GrandBase: generating actionable knowledge from Big Data

Xiu Susie Fang and Quan Z. Sheng
Macquarie University, Sydney, Australia

Xianzhi Wang
University of New South Wales, Sydney, Australia

Anne H.H. Ngu
Texas State University, San Marcos, Texas, USA, and

Yihong Zhang
Nanyang Technological University, Singapore

Received 6 January 2017
Revised 14 March 2017
Accepted 15 March 2017

Abstract

Purpose – This paper aims to propose a system for generating actionable knowledge from Big Data and use this system to construct a comprehensive knowledge base (KB), called GrandBase.

Design/methodology/approach – In particular, this study extracts new predicates from four types of data sources, namely, Web texts, Document Object Model (DOM) trees, existing KBs and query stream to augment the ontology of the existing KB (i.e. Freebase). In addition, a graph-based approach to conduct better truth discovery for multi-valued predicates is also proposed.

Findings – Empirical studies demonstrate the effectiveness of the approaches presented in this study and the potential of GrandBase. The future research directions regarding GrandBase construction and extension has also been discussed.

Originality/value – To revolutionize our modern society by using the wisdom of Big Data, considerable KBs have been constructed to feed the massive knowledge-driven applications with Resource Description Framework triples. The important challenges for KB construction include extracting information from large-scale, possibly conflicting and different-structured data sources (i.e. the knowledge extraction problem) and reconciling the conflicts that reside in the sources (i.e. the truth discovery problem). Tremendous research efforts have been contributed on both problems. However, the existing KBs are far from being comprehensive and accurate: first, existing knowledge extraction systems retrieve data from limited types of Web sources; second, existing truth discovery approaches commonly assume each predicate has only one true value. In this paper, the focus is on the problem of generating actionable knowledge from Big Data. A system is proposed, which consists of two phases, namely, knowledge extraction and truth discovery, to construct a broader KB, called GrandBase.

Keywords Big Data, Information extraction, DOM trees, Knowledge bases, Multi-valued predicates, Truth discovery

Paper type Research paper

PSU Research Review
Vol. 1 No. 2, 2017
pp. 105-126
Emerald Publishing Limited
2399-1747
DOI 10.1108/PRR-01-2017-0005

1. Introduction

The modern Web has gradually evolved into a huge information repository with hidden knowledge, thanks to the unprecedented information explosion. To exploit the full potential and support unified representation of such knowledge, knowledge base (KB) construction has become an important research topic for both database and knowledge management communities. Recent years have witnessed a proliferation of large-scale KBs, including academic KBs, such as YAGO (Suchanek *et al.*, 2007), NELL (Carlson *et al.*, 2010), DBpedia (Auer *et al.*, 2007), Elementary/DeepDive (Niu et al., 2012) and industrial KBs, such as those constructed by Microsoft[1], Google[2] and Facebook[3]. These KBs store millions of facts about the real world, including named entities, their semantic classes and their mutual relationships. The majority of current KBs store data in the form of {subject, predicate, object}, or *Resource Description Framework* (RDF) triples, which we call *actionable knowledge*. Such knowledge holds the potential to efficiently and effectively change human lives by enabling technologies such as disambiguation, deep reasoning, machine reading, semantic search in terms of entities and relations and entity-level linkage for the Web of data.

Despite the large scale of existing KBs, they are still far from being complete and accurate. For example, Freebase, the largest open-source KB (Bollacker *et al.*, 2008), covers 25 million entities, but only 4, 000 properties (note that in Freebase, predicates are referred to as properties). The type *University* has only nine properties in Freebase, whereas a person can easily spot more properties in real life. Another example is that a large number of people in Freebase have no known place of birth or nationality, owing to the conflicts that reside in multi-sourced data. In fact, the coverage for less common predicates and the values for multi-valued predicates might be even lower. As KB construction involves extracting information from large-scale, possibly conflicting, and different-structured data sources and determining the data veracity by estimating the reliability of data sources given the conflicting multi-sourced data, two of the major reasons regarding the unsatisfied coverage and accuracy of the existing KBs are the unsolved *knowledge extraction* and *truth discovery* problems (Fang, 2015).

To solve the knowledge extraction problem, tremendous knowledge extraction techniques (i.e. extractors) have been proposed to obtain machine-readable and interpretable knowledge from structured (e.g. relational databases), semi-structured (e.g. Extensible Markup Language [XML]) and/or unstructured (e.g. texts, documents, images) sources (Liu *et al.*, 2003; Bing *et al.*, 2011; Kopliku *et al.*, 2011; Grishman, 2012). However, there are two limitations with the current approaches:

(1) Most existing KBs, such as Freebase, DBpedia and DeepDive, are constructed by applying extractors that focus on extracting knowledge from a single type of data source (e.g. Web texts). In particular, these KBs simply remove tags and extract data from plain texts, and ignore the knowledge contained in the Document Object Model (DOM) tree structures formed by the tags. For this reason, these KBs cannot exploit the full knowledge contained in the data sources, leading to limited coverage and quality of the extractions. In fact, various types of data sources, such as DOM trees, HTML tables and human-annotated pages (Dong *et al.*, 2014a), can be used for more accurate and complete knowledge extraction.

(2) Previous research efforts commonly focus on extracting facts of entities in a *predefined ontology*, limiting the coverage of extractions.

Although several approaches, such as open information extraction (Open IE) (Etzioni *et al.*, 2011), manage to add new entities and relations to the extractions, they fail to distinguish synonyms, therefore introducing extra redundancy to the results.

To tackle the truth discovery problem, considerable research efforts have also been conducted (Galland *et al.*, 2010; Pasternack and Roth, 2010; Yin *et al.*, 2008; Li *et al.*, 2014b). Although the existing approaches apply different models and incorporate various implications such as data types, source dependency and source reliability, to iteratively evaluate value veracity and estimate source reliability from each other, they commonly assume every predicate has only one true value (i.e. *single-valued assumption*), which cannot reflect the fact that we can easily find various multi-valued predicates, such as the children of a person and the authors of a book, in reality. Specifically, the existing approaches do not consider the functionality degree of predicates. Given a predicate, they simply regard the value set – that is, several individual values for multi-valued predicate or single value for single-valued predicate – provided by a source as a joint single value, and identify the value set with the highest confidence score as the truth. This principle will impair the accuracy of the approaches, because it ignores the correlations among the value sets of different sources. For example, there may be overlaps between two sources' claimed value sets, indicating that the two sources partially support each other. In addition, by making the single-valued assumption, the current approaches regard the false positives and false negatives made by sources as equivalent. The consequence is, given multi-valued predicates, they cannot distinguish two types of sources – some sources are *cautious* by providing partial true values without erroneous values, making more false negatives, while some sources are *audacious* by providing erroneous values, making more false positives. In a nutshell, our work makes the following main contributions:

- Aiming at *Gene*rating *a*ctionable k*n*owledge from Big *D*ata, we propose a system, which consists of two phases, namely, *knowledge extraction* and *truth discovery*, as an overall solution to construct a comprehensive KB, called *GrandBase*.
- We propose a novel framework that extracts and merges the predicates from four types of sources, existing KBs (i.e. Freebase and DBpedia), query stream, Web texts and DOM trees, for comprehensive ontology augmentation. In particular, we first extract predicates from existing KBs and query stream as seeds. Then, we use those seeds to learn tag path patterns (from DOM trees) and lexical and parse patterns (from Web texts). Those patterns are in turn leveraged to extract new predicates from DOM trees and Web texts.
- As single-valued truth discovery has been widely studied, we propose a graph-based approach to conduct better truth discovery for multi-valued predicates. Two graphs are constructed by modeling the two-sided inter-source agreements. Random walk computations are applied on both graphs to derive two-sided source vote counts for value veracity estimation.
- We conduct extensive experiments to demonstrate the effectiveness of our approaches. In particular, empirical studies show that our extractors can increase the number of predicates effectively for five typical types in Freebase. Experiments on two large real-world data sets show that our truth discovery approach outperforms the state-of-the-art baseline methods. We also discuss the future research directions regarding GrandBase construction and extension.

The remainder of this paper is organized as follows. Section 2 provides an overview of GrandBase construction. Section 3 presents in detail our approaches for predicate extraction and multi-valued truth discovery. We report our experimental results in Section 4 and review the related work in Section 5. In Section 6, we suggest several future research directions. Finally, Section 7 provides some concluding remarks.

2. Overview of GrandBase construction

Resource description framework, or RDF, is a machine-readable and -interpretable data model, which describes information about resources (particularly Web resources) in the form of {subject, predicate, object} triples, where a subject represents a resource, a predicate denotes the property of a resource or the relationship between resources and an object depicts the value of a certain property or the resource that has correlation with a certain subject. The data structure of RDF is so simple that it has been widely used to model disparate, abstract concepts, and is fed to knowledge management applications. A data set of RDF triples is essentially a large labeled and expressive directed multi-graph. As such, an RDF-based data model is more naturally applicable to represent certain types of knowledge than the other ontological models. Moreover, the majority of existing KBs store RDF triples which can be used as priors for broader KB construction (Dong *et al.*, 2014a). Therefore, we refer to the collection of RDF triples as *actionable knowledge*, and the goal of our work is to generate larger amount of more accurate RDF triples based on the data extracted from the Web.

In our system, KB construction involves two main phases, namely, *knowledge extraction* and *truth discovery*. Generally, the knowledge extraction phase contains three tasks (Dong *et al.*, 2014b):

(1) *Triple identification*: The goal is to identify which words or phrases demonstrate triples in the Web.

(2) *Entity linkage*: It aims at linking entities in a predefined ontology to the words or phrases in the Web.

(3) *Predicate linkage*: It aims to decide which predicate in a predefined ontology a word or phrase it refers to.

Owing to the diverse reliability of different sources and the varied capacities of various extractors, it is common to observe conflicts in the extracted triples. The truth discovery phase aims at reconciling those conflicts. Figure 1 shows an overview of the framework for GrandBase construction. We introduce more details of the two phases as follows.

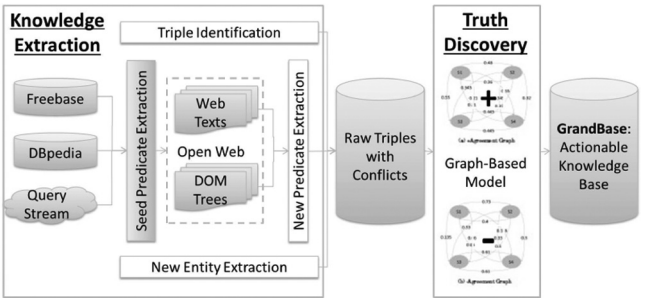

Figure 1.
The framework of GrandBase construction: white rectangles with underlined labels represent the two main phases of GrandBase construction, the three small white rectangles inside the knowledge extraction rectangle depict the three tasks of knowledge extraction

2.1 Knowledge extraction phase

We apply the open IE approach to extract RDF triples from four types of sources including query stream, existing KBs (Freebase and DBpedia), Web texts and DOM trees. To construct a more complete KB, we propose to augment the ontology of Freebase, as Freebase contains the largest number of entities and *isA* pairs. In particular, we change the traditional tasks of knowledge extraction, namely, *predicate linkage* and *entity linkage*, to *new predicate extraction* (i.e. discovering new predicates from the Web content and attaching them to the corresponding classes to augment the ontology) and *new entity extraction* (i.e. identifying new entities described in the Web content and attaching them to the corresponding classes to augment the ontology). For new predicate extraction, as the data in query stream and existing KBs would be more accurate, we first extract predicates from those sources. Then, we use the extractions as seed to learn extraction patterns of the open Web (Web texts and DOM trees), which are in turn used to extract more new predicates from the Web. Because of the differed features of Web texts (often presented by natural languages) and DOM trees (semi-structured data described by tags), we apply different extractors on them. In particular, as Web text extraction has been widely studied, we focus on DOM tree extraction in our work. For Web texts, we first perform standard natural language processing (NLP) technique; apply distant supervision to induce lexical and parse patterns, which are unified syntax rules over the Web; and finally, leverage these patterns to extract predicates from Web texts. For DOM tree extraction (explained in Section 3.1), as websites are different from each other in display style and format, no unified *tag path pattern* could be found that is applicable to all the Web pages. To this regard, our extractor learns tag path patterns for each Web page and then uses these patterns to extract new predicates from the Web pages. There is a work (Wick *et al.*, 2013b) related to new entity extraction in the literature, which jointly solves entity-linking and entity-discovery; our framework seeks to incorporate this technique to broaden the number of entities accommodated in GrandBase. To further enhance the ontology, we also conduct misspelling, synonym, sub-predicate identification (Gupta *et al.*, 2014). Finally, we propose to apply this enhanced ontology to explore more facts from the open Web[4].

2.2 Truth discovery phase

Our work relaxes the single-valued assumption commonly made by the previous work and tackles a more general problem, i.e. multi-valued truth discovery, which is defined as follows.

Definition 2.1 Multi-Valued Truth Discovery. Given a set of predicates (P), each of which may contain multiple true values, and the conflicting claimed values (V) collected from a set of sources (S), the goal is to identify a set of true values (V_p) from V, for each predicate p, satisfying that V_p is as close to the ground truth as possible.

Given a multi-valued predicate, the value sets provided by sources may be the same, totally different, or overlapping. To differentiate the cautious and audacious sources and be aware of false positives and false negatives, we propose to measure source reliability by *positive precision* and *negative precision*. Specifically, given a predicate p, suppose U_p is the set of all potential values of p, V_{s_p} is the set of values claimed by source s (i.e. *positive claims*), indicating that s believes that the values in V_{s_p} are true. By applying *mutual exclusion*, we consider s votes against all the other potential values, and regard $U_p - V_{s_p}$, denoted by $\overline{V}_{s_p}$, as the *negative claims* of s. Given a source, the positive (resp., negative) precision represents the probability of the positive (resp., negative) claims being true (resp., false). Intuitively, if the positive (resp., negative) claims of a source are agreed by the majority of other sources, this source is likely to have high positive (resp., negative) precision. This means that the

inter-source agreements indicate source reliability endorsement. This intuition motivates us to measure the two-sided source reliability by quantifying the two-sided agreements among sources regarding their positive claims and negative claims. The two-sided source reliability is then regarded as two-sided weighted vote for value veracity estimation. At the truth discovery phase, we design a graph-based approach to fuse the conflicts in the raw triples extracted by the extractors at the knowledge extraction phase (Section 3.2).

3. The approaches

3.1 Predicate extraction

In this section, we introduce our predicate extractors for extracting predicates from existing KBs, query stream and DOM trees [more details are referred to Fang *et al.* (2015)].

3.1.1 Predicate extraction from existing KBs. We use two dominate existing KBs, namely, Freebase (covers 4, 000 predicates) and DBpedia (covers 6, 000 predicates). We first analyze each KB separately by applying the intuition that each sub-type should inherit all the predicates of its super-types. Then we combine the predicate extractions from both KBs. In particular, we attach the predicates of each DBpedia class to the similar[5] type in Freebase. We also apply duplication removal for the combined predicates to avoid redundancy, by comparing the predicates of similar class or type in both KBs in terms of name, label and comment/description. For example, for the type "Book", there are 5 predicates attached to it in Freebase and 21 in DBpedia. After inheriting all the predicates from the corresponding super-types of "Book" in the corresponding KB, we can get 19 predicates extracted from Freebase and 48 from DBpedia. By combining those predicate extractions and applying duplication removal, we finally obtain 60 predicates to be attached to Freebase.

3.1.2 Predicate extraction from query stream. As query stream naturally captures users' collective convictions on possible predicates of entities, it is a high-quality resource for predicate extraction. We define the query stream extractor by using more patterns than previous methods, including "what/how/when/who is the p of (the/a/an) e", "the p of (the/a/an) e" and "e's p" (where e represents an entity and p depicts a possible predicate), and a set of filtering rules. Our designed patterns are applied to extract more possible predicates, while the filtering rules are used to purify the extractions. Specifically, our query stream extractor conducts a five-step process to extract predicates for each type (denoted as T) in Freebase.

The initial step is relevant query stream identification. We apply an *entity recognizer* to identify the queries that contain entities belonging to T, and denote the set of relevant queries as $Q_R(T)$. Second, we conduct predicate candidate identification. From $Q_R(T)$, our extractor finds all the queries that match any of our predefined patterns. We denote those queries as $Q_P(T)$ and add the identified predicate candidates to the set $P(T)$. For example, given a relevant query "Taken 3's box office" of type *movie*, as it matches the pattern "e's p", we add "box office" to $P(movie)$ as a predicate candidate of *movie* and this query to $Q_P(movie)$. Note that $P(T)$ may be noisy and contain non-predicate elements. For example, both queries "The University of Adelaide" and "the plural of country" match our predefined pattern "the p of (the/a/an) e", but "University" is not actually a predicate of "Adelaide", so as is "plural". Therefore, our third step focuses on filtering out non-predicates in $P(T)$. To this end, we sample a set of relevant queries that are related to various types, and extract predicate candidates for each type. Then, we rank the predicate candidates by the number of types they belong to, and add the top predicate candidates to a *blacklist* to avoid the appearance of the non-predicate elements, such as "lack", "rest", "meaning", "best", "definition", "summary" and "plural", in $P(T)$. For such cases as "The University of Adelaide", we apply named entity identification to remove this type of non-predicates from $P(T)$. In particular, we compare each query in $Q_P(T)$ with the entities of T in Freebase, and if

they match each other, we remove the corresponding predicate from $P(T)$. In the fourth step, we conduct entity–predicate pair identification. For each query in $Q_R(T)$, if the query containing both e belongs to T and $p \in P(T)$, we add the corresponding (p, e) pair to a set PE. To further purify the predicate extractions, we conduct the fifth step, credible predicate identification, by applying the following two rules. The result set $P(T)$ is attached to the type T in Freebase for ontology augmentation.

Rule 1: Given a type T, which contains N entities $\{e_1, e_2, \ldots, e_N\}$, and a predicate p, p will not be attached to T, if $\exists e_j \in \{e_1, e_2, \ldots, e_N\}$, $EntityFrequency(e_j) \geq \max_{(p,e^*)\in PE}(EntityFrequency(e^*))$ and $(p, e_j) \notin PE$.

Here $EntityFrequency(e)$ represents the number of queries in $Q_R(T)$ that cover a specific entity e.

Rule 2: For each $EntityDiversity(T, p) \neq 0$, we remove p from $P(T)$, if $\frac{EntityDiversity(T,p)}{N} \leq \alpha$ (a pre-defined threshold).

Here $EntityDiversity(T, p)$ depicts the number of distinct entities of T which co-appear with a predicate p in PE.

Algorithm 1: Algorithm for DOM Tree Extraction

Input: Given a *Type* T in Freebase; a set of Websites regarding T, denoted as $S(T)$, for each Website $s \in S(T)$, it contains a set of Web pages; the entity set $E(T)$ of T in Freebase; the seed predicate set $seed(T)$ extracted from query stream and existing KBs.

Output: Raw predicates for Type T in Freebase (i.e., enriched $seed(T)$).

1 **Initialization**: for each Web page $wp_s \in s$ and $s \in S(T)$, identify all the *entity node* and *non-entity node*, and obtain tag path set $TP(wp_s)$ for each Web page.
2 **for** *each* $s \in S(T)$ **do**
3 **for** *each* $wp_s{}^* \in s$, *and* $wp_s{}^*$ *contains at least an entity* $e \in E(T)$ *and a predicate* $p \in seed(T)$ **do**
 /* if $|seed(T)|$ is increased, the algorithm continues the loop for this Website; else the algorithm begins to traverse another Website */
4 extract the tag path(s) between e and p, and transfer them to the *induced tag path pattern set* $TP_I(wp_s{}^*)$;
5 compare all the other tag paths in $TP(wp_s{}^*)$ with the induced tag path(s) in $TP_I(wp_s{}^*)$;
6 **if** *(a tag path is similar to the induced tag path(s))* **then**
7 If the *non-entity node* is a new predicate, then add it to $seed(T)$;
8 remove the tag path from $TP(wp_s{}^*)$;

3.1.3 Predicate extraction from DOM trees. Typically, Web pages are semi-structured and described by nested HTML tags. The tree-like structures can be commonly found in Web pages that contain Web lists and Web tables, as well as deep-Web sources, and are regarded as DOM trees[6]. To explore knowledge from Web pages, traditional extractors simply remove the HTML tags and process the plain texts. Thus, they fail to exploit the knowledge contained in the DOM trees. We propose a two-step extractor to extract predicates from DOM trees to augment the ontology of Freebase: first, seeded by the predicate extractions from existing KBs and query stream, we extract additional predicates from the DOM trees; second, we purify the extractions and differentiate the quality of the extracted predicates by applying a set of filters.

At the first step, our approach alternatively extracts predicates and learns tag path patterns through an iterative process. The detailed procedure is described in Algorithm 1. Briefly, given a type T in Freebase, the algorithm first identifies the relevant Websites of T (e.g. www.imdb.com/for type *Film*), denoted by $S(T)$. For each Web page $wp_s \in s$ and $s \in S(T)$, the algorithm analyzes the DOM structure and classifies the text nodes into *entity node*s (i.e. the texts represent an entity e that belongs to T) and *non-entity node*s. The tag paths between each *entity node* and its corresponding *non-entity node* are then extracted. After the removal of noisy tags, the extracted tag paths are kept in a *tag path set*, denoted by $TP(wp_s)$ (Line 1). For each website $s \in S(T)$ (Line 2), the algorithm iteratively finds the Web pages that contain at least one (p, e)

pair, where e belongs to T, $p \in seed(T)$ and $seed(T)$ is the seed predicates extracted from existing KBs and query stream. For each eligible Web page wp_s^* (Line 3), the algorithm traverses $TP(wp_s^*)$ for this Web page to obtain the tag paths between the seed p and the corresponding e, and transfers these tag paths from $TP(wp_s^*)$ to an *induced tag path pattern set*, denoted by $TP_I(wp_s^*)$ (Line 4). We next compare all the tag paths in the $TP(wp_s^*)$ with the patterns in $TP_I(wp_s^*)$ (Line 5). Those *non-entity node*s with similar tag paths with the induced patterns are finally recognized as predicates (Line 6). If a predicate is not covered by $seed(T)$, then it is added to $seed(T)$ for augmentation (Line 7), with the corresponding tag paths removed from $TP(wp_s^*)$ (Line 8). The algorithm turns to another Website when the number of predicates in $seed(T)$ reaches a certain threshold. Because the number of Web pages and text nodes in a Web page is limited, the algorithm can always converge.

As the raw predicates extracted by the first step may contain considerable noises, in the second step, we use the following three types of features to purify the extracted predicates:

(1) The *inherent features* of a predicate displayed in a Web page. A node that denotes a predicate in a DOM tree always follows some inherent rules, e.g. the length of the text is always limited to a certain number of words (we discover that almost all predicates in Freebase is described by less than ten words), the text node always contains a colon as the end of the string or the first letter of every word in the text node is in upper case (Web page always capitalizes the first letter for each word of the name of a predicate).

(2) The *intra-site features* of a predicate displayed in a Web page. If a predicate is described by a Website, it tends to appear frequently in a considerable number of pages of this website.

(3) The *inter-site features* of a predicate displayed in a Web page.

Predicates tend to appear in multiple websites instead of very few websites.

We can simply neglect the extractions that mismatch these features, but this may result in some loss of recall. For example, in reality, the number of movies that win an Oscar award is quite limited. Thus, the predicate "winner in Oscar" would not appear frequently in the Web pages of a movie website, dissatisfying the second feature. If we strictly use the feature, we tend to not regard "winner in Oscar" as a predicate by mistake. Therefore, we sequentially use three filters to deliver three predicate sets in turn, i.e. *potential predicates*, *predicate candidates* and *credible predicates*. Each set represents a different balance between the precision and recall of the extractions and can be fed to knowledge-driven applications based on their own requirements.

In particular, we apply the *inherent feature filter* to obtain the set of potential predicates by using the specific rules followed in the DOM trees. By leveraging the *intra-site feature filter*, we remove all predicates with intra-site frequency lower than a predefined threshold β to obtain the predicate candidate set. We calculate the intra-site frequency of a predicate p in a website s by the following equation:

$$f_s(p) = \frac{N_s(p)}{N(s)} \tag{1}$$

where $N(s)$ is the number of Web pages in s, and $N_s(p)$ is the number of Web pages that contains p in Website s.

The intra-site feature filter has limitations, as it may incorrectly take some Website-specific terms as predicates. For example, for the sake of display, a node with text "edit" appears frequently in IMDb (a famous movie Website). In this case, the intra-site feature

filter would incorrectly identify *edit* as a predicate of type *Movie*. However, we notice that such predicates are often website-specific, which is in contrast seldom contained by other Web sites. Therefore, we can remove such noises by additionally examining the inter-site frequency feature of the predicate extractions. To construct the *inter-site feature filter*, we should not only consider the sum of the frequencies of a predicate in all the websites but also the distribution of the frequencies over the websites. The predicates that appear evenly and frequently in all the websites tend to be more credible. In this regard, we calculate the inter-site frequency of a predicate as follows:

$$F(p) = \sum_{j=1}^{|S|} \frac{N_{s_j}(p)}{N(s_j)} - \sum_{k=1}^{|S|} \frac{\frac{Ns_k(p)}{N(s_k)}}{\sum_{j=1}^{|S|} \frac{Ns_j(p)}{N(s_j)}} \log \frac{\frac{Ns_k(p)}{N(s_k)}}{\sum_{j=1}^{|S|} \frac{Ns_j(p)}{N(s_j)}} \tag{2}$$

where S is the set of all websites. We finally form the credible predicate set by keeping the predicates with inter-site frequencies that are higher than a predefined threshold γ.

3.2 Truth discovery

To harmonize the conflicts contained in the knowledge extractions obtained at the first phase, we present a graph-based approach as a solution for multi-valued truth discovery. Our approach consists of two steps:

(1) two-sided agreement graphs construction; and

(2) two-sided source reliability evaluation and value veracity estimation.

3.2.1 Agreement graph construction. Given a multi-valued predicate p, we formally define the common values claimed or disclaimed by two sources as *inter-source agreement*. We consider two-sided inter-source agreements based on mutual exclusion of values. In particular, + *agreement*, the agreement between two sources (e.g. s_1 and s_2) on their positive claims of p, is denoted as $A_p(s_1, s_2)$ and calculated as:

$$A_p(s_1, s_2) = V_{s_{1p}} \cap V_{s_{2p}} \tag{3}$$

Similarly, − *agreement*, the agreement between two sources on their negative claims of p, is denoted as $\tilde{A}_p(s_1, s_2)$ and calculated as:

$$\tilde{A}_p(s_1, s_2) = \tilde{V}_{s_{1p}} \cap \tilde{V}_{s_{2p}} = U - \left(V_{s_{1p}} \cup V_{s_{2p}}\right) \tag{4}$$

Based on the above quantified inter-source agreements, we construct two fully connected weighted digraphs, i.e. ± *agreement graph*s. Specifically, in each graph, each vertex denotes a source, and each weighted directed link between two sources represents that one source that agrees with/endorses the other source with an endorsement degree as weight. + Agreement graph models the + agreement among the sources, while − agreement graph focuses on capturing the − agreement among the sources.

To construct the + agreement graph, we first formalize the endorsement degree between two sources [denoted as $\mathcal{A}(s_1, s_2)$] as the average fraction of values of a source endorsed by the other source over all their shared predicates, which is calculated as follows:

$$\mathcal{A}(s_1, s_2) = \sum_{p \in P_{s_1} \cap P_{s_2}} \frac{|A_p(s_1, s_2)|}{|V_{s_{2p}}|} \tag{5}$$

where P_s is the set of predicates covered by s. By adding a "smoothing link" with a small weight between every pair of vertices, we calculate the weight of each link in + agreement graph as:

$$\omega(s_1 \rightarrow s_2) = \eta + (1 - \eta) \cdot \frac{\mathcal{A}(s_1, s_2)}{|P_{s_1} \cap P_{s_2}|} \tag{6}$$

where η is the smoothing factor that guarantees that the graph is always connected and the random walk computations can always converge. For our experiments, we simply set $\eta = 0.1$ [empirical studies such as the work done by Gleich *et al.* (2010) demonstrate more accurate estimation].

Similarly, we create the − agreement graph by applying the equations as follows:

$$\tilde{\mathcal{A}}(s_1, s_2) = \sum_{p \in P_{s_1} \cap P_{s_2}} \frac{|\tilde{A}_p(s_1, s_2)|}{|\tilde{V}_{s_{2p}}|} \tag{7}$$

$$\tilde{\omega}(s_1 \rightarrow s_2) = \eta + (1 - \eta) \cdot \frac{\tilde{A}(s_1, s_2)}{|P_{s_1} \cap P_{s_2}|} \tag{8}$$

For both graphs, we normalize the weights of out-going links from every vertex by dividing the link weights by the sum of all out-going link weights from the vertex. This normalization allows the link weights to be interpreted as the transition probabilities for the random walk computations directly.

3.2.2 Value veracity estimation. We apply the *Fixed Point Computation* (FPC) model to capture the transitive propagation of source reliability through endorsement links, based on the above constructed graphs. In particular, we consider each graph as a Markov chain, where each vertex serves as a state and each link weight serves as the probability of transition between the linked two states. We then compute the asymptotic stationary visiting probabilities of the Markov random walk. For each graph, the calculated probabilities sum up to 1, implying that they cannot be interpreted as the positive or negative precision of each source directly. However, this normalized feature renders the probabilities of each source in both graphs comparable. Also, the ranking of visiting probabilities of each graph implies the ranking of source precision. Moreover, the visiting probabilities capture the following two features:

(1) Vertices with more input links that have weights bigger than η are assigned with higher visiting probabilities, because those sources are endorsed by a larger number of sources and should be more trustworthy.

(2) Endorsement from a source with more input links that have weights bigger than η, should be more trusted than that from other sources, because authoritative sources tend to be of higher quality and the sources endorsed by authoritative sources tend to be more trustworthy as well.

Based on the above analysis, we refer to the visiting probability of each source in the + agreement (resp., − agreement) graph as a vote for its positive (resp., negative) claims being true (resp., false). Therefore, we estimate the veracity of each possible value of a predicate by:

$$Veracity(v) = \begin{cases} True; & \text{if} \sum_{s \in S_v} \mathcal{V}(s) > \theta \cdot \sum_{s \in S_{\bar{v}}} \tilde{\mathcal{V}}(s) \\ False; & \text{if} \sum_{s \in S_v} \mathcal{V}(s) < \theta \cdot \sum_{s \in S\bar{v}} \tilde{\mathcal{V}}(s) \end{cases} \tag{9}$$

where S_v (resp., $S\tilde{v}$) denotes the set of sources claim (resp., disclaim) v, $\mathcal{V}(s)$ (resp., $\tilde{\mathcal{V}}(s)$) represents the visiting probability of s in the + agreement (resp., − agreement) graph. θ is the source confidence score, which belongs to the range (0, 1). For a single-valued predicate, the mutual exclusion inherently holds. However, for the multi-valued predicates, sources may not know the number of the true values of the predicates. Thus, they do not necessarily reject their negative claims. For this reason, we introduce θ to relax the strict mutual exclusion and differentiate the confidence of each source on its positive claims and negative claims. We will study the impact of θ on the performance of our approach in Section 4.2.2.

Algorithm 2: The Algorithm of Our Graph-Based Truth Discovery Approach

Input: a set of predicates (P), and the conflicting claimed values (V) collected from a set of sources (S)
Output: V_p, identified truth for each $p \in P$.
Initialization: smoothing factor η, source confidence factor θ.
for *each* $s_i \in S$ **do**
  for *each* $s_j \in S, j \neq i$ **do**
    /* Construct ±Agreement Graphs. */
    calculate the weight of each link in +agreement graph by Equation 5, 6;
    calculate the weight of each link in –agreement graph by Equation 7, 8;
apply *FPC* to calculate $\mathcal{V}(s)$ and $\tilde{\mathcal{V}}(s)$ for each source;
for *each* $v \in V$, $p \in P$ **do**
  determine the veracity by Equation 9, and add the true values into V_p;

Algorithm 2 demonstrates the detailed procedure of our approach, which has a time complexity of $O(|\mathcal{S}|^2 + |\mathcal{V}|)$. Note that, there are many mature distributed computing tools that can be used for random walk computation to reduce the time complexity. For example, Apache Hama[7] is a framework for Big Data analytics, which uses the *Bulk Synchronous Parallel* (BSP) computing model. It includes the *Graph* package for vertex-centric graph computations. We can easily extend the *Vertex* class to create a class for realizing parallel random walk computation.

4. Experiments

In this section, we report the experiments to evaluate our approaches and discuss the results. In particular, we validate our predicate extractors in Section 4.1, and make comparative studies between our truth discovery approach and the existing methods on two real-world data sets in Section 4.2.

4.1 Experiments on predicate extractors

We implemented the proposed three types of extractors, including existing KB extractor, query stream extractor and DOM tree extractor in Java UDK 7. We ran experiments on an ASUS P550C computer with a 2.5 GHz i7 processor and 8GB RAM.

For the experimental settings, we conducted empirical studies on five representative types in Freebase, namely, *Book*, *Film*, *Country*, *University* and *Hotel*, to validate the capability of our extractors. For the entity recognition, each type is specified as a set of representative entities in Freebase Table I. Because our goal is to extract predicates rather than the facts of predicates and the entities of the same type should share the same predicates, pre-specifying the target type by a set of entities will not be a limiting factor. We used *precision* and *the number of extracted predicates* as the metric to measure the performance of our extractors. We took the voting of three volunteers to determine the precision of the results. Volunteers manually gave their opinions on whether each

predicate is reasonable for a type. The precision was calculated as the fraction of predicates that were labeled as reasonable.

4.1.1 Existing KB extractor. Because Freebase and DBpedia are relatively high-quality resources, we did not use precision to measure the predicate extractions, but simply regarded all the extractions are reasonable predicates. We first used the inheritance relations between types and their sub-types to conduct predicate extraction on each KB separately. Then, our extractor combined the extractions from both KBs. Table II shows the extraction results, indicating that combining the predicates of different KBs significantly increases the number of the predicates for each type.

4.1.2 Query stream extractor. We collected a query stream with 29,283,918 query records, which is the combination of two real-world data sets, namely, Google[8] and AOL[9]. To further study the distribution of *precision* regarding the extracted predicates, we ranked the predicates of each type by the *EntityDiversity* (T, p) of each predicate, and evaluated the precision of top-k (k = 10, 20, 50, 100) predicates for each type. Table III shows the empirical results regarding our query stream extractor, implying that the quality of extractions in terms of number and precision depends on both the number of relevant query records and the natural features of the type. In particular, as type *Country* and type *Film* both have relatively large sets of relevant query records, the precision of extractions for either of them is higher than that of other representative types. However, as type *Country* inherently has more predicates concerned by users, type *Country* had more credible

Table I.
Entities in the five representative types

Type	# Entities	Examples of entities
Book	1,200	Asia Grace, Cool Tools
Film	1,000	A Christmas Story, A Chump at Oxford
Country	727	Germany, Australia, Iran
University	1,000	Brandeis University, Maynooth University
Hotel	1,000	Hotel Sacher, Hotel Georgia

Table II.
Existing KB extraction results

	# Predicates				
Type	DBpedia	Extrac. (DBpedia)	Freebase	Extrac. (freebase)	Comb. (Freebase and DBpedia)
Book	21	48	5	19	60
Film	53	53	54	54	92
Country	191	360	22	150	489
University	21	484	9	57	518
Hotel	18	216	7	56	255

Table III.
Query stream extraction results

			Precision (%)			
Type	Relevant query records	Credible predicates	Top-10	Top-20	Top-50	Top-100
Book	259,556	96	80	65	62	N/A
Film	403,672	59	100	75	66	N/A
Country	393,244	182	100	96	95	93
University	24,633	20	100	100	N/A	N/A
Hotel	15,544	N/A	N/A	N/A	N/A	N/A

predicates (i.e. 182) extracted from a query stream than type *Film*. On the other hand, type *Hotel* has only 15, 544 relevant query records. Thus, no reasonable predicate could be found for it. For all the types, the precision of the top-k predicates peaked at $k = 10$, but decreased as k increased. This is consistent with our assumption that the predicates appear with various entities would be more credible. Although the results showed good precision (60-100 per cent), the query stream used for the experiments is still relatively small. It is reasonable to anticipate that more high-quality predicate extractions can be obtained if a larger query stream is available.

4.1.3 DOM tree extractor. We first merged the extractions obtained from the above two extractors to construct the seed predicate set. Duplication removal was also conducted during this procedure. For example, for type *Book*, which has 96 predicate extractions from query stream and 60 predicates extracted from existing KBs, 118 seed predicates were finally obtained. Induced by the seed predicates, we exploited crawler4j[10] to crawl the Web and jsoup 1.8.1 to reformat the collected Web pages. In particular, we collected 5 representative websites for each target type, and for each website, we crawled 100 representative Web pages. We filtered out all the nodes with long text (more than ten words in our case) to avoid tackling too many non-predicate nodes. Both the inter-site feature filter and intra-site feature filter were applied to further refine the extractions. Table IV shows the extraction results from DOM trees. For all the five representative types, more predicates were extracted from DOM trees than from either query stream or existing KBs. Also, more seeds tended to lead to more predicates extracted from DOM trees. The results also show the high precision achieved by our extractor with precision that ranges from 79.8 to 93.3 per cent. This is due to the phenomenon that the data contained in DOM trees are often structured and clean. Our empirical studies validate that DOM trees are high-quality resources for predicate extraction, which, unfortunately, are not considered in many recent research efforts regarding KB construction such as Biperpedia (Gupta *et al.*, 2014).

4.2 Experiments on truth discovery

4.2.1 Experimental setup. We used the following two real-world data sets in our experiments. Each predicate in both data sets is multi-valued:

(1) *Book-Author Data set* (Yin *et al.*, 2008). This data set is collected by crawling www.abebooks.com, which contains 33, 971 data records provided by numerous book stores (i.e. sources). Each record represents the positive claims provided by a specific source regarding the author list of a specific book (i.e. predicate). The ground truth contributed within the original data set was used as the gold standard. We conducted duplication removal to make the problem more challenging – otherwise, even a naive approach could return relatively high-quality results. Finally, we obtained 12, 623 distinct claims, where 649 sources provide author names on 664 books, and each book has 3.2 authors on average.

	# Predicates				
Type	Query stream	Existing KBs	Seed	DOM trees	Precision (%)
Book	96	60	118	168	81.5
Film	59	92	121	329	88.6
Country	182	489	621	725	92.7
University	20	518	536	539	93.3
Hotel	N/A	255	255	312	79.8

Table IV. DOM tree extraction results

(2) *Parent–Children Data set* (Pasternack and Roth, 2010). This data set contains 11,099,730 records about individuals' dates of birth, dates of death and/or the names of their parents/children and spouses. These records were collected from Wikipedia and were edited by numerous users (i.e. sources). We extracted the latest editing records from the data set as the ground truth for experimental purposes. For the sake of validating our multi-valued truth discovery approach, we specially extracted the records on the parent–children relations from the data set. We also removed the duplicated records for this data set, and finally, we obtained 55,259 sources claiming children for 2,579 people, where each person has 2.45 children on average.

To conduct comprehensive comparison studies, we selected three types of representative truth discovery methods as baselines.

4.2.1.1 Existing MTD (multi-valued truth discovery) methods. To the best of our knowledge, there are two methods of this type, LTM (Latent Truth Model) (Zhao *et al.*, 2012) and MBM (Multi-truth Bayesian Model) (Wang *et al.*, 2015). The former applies a probabilistic graphical model to infer source reliability and value veracity, and the latter incorporates source confidence and a finer-grained copy detection technique into a Bayesian model.

4.2.1.2 STD (single-valued truth discovery) methods. The majority of existing truth discovery methods belong to this type, because they commonly make the single-valued assumption. Note that some existing STD methods are inapplicable to the scenario of our problem. For example, the methods of Zhao and Han (2012) and Li *et al.* (2014b) focus on handling heterogeneous data and the method of Li *et al.* (2014a) is designed for continuous data, while our approach is designed for categorical data. The method of Pasternack and Roth (2010) requires the normalization of the veracity scores of values, which is infeasible for the MTD problem. We chose five typical and competitive methods from this type for comparison: *Voting*, regards a value set as true if the proportion of sources that claim the set is the highest among that of the other value sets; *Sums* (Kleinberg, 1999) and *Average-Log* (Pasternack and Roth, 2010), both of which are modified to incorporate mutual exclusion. They both compute the total reliability of all sources that claim and disclaim a value separately and label the value as true if the former is bigger than the later; *TruthFinder* (Yin *et al.*, 2008), which iteratively estimates *trustworthiness of source* and *confidence of fact* from each other by additionally considering the *influences between facts*; and*2-Estimates* (Galland *et al.*, 2010), which takes mutual exclusion into consideration.

4.2.1.3 Improved STD methods. We improved the above five STD methods by incorporating the prediction of the number of true values for each predicate. Specifically, we treated the values in each claimed value set of each source individually, and ran the original method to output source reliability and value confidence scores. Then, we predicted the number of true values for each predicate by:

$$P_p(n) = \sqrt[|S_p|]{\prod_{|V_{s_p}|=n, s\in S_p} \tau(s) \cdot \prod_{|V_{s_p}|\neq n, s\in S_p} (1-\tau(s))} \tag{10}$$

where $P_p(n)$ is the unnormalized probability[11] of the number of values of p to be n, S_p is the set of sources providing claims on p and $\tau(s)$ is the reliability of s measured by each method. Suppose the predicted number of true values is N, the improved method will output the values with the top-N highest confidence scores as the identified true values. We renamed the five improved methods as *Voting**[12], *Sums**, *Average-Log**, *TruthFinder** and *2-Estimates**, for short.

To ensure fair comparison, we used the same stop criterion for iterative methods and ran a series of experiments to determine the optimal parameter settings for each baseline method.

For our approach, we set $\theta = 0.6$. All algorithms were implemented in Python 3.4.0. We conducted experiments on a 64-bit Windows 10 PC with an octa-core 3.4GHz CPU and 16GB RAM. We ran each method ten times and used four evaluation metrics, including *precision*, *recall*, F_1 *score*[13] and *execution time*, to evaluate the average performance of each method.

4.2.2 Comparison studies. Table V shows the performance of different approaches on the two data sets in terms of accuracy and efficiency (i.e. precision, recall, F_1 score and execution time). The results show that our approach consistently performed well: it achieved the best recall and F_1 score among the methods. When compared with the two existing MTD methods (LTM and MBM), our approach required the lowest execution time. This is because LTM conducted complicated Bayesian inference over a probabilistic graphical model, and MBM included time-consuming copy detection. All the algorithms achieved lower precision on the Book–Author data set. The possible reasons include the small scale of this data set, missing values (i.e. true values that are missed by all the data sources) and poor quality of sources, leading to insufficient evidence to support obtaining all correct values. The majority of methods showed higher precision than recall, reflecting relatively high positive precision than negative precision of most real-world sources.

Specifically, Voting achieved relatively lower recall on both data sets among the five STD methods, i.e. the second worst on Book–Author data set and the worst on Parent–Children data set. This is because Voting ignores the differences of source quality and simply determines the truth of data by tuning the predefined threshold. To obtain the nearly perfect precision, the threshold of Voting is set as a high value bigger than 0.5. This result implies that instead of applying for solving multi-valued data fusion problem, Voting can be better used for generating the ground truth for semi-supervised truth-finding approaches. The improved STD methods commonly performed worse than their original versions with lower precision and recall, implying that most real-world sources tend to be cautious, making the predicted number of true values for each predicate smaller than it should be. Besides our approach, 2-Estimates and MBM performed better than other methods. This can be attributed to their consideration of mutual exclusion. Though LTM also takes this implication into consideration, it makes strong assumptions on the prior distributions of latent variables. Once the data set does not comply with the assumed distributions, it is likely to perform poorly. Although our approach achieved no significantly superior precision, the recall was improved

	Book–Author data set				Parent–Children data set			
Method	Precision	Recall	F1 score	Time(s)	Precision	Recall	F1 score	Time(s)
Voting	*0.84*	0.63	0.72	*0.07*	*0.90*	0.74	0.81	*0.56*
Sums	*0.84*	0.64	0.73	0.85	*0.90*	0.88	0.89	1.13
Avg-Log	*0.83*	0.60	0.70	0.61	*0.90*	0.88	0.89	*0.75*
TruthFinder	*0.84*	0.60	0.70	0.74	*0.90*	0.88	0.89	1.24
2-Estimates	0.81	0.70	0.75	0.38	*0.91*	0.88	0.89	1.34
Voting*	0.77	0.42	0.54	0.13	0.87	0.74	0.80	0.89
Sums*	*0.83*	0.24	0.38	0.99	0.86	0.88	0.87	1.45
Avg-Log*	0.74	0.49	0.59	*0.08*	0.89	0.87	0.88	0.92
TruthFinder*	0.70	0.71	0.70	0.99	0.85	*0.91*	0.88	1.16
2-Estimates*	*0.83*	0.24	0.38	0.79	0.86	0.89	0.87	1.47
LTM	0.82	0.65	0.73	0.98	0.88	0.90	0.89	0.99
MBM	*0.83*	*0.74*	*0.78*	0.67	*0.91*	0.89	*0.90*	2.17
Our Approach	0.81	*0.77*	*0.79*	0.63	*0.90*	*0.92*	*0.91*	0.91

Note: The best and second-best performance values are in italic

Table V. Comparison of different methods

drastically, resulting in highest F_1 scores for both data sets. The results reveal that our approach achieves the best overall performance among all the baseline methods, which is consistent with our expectation, because it makes no prior assumption and considers the endorsement relations among sources by combining with the graph-based method.

We also investigated the performance of our approach by tuning the values of the source confidence factor θ from 0 to 1 on both data sets. Figure 2 shows the experimental results in terms of precision, recall and F_1 score on Book–Author data set. The experimental results on Parent–Children data set showed similar conclusions. When θ equaled to 0, indicating that the negative claims were not trusted at all, all the positive claims were labeled as true. In this case, the precision was undoubtedly very low (0.49), as there should be a large number of low-quality sources providing false values; meanwhile, the recall was not surprisingly high (0.84), as all claimed values were regarded as true. The recall was less than 1, implying that some true values were missing and not claimed by any sources. As θ grew, the precision dramatically increased (from 0.49 to 0.83) while the recall slightly decreased (from 0.84 to 0.67), implying that by putting more confidence on source negative precision, our approach would reject more false values than true values. The overall performance peaked at the point of $\theta = 0.6$ with an F_1 score of 0.79, which is consistent with our intuition that source confidence on positive claims should be more respected. For $\theta \in [0.3,0.9]$, the lowest F_1 score is 0.76, which is still higher than the baselines.

5. Related work

Recent years have seen the emergence and wide application of many large-scale KBs (Weikum and Theobald, 2010). The research efforts on KB construction can be generally divided into four groups. First, some existing KBs, including YAGO (Suchanek *et al.*, 2007) and DBpedia (Auer *et al.*, 2007), are constructed based on high-quality structured sources such as Wikipedia infoboxes. Second, some KBs are built by using open information extraction (Open IE) techniques and extracting data from the open Web. Such techniques, for example, Reverb (Fader *et al.*, 2011), OLLIE (Mausam *et al.*, 2012) and PRISMATIC (Fan *et al.*, 2010), can obtain lots of new facts and entities from the Web. However, they work only at the lexical level, thus usually result in redundant facts that are worded differently but have the same semantic. Third, some techniques, such as NELL/ReadTheWeb (Carlson *et al.*, 2010), DeepDive/Elementary (Niu *et al.*, 2012) and Knowledge Vault (Dong *et al.*, 2014a), construct KBs by using a fixed ontology to extract data from the open Web. Compared with Open IE techniques, they generate smaller amount but higher quality of entities from the Web. Fourth, different from general KBs with multiple types of predicates, there are also some methods, such as Probase (Wu *et al.*, 2012), which focus on constructing

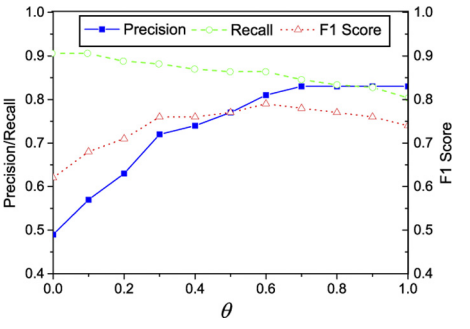

Figure 2.
Performance of our graph-based truth discovery approach under varying source confidence factor, i.e. θ

taxonomies (i.e. *isA* hierarchies). Despite the differences, a general KB construction approach follows two steps: *knowledge extraction* (discovering data sources, tapping unstructured data, connecting structured and unstructured data sources) and *truth discovery* (making sense of heterogeneous, dirty or uncertain data). To our knowledge, although KB construction has been studied for many years, this research area is still far from mature. Both knowledge extraction and knowledge fusion techniques need to be further improved. In the following subsections, we review the representative research efforts that are relevant to the two research areas: *knowledge extraction* and *Truth Discovery*.

5.1 Knowledge extraction

With the popularity of the open linked data, nowadays, a tremendous number of works have been inspired and contribute to extract Web data into semantic Web format (RDF triples), such as Virtuoso Sponger[14], Semantic Fire[15] and DeiXTo[16]. The knowledge extractors can be divided into four groups by the types of extracted knowledge:

(1) *Taxonomic knowledge extractors*, which search for individual entities and label them into semantic classes. These extractors can be further classified into two groups, namely, *Wikipedia-centric* (Suchanek *et al.*, 2007; Wu and Weld, 2008) and *Web-based* (Wu *et al.*, 2012) methods.

(2) *Factual knowledge extractors* aim at determining the truthfulness (i.e. truth/false) of a given piece of information from the Web (Dong *et al.*, 2014a; Carlson *et al.*, 2010).

(3) *Emerging knowledge extractors* typically use Open IE techniques to discover new relationships and new entities from the Web (Nakashole *et al.*, 2012; Etzioni *et al.*, 2011).

(4) *Temporal knowledge extractors* focus on identifying the facts on given predicates at different time points (Alonso *et al.*, 2009).

For predicate extraction, to the best of our knowledge, rare works focus on extracting predicates from multiple types of sources. Pasca and Durme (2007) discovery that the predicates extracted from query stream are of 45 per cent higher accuracy than those from Web texts by adopting a head-to-head qualitative comparison. They further conduct in-depth predicate extraction from both query logs and query sessions by Pasca *et al.* (2010) to prove this point. Kopliku *et al.* (2011) combine extractions from structured data sources including Web tables, search hit counts, Wikipedia and DBpedia. Lee *et al.* (2013) extract predicates from query logs, Web documents and external KBs independently to compute the typicality for a class (resp. attribute), given a specific attribute (resp. class). The most relevant work is proposed by Gupta *et al.* (2014). They constructed a novel ontology named Biperpedia. However, our work is different from Biperpedia in three aspects. First, we combine the predicates extracted from two existing KBs (Freebase and DBpedia) instead of a single KB. Second, we define filters and more practical patterns for better query stream extraction. Third, while Biperpedia regards Web tables as meaningless, the value of Web tables for predicate extraction has been proved by many works (Kopliku *et al.*, 2011). Our approach additionally extracts predicates from DOM trees, of which Web tables are only regarded as a sub-type.

5.2 Truth discovery

Since Yin *et al.* (2008) first formulated the truth discovery problem in 2008, considerable research efforts have been conducted for truth discovery by incorporating various implications of multi-sourced data in various application scenarios [(Li *et al.*, 2012; Waguih and Berti-Equille, 2014; Li *et al.*, 2016) for surveys]. The existing approaches can be roughly divided into five groups:

(1) *Web link*-based methods (Pasternack and Roth, 2010; Kleinberg, 1999; Pasternack and Roth, 2010, 2011; Yin and Tan, 2011) typically construct a bipartite graph between sources and values of predicates, and apply PageRank to compute source reliability and estimate value veracity.

(2) *Iterative* methods (Yin *et al.*, 2008; Pasternack and Roth, 2010; Galland *et al.*, 2010) compute value veracity and source reliability from each other in an iterative manner.

(3) *Bayesian point estimation* methods (Dong *et al.*, 2009, 2012; Wang *et al.*, 2015) use *Bayesian analysis* to calculate the maximum posteriori probability for each predicate.

(4) *Probabilistic graphical model*-based methods (Zhao and Han, 2012; Pasternack and Roth, 2013; Zhao *et al.*, 2012) adopt probabilistic graphical models to reason about the truth of each predicate of interest.

(5) *Optimization*-based methods (Li *et al.*, 2014b; Wang *et al.*, 2012; Li *et al.*, 2014a) consider the truth discovery problem as an optimization problem.

Most of the existing truth discovery methods make the single-valued assumption. To the best of our knowledge, only two related methods take multi-valued predicates into account. The first solution, LTM (Latent Truth Model) (Zhao *et al.*, 2012), applies a probabilistic graphical model. However, Waguih and Berti-Equille (2014) conclude with extensive experiments that this model impairs the scalability of the approach. In addition, LTM makes strong assumptions about prior distributions for nine latent variables, which impairs the flexibility of the approach. To relax unnecessary assumptions, Wang *et al.* (2015) analyze the unique features of MTD and propose an MBM. Different from the above two methods, our graph-based approach requires no initialization of source reliability and makes no prior distribution assumptions. Thus, it is more robust and insensitive to various problems scenarios and parameter settings.

6. Future research directions

There are many opportunities to extend this work for full-fledged KB construction. In this section, we lay out a research agenda by proposing several future research directions.

6.1 Quantifying extraction uncertainty

While many extractors have been proposed, rare research efforts have been devoted to investigating the uncertainty of extractions. Few knowledge extraction techniques simultaneously assign confidence scores to their extractions (Dong *et al.*, 2014a; Wick *et al.*, 2013a), and consequently, these scores are rarely leveraged to improve the quality of extractions. Moreover, the criterion of confidence assignment in different extractor is varied from one another, making the confidence scores incomparable and tricky to be used. In our future work, we plan to assign a confidence score to each triple extracted by our extractors by following a unified criterion and incorporate those scores into our graph-based truth discovery model for better value veracity estimation.

6.2 Considering noises introduced by extractors

Existing truth discovery methods refer to the real-world sources, e.g. websites, as the provenance of data. However, the data sets, on which the existing approaches conduct truth discovery, are extracted from the real-world sources by various extractors with different capabilities. The issue is that not only are the real-world sources error-prone but also the extractors may introduce additional errors into the data sets, including predicates linkage errors, triple identification errors and entity linkage errors. Ignoring the noises introduced by extractors would impair the accuracy of truth discovery. By additionally considering

extractors as one of the provenances of data, a more challenging problem, *knowledge fusion*, should be considered. Dong *et al.* (2014b) recently investigated data fusion techniques and found that some of the techniques are still promising in solving the knowledge fusion problem. However, these methods are all under the single-valued assumption. We will further incorporate the multi-valued knowledge fusion approach into our system.

6.3 Detecting inter-source and inter-extractor relations

There are complex relations among real-world sources, for example, one source may directly or transitively copy the other sources and several sources may copy data from one authoritative source. The relations among extractors can be even richer. There may be correlations among extractors, if they focus on the same types of Web content or apply the same extraction techniques. On the other hand, there may also be anti-correlations among extractors if they apply significantly different extraction techniques. Taking these relations into consideration may lead to better fusion results.

6.4 Considering hierarchical value spaces

Previous research efforts (Li *et al.*, 2012) have proposed improving the accuracy of truth discovery by considering value similarity. However, they all focus on the similarity of string or numeric values. To the best of our knowledge, there is no existing work that considers value hierarchy. For example, for "the hometown of a person", "Wuhan" and "Hubei" can both be the true values (Wuhan is the capital city of Hubei province). In the future, we will propose a strategy that can infer the hierarchy and similarity of the values of predicates, where the information is presented by our extracted triples.

7. Conclusion

In this paper, we focus on the problem of generating actionable knowledge from Big Data. As the existing KBs are still far from complete and accurate, we propose a system, which consists of two phases, namely, *knowledge extraction* and *truth discovery*, to construct a broader KB, called *GrandBase*. In particular, for knowledge extraction, we propose an approach for new predicate extraction to augment the ontology of Freebase: we first extract high-quality predicates from the existing KBs (i.e. DBpedia and Freebase) and query stream, and then apply these extractions as seeds to induce extractions from the open Web (Web texts and DOM trees), whereas the new entity extraction and corresponding fact extraction are left for our future work. For truth discovery, we propose a graph-based approach as the solution of the multi-valued truth discovery problem. In particular, two graphs are constructed by quantifying the two-sided inter-source agreements, from which two-sided source vote counts are derived to estimate value veracity for each predicate. We conduct experimental studies to show the effectiveness of our approaches and analyze the future research directions regarding GrandBase construction and extension.

Notes

1. www.bing.com/blogs/site_blogs/b/search/archive/2013/03/21/satorii.aspx
2. www.google.com/insidesearch/features/search/knowledge.html
3. www.insidefacebook.com/2013/01/14/facebook-builds-knowledge-graph-with-info-modules-on-community-pages/
4. New entity extraction and triple identification will be the focuses of our future work.

5. By similar types, we mean the types with synonymous names of the class, or the types that have high overlaps (e.g., more than 50 per cent) with the class in their covered entities.
6. www.w3.org/DOM
7. https://hama.apache.org/
8. https://code.google.com/p/hypertable/downloads/detail?name=query-log.tsv.gz
9. www.cim.mcgill.ca/dudek/206/Logs/AOL-user-ct-collection/
10. http://code.google.com/p/crawler4j/
11. Such values are then normalized to represent probabilities.
12. For Voting*, we predict the number of true values as the number with the highest vote counts.
13. F_1 score is an overall metric as neither precision nor recall could represent the method accuracy independently.
14. http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/VirtSponger/
15. https://code.google.com/p/semantic-fire/
16. http://deixto.com/

References

Alonso, O., Gertz, M. and Baeza-Yates, R.A. (2009), "Clustering and exploring search results using timeline constructions", *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM'09)*, *New York, NY*.

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R. and Ives, Z. (2007), "DBpedia: a nucleus for a web of open data", *The Semantic Web*, Vol. 4825, Springer, Berlin, Heidelberg, pp. 722-735.

Bing, L., Lam, W. and Yuan, G. (2011), "Towards a unified solution: data record region detection and segmentation", Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM 2011), *New York, NY*.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T. and Taylor, J. (2008), "Freebase: a collaboratively created graph database for structuring human knowledge", *SIGMOD*, ACM, Vancouver, BC, pp. 1247-1250.

Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka, E.R. Jr. and Mitchell, T. (2010), "Toward an architecture for never-ending language learning", *AAAI*, Atlanta.

Dong, X.L., Berti-Equille, L. and Srivastava, D. (2009), "Integrating conflicting data: the role of source dependence", *Proceedings of the VLDB Endowment*, Vol. 2 No. 1, pp. 550-561.

Dong, X.L., Saha, B. and Srivastava, D. (2012), "Less is more: selecting sources wisely for integration", *Proceedings of the VLDB Endowment*, Vol. 6 No. 2, pp. 37-48.

Dong, X.L., Gabrilovich, E., Heitz, G., Horn, W., Murphy, K., Sun, S. and Zhang, W. (2014b), "From data fusion to knowledge fusion", Proceedings of the VLDB Endowment, Vol. 7 No. 10, pp. 881-892.

Dong, X.L., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S. and Zhang, W. (2014a), "Knowledge vault: a web-scale approach to probabilistic knowledge fusion", Proceedings ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, *New York, NY*, pp. 601-610.

Etzioni, O., Fader, A., Christensen, J., Soderland, S. and Mausam, M. (2011), "Open information extraction: the second generation", Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI'11), *Barcelona*.

Fader, A., Soderland, S. and Etzioni, O. (2011), "Identifying relations for open information extraction", *EMNLP*, Edinburgh.

Fan, J., Ferrucci, D., Gondek, D. and Kalyanpur, A. (2010), "Prismatic: inducing knowledge from a large scale lexicalized relation resource", *First International Workshop on Formalisms and Methodology for Learning by Reading*, Association for Computational Linguistics, pp. 122-127.

Fang, X.S. (2015), "Generating actionable knowledge from big data", *Proceedings the 2015 SIGMOD PhD Symposium (SIGMOD)*, *Melbourne*, pp. 3-8.

Fang, X.S., Wang, X. and Sheng, Q.Z. (2015), "Ontology augmentation via attribute extraction from multiple types of sources", Proceedings the 26th Australasian Database Conference (ADC), *Springer*, *Cham*, pp. 16-27.

Galland, A., Abiteboul, S., Marian, A. and Senellart, P. (2010), "Corroborating information from disagreeing views", Proceedings ACM International Conference on Web Search and Data Mining (WSDM), *New York, NY*, pp. 131-140.

Gleich, D.F., Constantine, P.G., Flaxman, A.D. and Gunawardana, A. (2010), "Tracking the random surfer: empirically measured teleportation parameters in PageRank", Proceedings International World Wide Web Conference (WWW), *Raleigh, NC*, pp. 381-390.

Grishman, R. (2012), "Information extraction: capabilities and challenges", *Notes for the 2012 International Winter School in Language and Speech Technologies*, Rovira i Virgili University, Tarragona.

Gupta, R., Halevy, A., Wang, X., Whang, S. and Wu, F. (2014), "Biperpedia: an ontology for search applications", *The VLDB Endowment (PVLDB)*, Vol. 7 No. 7, pp. 505-516.

Kleinberg, J.M. (1999), "Authoritative sources in a hyperlinked environment", *Journal of the ACM*, Vol. 46 No. 5, pp. 604-632.

Kopliku, A., Boughanem, M. and Pinel-Sauvagnat, K. (2011), "Towards a framework for attribute retrieval", *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM'11)*, *New York, NY*.

Lee, T., Wang, Z., Wang, H. and Hwang, S.W. (2013), "Attribute extraction and scoring: a probabilistic approach", Proceedings of 29th International Conference on Data Engineering (ICDE'13), *Brisbane*.

Li, X., Dong, X.L., Lyons, K., Meng, W. and Srivastava, D. (2012), "Truth finding on the deep web: is the problem solved?", *Proceedings the VLDB Endowment*, Vol. 6 No. 2, pp. 97-108.

Li, Q., Li, Y., Gao, J., Zhao, B., Fan, W. and Han, J. (2014b), "Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation", Proceedings ACM SIGMOD International Conference on Management of Data, *Snowbird, UT*, pp. 1187-1198.

Li, Q., Li, Y., Gao, J., Su, L., Zhao, B., Demirbas, M., Fan, W. and Han, J. (2014a), "A confidence-aware approach for truth discovery on long-tail data", *Proceedings of the VLDB Endowment*, Vol. 8 No. 4, pp. 425-436.

Li, Y., Gao, J., Meng, C., Li, Q., Su, L., Zhao, B., Fan, W. and Han, J. (2016), "A survey on truth discovery", *ACM SIGKDD Explorations Newsletter*, Vol. 17 No. 2, pp. 1-16.

Liu, B., Grossman, R. and Zhai, Y. (2003), "Mining data records in web pages", Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03), *New York, NY*.

Mausam, M., Schmitz, M., Bart, R., Soderland, S. and Etzioni, O. (2012), "Open language learning for information extraction", *EMNLP*, Jeju Island.

Nakashole, N., Weikum, G. and Suchanek, F. (2012), "Discovering and exploring relations on the web", *The VLDB Endowment (PVLDB)*, Vol. 5 No. 12, pp. 1982-1985.

Niu, F., Zhang, C., Re, C. and Shavlik, J.W. (2012), "DeepDive: web-scale knowledge-base construction using statistical learning and inference", *VLDS*, Istanbul.

Pasca, M. and Durme, B.V. (2007), "What you seek is what you get: extraction of class attributes from query logs", Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07), *Hyderabad*.

Pasca, M., Alfonseca, E., Robledo-Arnuncio, E., Martin-Brualla, R. and Hall, K. (2010), "The role of query sessions in extracting instance attributes from web search queries", Proceedings of the 32th European Conference on Information Retrieval (ECIR'10), *Milton Keynes*.

Pasternack, J. and Roth, D. (2010), "Knowing what to believe (when you already know something)", Proceedings International Conference on Computational Linguistics (COLING), *Beijing*, pp. 877-885.

Pasternack, J. and Roth, D. (2011), "Making better informed trust decisions with generalized fact-finding", *Proceedings International Joint Conferences on Artificial Intelligence (IJCAI)*, pp. 2324-2329.

Pasternack, J. and Roth, D. (2013), "Latent credibility analysis", *Proceedings International World Wide Web Conference (WWW)*, *Rio de Janeiro*, pp. 1009-1020.

Suchanek, F., Kasneci, G. and Weikum, G. (2007), "YAGO: a core of semantic knowledge", *WWW*, Banff.

Waguih, D.A. and Berti-Equille, L. (2014), "Truth discovery algorithms: an experimental evaluation", *arXiv preprint arXiv:1409.6428*.

Wang, D., Kaplan, L., Le, H. and Abdelzaher, T. (2012), "On truth discovery in social sensing: a maximum likelihood estimation approach", *Proceedings ACM International Conference on Information Processing in Sensor Networks (Sensys)*, *Beijing*, pp. 233-244.

Wang, X., Sheng, Q.Z., Fang, X.S., Yao, L., Xu, X. and Li, X. (2015), "An integrated Bayesian approach for effective multi-truth discovery", *Proceedings the 24th ACM International Conference on Information and Knowledge Management (CIKM)*, pp. 493-502.

Weikum, G. and Theobald, M. (2010), "From information to knowledge: harvesting entities and relationships from web sources", Proceedings of the Twenty-Ninth ACM Sigmod-Sigact-Sigart Symposium on Principles of Database Systems, *ACM*, *Indianapolis*, pp. 65-76.

Wick, M., Singh, S., Kobren, A. and McCallum, A. (2013a), "Assessing confidence of knowledge base content with an experimental study in entity resolution", AKBC workshop.

Wick, M., Singh, S., Pandya, H. and McCallum, A. (2013b), "A joint model for discovering and linking entities", *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction (AKBC'13)*, *New York, NY*.

Wu, F. and Weld, D. (2008), "Automatically refining the Wikipedia infobox ontology", *Proceedings of the 17th International Conference on World Wide Web (WWW'08)*, *New York, NY*.

Wu, W., Wang, H., Li, H. and Zhu, K.Q. (2012), "Probase: a probabilistic taxonomy for text understanding", *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data (SIGMOD'12)*, *Madison, WI*.

Yin, X., Han, J. and Yu, P.S. (2008), "Truth discovery with multiple conflicting information providers on the web", *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, Vol. 20 No. 6, pp. 796-808.

Yin, X. and Tan, W. (2011), "Semi-supervised truth discovery", *Proceedings International World Wide Web Conference (WWW)*, *Hyderabad*, pp. 217-226.

Zhao, B. and Han, J. (2012), "A probabilistic model for estimating real-valued truth from conflicting sources", Proceedings International Workshop on Quality in DataBases (QDB), *Coheld with VLDB*.

Zhao, B., Rubinstein, B.I.P., Gemmell, J. and Han, J. (2012), "A Bayesian approach to discovering truth from conflicting sources for data integration", *Proceedings the VLDB Endowment*, Vol. 5 No. 6, pp. 550-561.