile differentiate itself from other clusters *informativeness*. The completeness and the informativeness could be contradicting to each other. In general, the more abstract a label is, the more entities it can cover, but the less informative it might be.

Since we use the (directed and indirected) categories as the feature vectors of the entities, we can select an appropriate category as the cluster label. In order to carefully balance the completeness and informativeness, we propose a LCA (Least Common Ancestor) model. The LCA model is defined on the taxonomy graph G(Wikipedia category system). Given a cluster of entities, we search the LCA of these entities in the taxonomy G. The LCA is the nearest category which is reachable from all entities in the cluster via the hypernym edges in the taxonomy graph G. In this way, the label we get is specific enough (i.e., informativeness) yet still covers the entire cluster (i.e., completeness).

Word as Label. Using the above labeling strategy, we assign an appropriate category for each cluster. However, the categories in Wikipedia are manually edited, which are often represented as a phrase and organized in a specific Wiki style. Thus, some categories might be hard for the machine to understand (e.g., *"people by status"* instead of *"people"*). Therefore, our WiiCluster uses additional strategies to generate a more structural and machine processable label.

To distinguish these two labeling methods, we refer to the category label as the *secondary property*, and the word label as *primary property* since the word label might be upper concept of the category label.

3.2 Cluster Reuse

If we apply the above *cluster-then-label* procedure to each of the million of articles in wikipedia independently, the computation quickly becomes the bottleneck. In order to address the scalability issue, we propose a novel *cluster reuse* strategy to speedup the computation. Here, the key observation is that different articles might share many common linked entities. Thus, once the knowledge extraction for one article is done, the other article might reuse/inherit its summarization result instead of recomputing from scratch. For example, the two Wikipedia articles, *"Shanghai"* and *"Pudong"*, have some common linked entities such as *"Huangpu District", "Yangpu District"* etc. district in Shanghai. When we process *"Shanghai"*, suppose that we have grouped *"Huangpu District", "Yangpu District"* together with the label *"Districts of Shanghai"*. Thus, when we process *"Pudong"*, we might be able to directly inherit/reuse this result. Apparently, this strategy would be much more efficient in terms of computation. To do this, we construct a maximal spanning tree from the article graph (article as node, hyper link between articles as edge) in Wikipedia, and serve weight of edge as percent of their common linked entities. It's obvious that the more common linked entities exist, the larger probability reuse working and the less time cost on direct clustering and labeling.

4. RELATED WORK

Extensive efforts have focused on extracting structural information from unstructural text from Wikipedia to supply knowledge for infobox, such as [3, 2]. Limitation for these methods are: First, these methods, such as [2], rely on several natural language understanding tasks (e.g., named entity recognition, dependency parsing, and relationship extraction), which themselves are extremely challenging and error prone. [2] can harvests knowledge only from the sentences that contain both the article or its variation and the linked entity, it's obvious time consuming and is not scale guaranteed. Second, many of the existing approaches are costly, such as [3], since they are essentially supervised learning methods, and hence require a large amount of labeled training examples. In our approach, we harvest structural information though summarizing linked entities instead of pairwise relationship extraction or template constrained extraction, then we will harvest more knowledge in an efficient manner.

5. CONCLUSION

Discovering and enriching structural information in online encyclopedia is a valuable yet challenging task. We present WiiCluster, a scalable platform for automatically generating infobox for articles in Wikipedia. Unlike the existing "Information Extraction+Supervised Learning" paradigm, WiiCluster offers a radically different, yet promising avenue for automatic infobox generation. The heart of our system is an effective *cluster-then-label* algorithm over a rich set of semi-structured data in Wikipedia articles: *linked entities*. It has three key features, including (a) requiring no human labels; (b) effective in generating meaningful knowledge and (c) scalable to millions of new facts. We demonstrate its main functionalities through web-based platform.

6. REFERENCES

[1] G. Hamerly and C. Elkan. Learning the k in k-means. In *Proc. of 17th NIPS*, 2003.

[2] D. P. Nguyen, Y. Matsuo, and M. Ishizuka. Exploiting syntactic and semantic information for relation extraction from wikipedia. *IJCAI07-TextLinkWS*, 2007.

[3] A. Sultana, Q. M. Hasan, A. K. Biswas, S. Das, H. Rahman, C. Ding, and C. Li. Infobox suggestion for wikipedia entities. In *Proc. of CIKM*, 2012.

[4] K. Zhang, Y. Xiao, H. Tong, H. Wang, and W. Wang. The links have it: Infobox generation by summarization over linked entities. arXiv:1406.6449, 2014.