Generating Actionable Knowledge from Big Data

Xiu Susie Fang
School of Computer Science
The University of Adelaide, Australia
Adelaide, SA 5005, Australia
xiu.fang@adelaide. edu.au

Supervised by A/Prof. Michael Sheng
Expected Graduate Date: September 2017

ABSTRACT

The last few years have seen a rapid increase of sheer amount of data produced and communicated over the Internet and the Web. While it is widely believed that the availability of such "Big Data" holds the potential to revolutionize many aspects of our modern society (e.g., intelligent transportation, environmental monitoring, and energy saving), many challenges need to be addressed before this potential can be realized. This PhD project focuses on one critical challenge, namely *extracting actionable knowledge from Big Data*. Tremendous efforts have been contributed on mining large-scale data on the Web and constructing comprehensive knowledge bases (KBs). However, existing *knowledge extraction* systems retrieve data from limited types of Web sources. In addition, *data fusion* approaches consider very little of the noises produced by those knowledge extraction systems. Consequently, the constructed KBs are far from being comprehensive and accurate. In this paper, we present our initial design of a framework for extracting machine-readable data with high precision and recall from four types of data sources, namely Web texts, Document Object Model (DOM) trees, existing KBs, and query stream. Confidence scores are attached to the resulting knowledge, which can be used to further improve the knowledge fusion results.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications

Keywords

Knowledge Base, Knowledge Fusion, DOM Tree

1. INTRODUCTION

The availability of large amounts of data has soared dramatically in the last several years. According to IBM[1], 2.5 quintillion bytes of data are created every day and 90% of data in the world has been created in the past two years. Indeed, the 5-V features (volume, velocity, variety, veracity, and value) of big data have become a major concern of data integration research. Although various tools and techniques have been developed to combine data from different sources and to support unified representation of these data, data integration still faces three challenges, i.e., value heterogeneity, instance heterogeneity, and structure heterogeneity of data. Traditional data integration approaches resolve these challenges by a four-step process, namely *source selection* [14], *schema alignment* [4], *record linkage* [16], and *data fusion* [11].

As the scale of data increases unprecedentedly, it becomes more urgent than ever to exploit the full values of these data by extracting richer knowledge from the data. In response, many large-scale knowledge bases (KBs), such as DBpedia[2], Freebase[3], and YAGO[4], have been constructed for both human use and feeding knowledge-driven applications. Most of these KBs represent their data by Resource Description Framework (RDF) triples, which we call *actionable knowledge*. Since KBs are built by extracting and fusing data from the open Web [13], KB construction faces more challenges than traditional data integration. Firstly, the scalability of KB construction techniques becomes more critical. Secondly, the extraction methods may provide additional noises, such as attribute linkage errors, triple identification errors, and entity linkage errors, into the system. Thirdly, the provenance of data in knowledge fusion is more complicated than data fusion. Although some countermeasures exist, they are still far from satisfaction. In this PhD project, we aim at investigating novel ways to effectively and efficiently generate actionable knowledge from Big Data. Specifically, we focus on two key steps of KB construction, namely *knowledge extraction* and *knowledge fusion*.

Knowledge extraction techniques (i.e., *extractors*) aim at obtaining machine-readable and interpretable knowledge from structured (e.g., relational databases), semi-structured (e.g., Extensible Markup Language (XML)) and/or unstructured sources (e.g., texts, documents, images). We identify the following three limitations of the existing techniques.

- Most existing KBs such as YAGO, NELL[5], and DeepDive[6], are constructed by applying extractors that fo-

[1]http://www-01.ibm.com/software/data/bigdata/

SIGMOD'15 PhD Symposium, May 31 - June 04 2015, Melbourne, VIC, Australia.

http://dx.doi.org/10.1145/2744680.2744687.

[2]http://dbpedia.org/About
[3]https://www.freebase.com/
[4]http://datahub.io/dataset/yago
[5]http://rtw.ml.cmu.edu/rtw
[6]http://deepdive.standord.edu/

cus on extracting knowledge from a single kind of data sources (e.g., Web texts). In particular, these KBs simply remove the tags and extract data from plain texts, and ignore the knowledge contained in the DOM tree structures formed by these tags. As a result, these KBs fail to exploit the full knowledge contained in the data sources, leading to the limited coverage and quality of the extractions. In fact, various types of data sources, such as DOM trees, HTML tables, and human annotated pages [12], can be used for more accurate and complete knowledge extraction.

- The uncertainties of extractions are seldom investigated. Although some extractors assign confidence scores to their extractions to bridge this gap, these scores are rarely leveraged to improve the extraction quality. In addition, the criterion of confidence assignments for different extractors remains undefined.

- Most previous work focuses on extracting facts of entities in a *predefined ontology*, which limits the coverage of the extractions. Although several approaches, such as open information extraction (Open IE) [15], manage to add new entities and relations to the extractions, they fail to distinguish synonyms, therefore introducing additional redundancy to the results.

Traditional *data fusion* methods [7, 20] aim at resolving the conflicts among multi-sourced data by discovering the true values of each data item (e.g., the profession of Barack Obama). Some methods consider additional factors, such as the accuracies of and the correlations among Web sources, to improve the fusion quality. These approaches can only achieve limited precision because they ignore the noises introduced by the extractors. In this regard, *Knowledge fusion* has been proposed to take the qualities of extractions explored by extractors into account. However, this is merely a starting point and few of the open problems have been solved. For example, very few works have considered the functionality degree of attributes. All existing KBs ignore the fact that values can be hierarchically structured. For example, *South Australia-Australia-Adelaide* forms a chain in the location hierarchy. Because of such value hierarchy, even for data items with functional attributes, there can be multiple truths (e.g., the triples (*Susie Fang, birth place, China*) and (*Susie Fang, birth place, Wuhan*) can both be true). They simply consider the values represented at multiple levels of abstraction as conflicting values. Moreover, the correlations among sources, as well as among extractors have been rarely explored.

This PhD project is at the end of its first year. We have extensively reviewed the literature of related research areas. We also propose an initial design of a framework for extracting actionable knowledge with high precision and recall from four different types of data sources, namely Web texts, DOM trees, existing KBs, and query stream. We employ an open IE (information extraction) approach to extract new knowledge from the open Web, and use a unified criterion to assign confidence scores to the resulting triples. The scores will be used in the subsequent knowledge fusion tasks to further improve the fusion results.

The remainder of this paper is organized as follows. Section 2 provides an overview of the related work. Section 3 introduces our design of the overall framework. We report our current progress in Section 4. Section 5 provides some concluding remarks.

2. RELATED WORK

In this section, we overview the representative research efforts that are relevant to this PhD project, on two main research areas: *knowledge extraction* and *knowledge fusion*.

2.1 Knowledge Extraction

We group existing knowledge extraction techniques into four groups by the types of extracted knowledge, to be detailed in the following. It is worthwhile to note that due to the popularity of the open linked data, many research efforts focus on extracting Web data into RDF triples.

The *Taxonomic Knowledge Extractors* search for individual entities and organize them into semantic classes. These extractors can be further classified into two groups: *Wikipedia-centric* and *Web-based* methods. Wikipedia-centric methods include the methods proposed in [27, 26], which link Wikipedia categories to WordNet, and the Kylin Ontology generator designed by Wu et al. [30], which learns more mappings by applying advanced machine learning techniques such as Support Vector Machine (SVMs), Markov Logic Network (MLNs). Web-based methods include Probase proposed by Wu et al. [31], which constructs a taxonomy from the Web. Unfortunately, the coverage and the qualities of the extractions from these extractors are generally limited because they only focus on specific types of data sources.

The *Factual Knowledge Extractors* focus on determining the truthfulness (i.e., truth/false) of a given piece of information from the Web. Many methods have been proposed for this purpose (e.g., the works in [12, 8]), including Regex-based extraction, pattern-based harvesting, consistency reasoning, probabilistic methods and Web-table methods. However, these extractors are not robust with both high precision and recall, and are not scalable, which need to be further improved. The *Emerging Knowledge Extractors* typically use open information (schema-less) extraction techniques (Open IE) (e.g., the works in [23, 15]) to discover new relationships and new entities from the Web. Instead of using a fixed ontology, such methods can enhance the ontology. However, these methods work at the lexical level, which usually result in redundant facts that are denoted by different words but indicate the same semantic meaning. Finally, the *Temporal Knowledge Extractors* identify the facts on given relations at different time points (e.g., the works in [2, 5]). As the temporal knowledge additionally require extracting the valid time points of facts, the solutions are more complex.

We also overview the related work on attribute extraction, particularly the approaches on DOM trees. Extracting attributes from DOM trees is not completely new. Early supervised approaches [1, 22] use manually defined wrappers to extract attributes from each Website, which are time-consuming and non-scalable. Wrapper learning techniques (e.g., [28] proposed by Turmo et al.) can help reduce human intervention, but additionally requires labeled data for the training, and are inapplicable to new websites that have not been handled before. Generative models designed in [33] alleviate this problem by segmenting and labeling the training samples. However, they can only extract the attributes that are predefined in the training data. Interactive learning techniques developed by Irmak

et al. [18] and Kristjansson et al. [19] can also help reduce human efforts on preparing the training data. They are unfortunately not automated. Unsupervised methods include *template-based* methods and *pattern-based* methods. The template-based methods, represented by RoadRunner (designed by Crescenzi et al. [9]) and EXALG (developed by Arasu et al. [3]), detect website-specific templates to extract attribute values. The pattern-based methods proposed by Liu et al. [21] and Bing et al. [6] extract data records from a single list page, based on some patterns that repeatedly occur in multiple data records. Both methods however require some re-implementation for new websites. Comparing with previous works, our approach enables more accurate and extensive attribute extraction from DOM trees, which is achieved automatically.

2.2 Knowledge Fusion

Comparing to data fusion, knowledge fusion is newer yet more challenging. Very few works have been conducted in this direction. Dong et al. [13] investigate data fusion techniques and find that some of them are still promising in solving the knowledge fusion problem. The authors adapt three existing data fusion techniques, namely VOTE, ACCU and POPACCU, and scale them up by using a MapReduce based framework for knowledge fusion. They also improve these techniques, by exploiting a number of techniques such as using provenances with finer-granularity, making wise selection of provenances, and making use of the gold standard to calculate more accurate initial quality values of the data sources, rather than simply setting some default values. However, these methods assume that every data item has a single true value, which fails to reflect the real world.

Many existing KBs, such as YAGO, NELL, and Knowledge Vault, apply (semi-)supervised methods to improve the qualities of the extractions. However, all these methods reply on the availability of training data, which limits their applicability. Pochampally et al. [25] propose a *relation-based* method, which takes the noises introduced by extractors into consideration. However, the approach refers to the extractors as data sources, only considers the correlations among extractors and ignores the correlations among original data sources.

3. AN OVERVIEW OF THE APPROACH

Figure 1 shows our initial design of a framework for extracting and fusing actionable knowledge from Big Data. There are two main phases in our approach, including the *knowledge extraction* phase and the *knowledge fusion* phase.

At the knowledge extraction phase, we proposed to apply open IE approach to extract RDF triples from four types of sources, namely query stream, existing KBs, Web texts and DOM trees. In particular, we use query stream as well as two major KBs, DBpedia and Freebase, to seed the attribute extraction from Web texts and DOM trees. In this way, we can generate RDF triples (including new entities and relationships) with high coverage and minimal redundancy.

At the knowledge fusion phase, we consider both functional and non-functional attributes and resolve the hierarchical value spaces of data to fuse the extractions produced by four different extractors. The misspellings, synonyms, and sub-attributes are identified at this stage and the relation-based knowledge fusion method is improved by considering the complex correlations among sources, as well

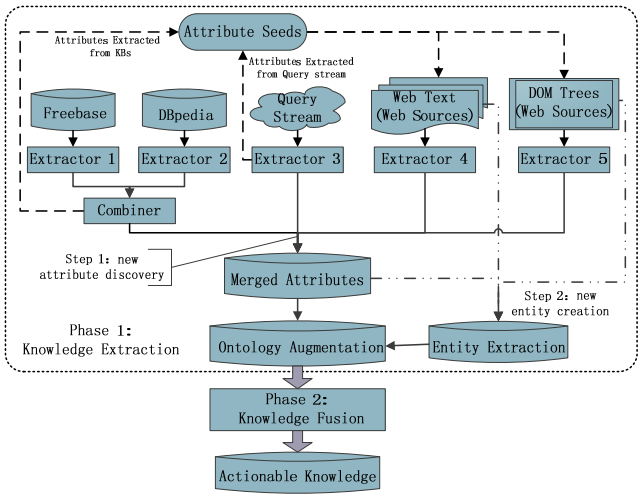

Figure 1: The architecture of our framework for KB construction

as extractors. The fusion results are then attached to Freebase for KB augmentation. In the rest of this section, we will discuss more on the two phases.

3.1 Knowledge Extractors

Our framework extracts knowledge in two steps: *new attribute discovery* and *new entity creation*.

Since query stream and existing KBs are generally more accurate, we propose to first extract attributes from these two types of sources. We then use the extractions as the seed to learn extraction patterns of Web texts and DOM trees. The learned patterns in turn are used to extract new attributes from the open Web.

Due to the different nature of the Web texts (often described by natural languages) and the DOM trees (semi-structured data described by tags), we apply different techniques for knowledge extraction. For Web texts, we learn regular lexical and parse patterns (which are unified syntax rules over the Web) from sentences and adopt these patterns directly to conduct knowledge extraction. Since Web sites may differ in their display styles and formats, the *tag path patterns* extracted from one Web page can hardly be applied to another Web page, even when they belong to the same Web site. For this reason, we learn tag path patterns for each Web page, and apply these patterns to extract new attributes from the Web pages. Based on the discovered new attributes, we create new entities automatically by improving the existing techniques [29]. Specifically, we propose to solve entity-linking and entity-discovery jointly. To improve the scalability of the solution, based on the more comprehensive attribute set, we will develop a novel model that reasons over the compact hierarchical entity representations, as well as a new distributed inference architecture, which is inherent in the MapReduce architectures, that avoids the synchronicity bottleneck. We will apply this enhanced ontology to explore more facts from the open Web environment. To deal with the uncertainty of the extracted triples and further support high-quality knowledge fusion, we also propose to assign a confidence score to each triple based on an unified criterion.

3.2 Knowledge Fusion

We focus on developing unsupervised techniques for knowledge fusion due to the diverse and dynamic nature of the Web. Our proposed solution on knowledge fusion features the following important aspects:

- *Handling functional and non-functional attributes.* Traditional data fusion methods have been proposed to solve the knowledge fusion problem [13]. However, they only tackle functional attributes. Zhao et al. [32] propose a graphical model that can predict truthfulness when there are multiple truths. The approach considers sensitivity (i.e., recall) and specificity of each source when deciding value correctness instead of reasoning about source accuracy. However, this approach aims at solving the data fusion problem, which is not scalable for knowledge fusion. We will consider a more scalable approach based on this technique for knowledge fusion and develop comprehensive solution to handle both functional and non-functional attributes.

- *Considering hierarchical value spaces.* Previous research efforts [11, 20, 24] have proposed to improve fusion quality by considering value similarity. However, they all focus on similarity of values for strings, numbers, etc. To the best of our knowledge, there is no existing work that considers value hierarchy. In this PhD project, we will propose a strategy that can reason about the hierarchy and similarity of the values of data items, where the information is presented by triples in the extracted knowledge. Thus, we can further improve the results of knowledge fusion.

- *Considering inter-Web sources and inter-extractors correlations.* Instead of simply considering extractors as sources [25], we will consider the correlations among Web sources and among extractors to improve fusion quality. We will investigate ways to improve the existing techniques by applying the Bayesian techniques [10].

- *Leveraging confidence scores.* The knowledge fusion technique can be further improved by leveraging the confidence scores calculated from the first phase of our system. Pasternack and Roth [24] propose an approach that leverages source-defined confidence scores to improve the Web-link based data fusion technique. In our project, we plan to follow this strategy to develop new unsupervised knowledge fusion techniques.

4. CURRENT PROGRESS

This PhD project is at the end of its first year. We have done an extensive literature review and also begun to tackle technical parts of the project. We have developed solutions for the attribute extraction in the knowledge extraction phase (see Figure 1). We briefly report our progress in the rest of this section.

While the ontologies of existing KBs already include a wide coverage of entities, the number of attributes contained in these KBs are relatively small (see Table 1 for some statistics we have done). For example, Freebase has 25 million entities, but only with 4,000 attributes. The type *University* in Freebase (note that in Freebase, classes are referred to as *types* and attributes are referred to as *properties*) only has 9 properties (see Table 2), while in reality we can easily identify many more attributes for the same class. The recently proposed ontology, Biperpedia [17], aims to discover more attributes from the Web. However, it mainly extracts attributes from Web texts and cannot handle the vastly available DOM trees on the Web.

Table 1: Statistics of Representative KBs

KB	# Entities(million)	# Attributes
YAGO	10	100
DBpedia	4	6,000
Freebase	25	4,000
NELL	0.3	500

To enable more complete and precise ontology augmentation, we propose to extract attributes from four types of sources, including query stream, Web texts, DOM trees and existing KBs (Freebase and DBpedia in our case). As attribute extraction techniques for Web texts have been widely studied, we focus on extracting attributes from the other three types of sources.

Attribute Extraction from Existing KBs. To the best of our knowledge, we are the very first few to combine existing KBs for knowledge extraction (we use Freebase and DBpedia). The attributes are first analyzed separately for both KBs and then we combine the attribute extractions from Freebase and DBpedia after some preprocessing (e.g., duplicate removal). Due to space constraint, we will not give the technical details. Table 2 shows the experimental results by applying our approach to five representative classes. It shows that our approach can increase the number of attributes effectively for all five classes in Freebase.

Table 2: Statistics of Five Representative Classes

Class	# Attributes				
	DBpedia	Extrac. (DBpedia)	Freebase	Extrac. (Freebase)	Combine (Freebase &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

Attribute Extraction from Query Stream. We propose an improved query stream extraction technique by using more patterns, such as "*what/how/when/who is the A of (the/a/an) E*", "*the A of (the/a/an) E*" and "*E's A*", and a set of filtering rules. These new patterns are used to extract more attributes, while the rules are used to exclude meaningless attributes to improve the quality of the extraction results.

To study and evaluate the capability of our approach, we conducted some preliminary experiments for the above five representative classes. We collected a query stream with 29,283,918 query records by combining two real-world datasets from Google[7] and AOL[8]. For entity recognition,

[7] https://code.google.com/p/hypertable/downloads/detail?name=query-log.tsv.gz

[8] http://www.cim.mcgill.ca/~dudek/206/Logs/AOL-user-ct-collection/

Table 3: Query Stream Extraction Results

Class	Relevant Query Records	Credible Attributes
Book	259,556	96
Film	403,672	59
Country	393,244	182
University	24,633	20
Hotel	15,544	N/A

each of these classes is specified as a set of representative entities of Freebase. The experimental results shown in Table 3 indicate that more relevant query records can lead to the extraction of more credible attributes. It is hard to find any attributes for the Hotel class.

Attribute Extraction from DOM Trees. Different from attribute extraction from Web texts where lexical and parse patterns can be learned from the Web, extracting attributes from DOM trees is more challenging due to different styles and formats of different Web sites. Usually, tag path patterns extracted from one Web page can hardly be applied to another page. To solve this problem, we develop an algorithm (see Algorithm 1).

Algorithm 1: Algorithm for DOM Tree Extraction

Input: *Type* T_k in Freebase; a set of Web sites regarding to T_k, S=$\{S_1, S_2, \ldots, S_n\}$, for each Web site $S_j \in S$, it contains a set of Web pages, P_j=$\{P_{j_1}, P_{j_2}, \ldots, P_{j_m}\}$,$j_m$ is the number of Web pages belong to S_j; the entity set Set_E of T_k in Freebase; the seed attribute set A_{T_k} extracted from query stream and existing KBs for T_k

Output: Original attributes for Type T_k in Freebase (i.e., enriched A_{T_k}).

Initialization: identify all the *entity node* and *non-entity node* in every Web pages, and obtain *tag path set*(denote as Tagpath) for each Web page, e.g., for $P_{j_l} \in P_j$, we keep a set of tag paths Tagpath(P_{j_l}).

for *each* $S_j \in S, j = 1, 2, \ldots, n$ **do**

  for *each* $P_{j_i} \in P_j, i = 1, 2, \ldots, j_m$, *and* P_{j_i} *contains at least an entity* $E \in Set_E$ *and an attribute* $A \in A_{T_k}$ **do**

    /* if $|A_{T_k}|$ is increased, the algorithm continues the loop for this Web site; else the algorithm begins to traverse another Web site */

    extract the tag path(s) between E and A, and transfer them to the *induced tag path pattern set*;

    compare all the other tag paths $\in$ Tagpath(P_{j_i}) with the induced tag path(s) in *induced tag path pattern set*;

    if *(a tag path is similar to the induced tag path(s))* **then**

      add the text of that *non-entity node* to A_{T_k};

      remove the tag path from $Tagpath(P_{j_i})$;

Briefly, given a type T, the algorithm first identifies the Websites related to T (e.g., http://www.imdb.com/ for type *Film*). For each Web page, the algorithm analyzes the DOM structure and classifies the text nodes into *entity node* (the texts represent the name of an entity E of T) and *non-entity node*. The tag paths between each *entity node* and their corresponding *non-entity node* are then extracted, removed of noisy tags, and kept in a *tag path set*. For each Website, the algorithm iteratively finds out Web pages that contain at least one (A, E) pair, where E is an *entity node*, A is the content of a *non-entity node* and $A \in$ *SEED_SET(T)* (the set of attribute seeds extracted from query stream and existing KBs). For each Web page, the algorithm traverses the *tag path set* for this Web page to obtain the tag paths between the seed A and E, and transfers these tag paths from the *tag path set* to an *induced tag path pattern set* for this Web page. We next compare all the tag paths in the *tag path set* with the patterns in the *induced tag path pattern set*. Those *non-entity nodes* with tag paths that are similar with the induced patterns are finally recognized as new attributes, and are added to *SEED_SET(T)*, with the corresponding tag paths removed from the *tag path set*.

The algorithm turns to another Website when the number of attributes in *SEED_SET(T)* reaches a certain threshold. Since the number of Web pages and text nodes in a Web page are limited, the algorithm can always terminate with an output.

5. CONCLUSIONS

The rapid increase of sheer amounts of data presents many challenges (e.g., effectively discovering actionable knowledge from Big Data). The main goal of this PhD project is to improve the knowledge base (KB) construction by generating more actionable knowledge from open Web data. This PhD project is currently at its end of the first year. We have already completed an extensive literature review and begun to propose technical solutions. In particular, we design a framework for the knowledge extraction and knowledge fusion. We propose to extract RDF triples with high precision and recall from four types of data sources. Preliminary experimental results show the capability of our approach to obtain additional new attributes with high quality. For the next stage, we will develop knowledge fusion techniques to tackle both functional and non-functional attributes. The confidence scores, the hierarchical value spaces of data, and correlations among sources and among extractors will also be used to improve the knowledge fusion results.

6. REFERENCES

[1] B. Adelberg. NoDoSE - A Tool for Semi-automatically Extracting Structured and Semistructured Data from Text Documents. *ACM SIGMOD Record*, 27(2):283–294, June 1998.

[2] O. Alonso, M. Gertz, and R. A. Baeza-Yates. Clustering and Exploring Search Results Using Timeline Constructions. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, (CIKM'09), New York, NY, USA, 2009.

[3] A. Arasu and H. Garcia-Molina. Extracting Structured Data from Web Pages. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, (SIGMOD'03), New York, NY, USA, 2003.

[4] Z. Bellahsene, A. Bonifati, and E. R. (Eds.). *Schema Matching and Mapping*. Springer, 2011.

[5] K. Berberich, S. Bedathur, O. Alonso, and G.Weikum. A Language Modeling Approach for Temporal Information Needs. In *Proceedings of the 32th European Conference on Advances in Information Retrieval*, (ECIR'10), Milton Keynes, UK, 2010.

[6] L. Bing, W. Lam, and Y. Gu. Towards a Unified Solution: Data Record Region Detection and Segmentation. In *Proceedings of the 20th ACM*

International Conference on Information and Knowledge Management, (CIKM'11), New York, NY, USA, 2011.

[7] J. Bleiholder, F. Naumann, and W. Y. Ma. Data Fusion. *ACM Computing Surveys*, 41(1):1–41, 2008.

[8] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. H. Jr., and T. Mitchell. Toward An Architecture for Never-ending Language Learning. In *Proceedings of the 24th Conference on Artificial Intelligence*, (AAAI'10), Atlanta, Georgia, USA, 2010.

[9] V. Crescenzi, G. Mecca, and P. Merialdo. RoadRunner: Automatic Data Extraction from Data-intensive Web Sites. In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, (SIGMOD'02), New York, NY, USA, 2002.

[10] X. L. Dong, L. Berti-Equille, Y. Hu, and D. Srivastava. Global Detection of Complex Copying Relationships Between Sources. *The VLDB Endowment (PVLDB)*, 3(1-2):1358–1369, 2010.

[11] X. L. Dong, L. Berti-Equille, and D. Srivastava. Integrating Conflicting Data: The Role of Source Dependence. *The VLDB Endowment (PVLDB)*, 2(1):550–561, 2009.

[12] X. L. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. Knowledge Vault: A Web-scale Approach to Probabilistic Knowledge Fusion. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (KDD'14), New York, NY, USA, 2014.

[13] X. L. Dong, E. Gabrilovich, G. Heitz, W. Horn, K. Murphy, S. Sun, and W. Zhang. From Data Fusion to Knowledge Fusion. In *Proceedings of the 40th International Conference on Very Large Data Bases*, (VLDB'14), Hangzhou, China, 2014.

[14] X. L. Dong, B. Saha, and D. Srivastava. Less is More: Selecting Sources Wisely for Integration. *The VLDB Endowment (PVLDB)*, 6(2):37–48, 2013.

[15] O. Etzioni, A. Fader, J. Christensen, S. Soderland, and Mausam. Open Information Extraction: The Second Generation. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, (IJCAI'11), 2011.

[16] L. Getoor and A. Machanavajjhala. Entity Resolution: Theory, Practice & Open Challenges. *The VLDB Endowment (PVLDB)*, 5(12):2018–2019, 2012.

[17] R. Gupta, A. Halevy, X. Wang, S. Whang, and F. Wu. Biperpedia: An Ontology for Search Applications. *The VLDB Endowment (PVLDB)*, 7(7):505–516, 2014.

[18] U. Irmak and T. Suel. Interactive Wrapper Generation with Minimal User Effort. In *Proceedings of the 15th International Conference on World Wide Web*, (WWW'06), New York, NY, USA, 2006.

[19] T. Kristjansson, A. Culotta, P. Viola, and A. McCallum. Interactive Information Extraction with Constrained Conditional Random Fields. In *Proceedings of the 19th National Conference on Artifical Intelligence*, (AAAI'04), San Jose, California, 2004.

[20] X. Li, X. L. Dong, K. B. Lyons, W. Meng, and D. Srivastavag. Truth Finding on the Deep Web: Is the Problem Solved? *The VLDB Endowment (PVLDB)*, 6(2):97–108, 2013.

[21] B. Liu, R. Grossman, and Y. Zhai. Mining Data Records in Web Pages. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (KDD'03), New York, NY, USA, 2003.

[22] L. Liu, C. Pu, and W. Han. XWRAP: an XML-enabled Wrapper Construction System for Web Information Sources. In *Proceedings of the 16th International Conference on Data Engineering*, (ICDE'00), San Diego, California, USA, 2000.

[23] N. Nakashole, G. Weikum, and F. Suchanek. Discovering and Exploring Relations on the Web. *The VLDB Endowment (PVLDB)*, 5(12):1982–1985, Aug 2012.

[24] J. Pasternack and D. Roth. Making Better Informed Trust Decisions with Generalized Fact-finding. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, (IJCAI'11), 2011.

[25] R. Pochampally, A. D. Sarma, X. L. Dong, A. Meliou, and D. Srivastava. Fusing Data with Correlations. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, (SIGMOD'14), Snowbird, Utah, USA.

[26] S. Ponzetto and M. Strube. Deriving a Large-Scale Taxonomy from Wikipedia. In *Proceedings of the 22th National Conference on Artificial Intelligence*, (AAAI'07), Vancouver, BC, Canada, 2007.

[27] F. Suchanek, G. Kasneci, and G. Weikum. YAGO: A Core of Semantic Knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, (WWW'07), New York, NY, USA, 2007.

[28] J. Turmo, A. Ageno, and N. Català. Adaptive Information Extraction. *ACM Computing Surveys (CSUR)*, 38(2):4–es, July 2006.

[29] M. Wick, S. Singh, H. Pandya, and A. McCallum. A Joint Model for Discovering and Linking Entities. In *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction*, (AKBC'13), New York, NY, USA, 2013.

[30] F. Wu and D. Weld. Automatically Refining the Wikipedia Infobox Ontology. In *Proceedings of the 17th International Conference on World Wide Web*, (WWW'08), New York, NY, USA, 2008.

[31] W. Wu, H. W. H. Li, and K. Q. Zhu. Probase: A Probabilistic Taxonomy for Text Understanding. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, (SIGMOD'12), Madison, WI, USA.

[32] B. Zhao, B. I. P. Rubinstein, J. Gemmell, and J. Han. A Bayesian Approach to Discovering Truth from Conflicting Sources for Data Integration. *The VLDB Endowment (PVLDB)*, 5(6):550–561, 2012.

[33] J. Zhu, Z. Nie, J.-R. Wen, B. Zhang, and W.-Y. Ma. Simultaneous Record Detection and Attribute Labeling in Web Data Extraction. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (KDD'06), New York, NY, USA, 2006.