International Journal of Integrated Engineering:
Special Issue 2018: Data Information Engineering, Vol. 10 No. 6 (2018) p. 176-182.
© Penerbit UTHM DOI: https://doi.org/10.30880/ijie.2018.10.06.025

Self-adaptive Based Model for Ambiguity Resolution of The Linked Data Query for Big Data Analytics

Nurfadhlina Mohd Sharef[1*], Yasser M. Shafazand[1], Mohd Zakree Ahmad Nazri[1], Nor Azura Husin[1]

[1]Intelligent Computing Research Group
Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, MALAYSIA.

Received 28 June 2018; accepted 5 August 2018, available online 24 August 2018

Abstract: Integration of heterogeneous data sources is a crucial step in big data analytics, although it creates ambiguity issues during mapping between the sources due to the variation in the query terms, data structure and granularity conflicts. However, there are limited researches on effective big data integration to address the ambiguity issue for big data analytics. This paper introduces a self-adaptive model for big data integration by exploiting the data structure during querying in order to mitigate and resolve ambiguities. An assessment of a preliminary work on the Geography and Quran dataset is reported to illustrate the feasibility of the proposed model that motivates future work such as solving complex query.

Keywords: Self-adaptive model, Ambiguity Resolution, Link Data Query, Big data, Heterogeneous Data Integration

1. Introduction

Big data analytics involves complex query which requires the fusion or integration of heterogeneous information which are typically of various types, domains, vocabulary, granularity and structure [1]. Intelligence through complex queries on Big Data involves semantic operations including (i) data integration, (ii) data ingestion in structured data (schema existed), unstructured data (through annotation and extraction), ontology (expression of data relations across schema) and semantic enrichment (expression of data relations within and across schema), (iii) data indexing and search, and (iv) data analytics.

The Linked Open Data (LOD) is an example of big data structure which makes integration of heterogeneous sources feasible because the data on the web is formatted in a machine readable way (i.e., Resource Description Framework (RDF) and ontology language (OWL)) with typed links between related entities [2-3]. An LOD infrastructure influences how query processing can be implemented on top of several characteristics namely central or distributed data storage, central or distributed data indexing, and independent or cooperative data sources.

However, answering complex queries that requires heterogeneous big data integration (BDI) is challenging because domain heterogeneity prevents users from manually translating queries due to their limited domain knowledge. Besides, users may not be aware that their query requires multiple sources aggregation, which results to multiple databases with potential duplicates, varying answers, and collection of different subsets of relevant data [4]. Therefore, an assistive query interface for big data is needed, which allows the users to input the query in natural language (NL) form [5]. We will refer to the assistive query interface for big data as natural language interface (NLI) henceforth. In order to compute the translation process, ambiguity issue occurs due to (i) variants in the user's terms in the query, (ii) degree of matching between user's terms with the data structure, and (iii) granularity within structure of the integrated data sources.

The existing disambiguation approaches for big data assistive query interface [6–10] have mainly exploited synonym based matching, reasoning through OWL and RDF's concepts linking mechanism (e.g., the same As function), similarity scoring functions, and depending on the user's intervention for the consolidation through clarification dialogues. However, the synonym-based matching, OWL and RDF's concept linking mechanism, and similarity scoring function do not guarantee contextual and structural understanding of the user's query and therefore returning low precision. User's based consolidation method can confuse users who are unfamiliar with the data structure being queried and may have limited knowledge on the formal query language. Therefore, a self-adaptive method that addresses the disambiguation of the complex query over big data structure is needed. This paper extends the Self-Adaptive Natural Language Interface (SANLI) model for question answering [11] scenario based on the dynamic concept-type identification and resolution.

The paper is organized as follows. The first part introduces the background of the research. The second part describes our perspectives on big data analytics and focus on the structured and unstructured data analysis

*Corresponding author:nurfadhlina@upm.edu.my

issues. This is followed by the details of the self-adaptive model in section three and result in section four.

2. Big Data Analytics

The emergence of Big Data has become a new source of opportunity for Semantic Computing researches. Open source frameworks in Big Data including Apache Hadoop and design patterns such as Map–Reduce present unique challenges for semantic technologies to provide a promising means for publishing, sharing, and interlinking data to facilitate data reuse [12-13]. However, even when these data become discoverable and accessible, significant challenges remain in achieving intelligent understandings of the anticipated data and scientific discoveries.

The LOD paradigm is the semantic technology to cope with Big Data, as it advances the hypertext principle from a web of documents to a web of rich data. Linked Data enables semantically interconnected, well structured, syntactically interoperable datasets that are distributed among several repositories either inside or outside organizations. The linking component of Linked Data, however, puts an additional focus on the variety issue; specifically the integration and conflation of data across multiple sources. The variety dimension associates the data heterogeneity both at the schema and instance level [14].

Querying LOD knowledge requires proficiency with the schema structure, formal query language (i.e. SPARQL) and understanding of semantic knowledge bases (KB) such as RDF and ontologies. Therefore, a tool that can assist the smart big data manipulation and querying is vital. Users prefer NL interfaces as compared to keyword, GUI or partial sentence based interfaces [15]. In a natural language interface (NLI) system, the NL input query should be translated into formal semantic query i.e., SPARQL. There are several efforts to assist querying the data such as the Query Wizard which is based on keyword search as an entry point and tabular interfaces for filtering and exploration [3].

However, this inherits the BDI ambiguity challenges. The magnitude of conceptual ambiguity increases when heterogeneous KB is queried due to variation of data type and data structure. Recent research studies on big data (e.g., DBPedia and Yago) querying tool have emphasized the deep analysis of queries and the techniques involved to translate to the NL query into a SPARQL equivalent [16–18]. There are generally two steps for BDI based on assistive query interface (as shown in Figure 1) which are (i) query decomposition, concept equivalence, schema mapping, ontology alignment/reconciliation, and (ii) query aggregation, aggregation rewriting, query federation.

Very limited work in NLI except [6–10] focused on the ambiguity problem in the query translation process. Ambiguity in complex query processing occurs when there is no exact matching or more than one possible matching between entity names or terms hence requires consolidation and approximation. Ambiguity can be classified into (i) linguistic, which happened due to the variation of the terms used in the user's query compared to the terms in the structure of the queried, and (ii) conceptual ambiguity which occurs during the mapping between several concepts in the queried data. The magnitude of ambiguity increases when heterogeneous KB is queried [19]. This paper will be evaluated against the ambiguity resolution as performed in [6] which has implemented clarification dialogue. On the contrary, this paper adopts the SANLI model that does not require interaction for the disambiguation.

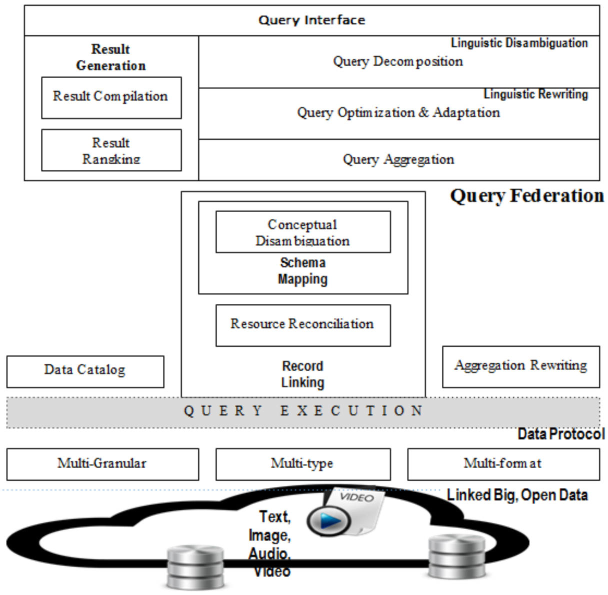

Fig 1. Semantic Analysis Querying on Big Data

Several researches have studied on various issues to map and fuse data on multiple sources as a means to translate the user query. The efforts include user interface design [20-21], usability [5], data management model [22–26], query language format (i.e., SPARQL) [27-28], query expressivity [4], [29-30], mapping [9], [26], [31-32], fusing [33–35] and ranking [3], [36–38]. Most of these approaches rely on linguistic triple (Subject-Predicate-Object) identification [38] which may be grammar and language dependent. Besides answering the queries require many customizations when involving untrained domain and dataset. This linguistic triple then provides the basis for translation to SPARQL. However, generating linguistic triples from NL queries is not a straightforward task due to the complexities of NL question patterns.

Also relying just on linguistic triples is not sufficient, as complex questions may contain queries for FILTER expressions, arithmetic operations (e.g. count, sum etc.), comparison (e.g. sorting), and negation (e.g. not, outside etc.). This is because triples should be uniquely identified while keeping the originality of query in consideration with linguistic ambiguity (e.g. lexical semantics ambiguity) resolution. The relations among different triples need to be identified and disambiguated as complex queries may contain multiple triples related to each other. Second, these triples need to be mapped with ontology triples (concepts and properties). This mapping also poses ambiguities when more than one linguistic triples are candidates for one ontology triple or vice versa. Besides, more than one KB can be a candidate for the retrieval of one concept.

3. Self-adaptive model for ambiguity resolution in big data integration

Impressive achievements of query disambiguation can be regarded as the clarification dialogue technique, disambiguation graph [18], [40], and the conjunctive SPARQL queries over a set of interlinked data sources [41]. However, it is found that there is a need to improve the ambiguity resolution through an adaptive model [41].

Therefore, we extend a self-adaptive model called the SANLI which was first introduced in [11]. The previous version of SANLI is limited compared to the version presented in this paper because it employed dice similarity matching, one-way traversing of the ontology graph and linguistic rules to find matching triples before proceeding with the SPARQL construction.

1. Read sentence and create a set of tokens $Q = \{t_1,...,t_n\}$.
2. Find the set of concepts similar to the set of tokens by searching the ontology concepts, O_c. We now have the probable concept set:
   $C = \{(t_i, c_j, w_j) : t_i \in Q, c_j \in O_c, w_j \geq \omega\}$
   where w_j is the similarity weight and ω is a predefined constant.
3. Find the set of similar relationships in the query by searching the ontology relationships, O_r. We now have the probable relationships set:
   $R = \{(t_i, r_j, w_j) : t_i \in Q, r_j \in O_r, w_j \geq \omega\}$
   where w_j is the similarity weight and ω is a predefined constant.
4. Create the initial SPARQL template with the initial triplet: $\langle ?s, ?p, ?o \rangle$.
5. for each of the *i* members of set R do:
   a. substitute ?o with c_i and run the query.
   b. if query has results do:
      i. for each of the *j* members of set C do:
         1. substitute ?p with r_j and run query
         2. if query has results save triplet$\langle ?s, r_i, c_i \rangle$ in possible answer set.
   c. else, substitute ?s with c_i and run the query.
   d. if query has results do:
      i. for each of the *j* members of set C do:
         1. substitute ?p with r_j and run query
         2. if query has results save triplet$\langle c_i, r_i, ?o \rangle$ in possible answer set.
6. Return answer which has the maximum sum of weights.
7. Select best answer set with the highest sum of concept/relationship similarity weight values

Fig 2. SANLI Algorithm.

The previous version of SANLI is called the MyAutoSPARQL [39] which has relied on the dice similarity matching technique to construct SPARQL queries. In this paper, we address the automatic mapping of the input with the data source, consolidate the ambiguity between sources granularity and construct the SPARQL by iterative querying of the KB structure before constructing the final SPARQL. This also allows SANLI to be domain and language independent. The algorithm to find the possible triples for the SPARQL is shown in Figure 2. The SANLI model is developed using the Java language and JENA as the reasoning engine.

The algorithm in Figure 2 starts by reading an ontology file and creating a memory structure of the classes, object properties and instances. Having an in memory structured copy benefits in later searching through the ontology, as shown in Figure 3. The first step is saving the NL query in a tokenized structure (as shown in Figure 4), which includes a matrix representation based on the concept type's annotations.

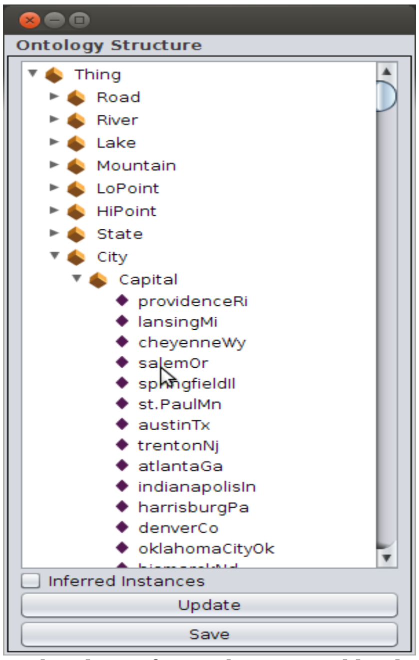

Fig 3. A graphical interface to browse and load ontology files.

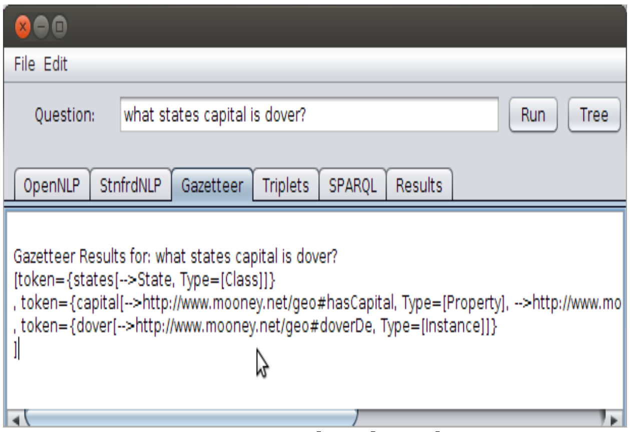

Fig 4. Gazetteer based matching.

In the second and third steps, the dictionary of probable concepts and relationships is created. The dictionaries are a structured array of relations found between each meaningful term in the NL query and its relevant object found in the ontology. The query is annotated with this dictionary. By annotating the query, we will have possible patterns of annotations for it. For example in the geography ontology, the query "What states border Oklahoma?" has a possible <s, p,o> pattern regarding <state:subject, borders:predicate, oklahoma:object>. Or for example the query "what is the capitol of Oklahoma" has a <p, o> pattern

(<isCapitolOf:predicate, Oklahoma:object>). Another example is provided in Figure 5.

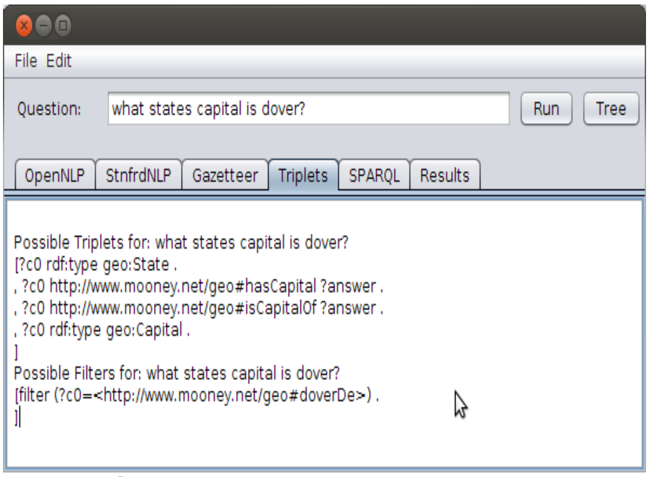

Fig 5. Triplets Generation.

In step 4, a general SPARQL template is initialized so that its elements could later be replaced and substituted. In step 5, we iterate through the relations to find the possibilities for each relation. In each iteration, possible concepts are substituted and tested to see if any results could be obtained. The ones giving results would be saved in the result answer set. In the last step, the SPARQL answer that has the maximum sum of weights would be returned Figure 6.

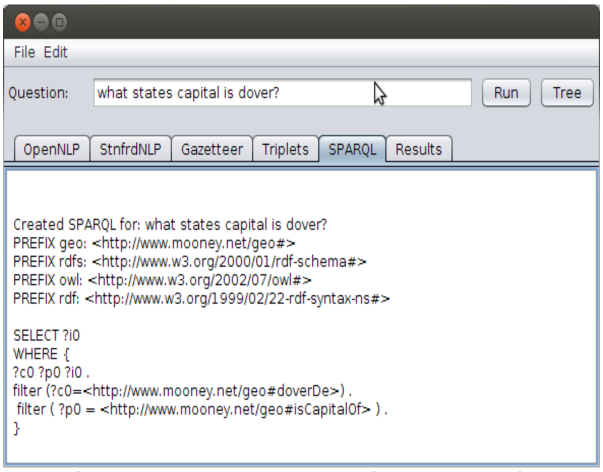

Fig 6. The SPARQL query executed run against the ontology file.

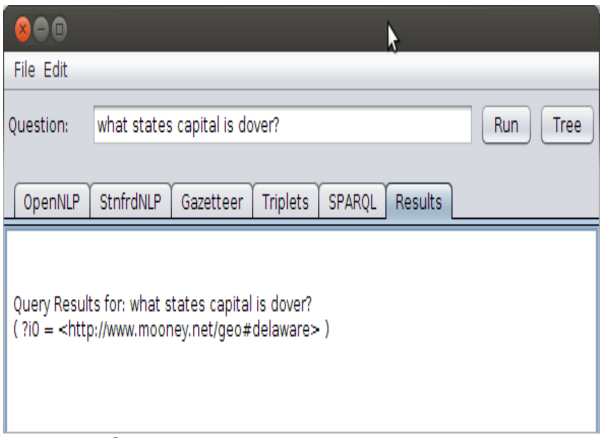

Fig 7. Results Generation.

Later, predefined rules specify if a triplet, filter or optional statement has to be included. For example, if a statement has negative words like “not”, “don’t” or other similar words, then the negation rules is applied. The negation rules is based on the filter operator on the object instance which is extracted from the input. Another example is when the input is a complex query which is characterized by the word “and”; where the rules utilize the optional operator. Figure 7 shows the answer from the generated SPARQL query. This approach is an improvement compared to the previous version of SANLI as proposed by [38], [39]. The new SANLI model is more flexible because of the disambiguation technique based on the dynamic concept-type identification and resolution.

4. Result

There are several levels of query complexity in BDI [39], [44-45] such as visualization, selection, path, negation, arithmetic, auxiliary and composition. In this paper we present the result of extended SANLI application on the visualization query type, which classifies queries that can be sufficiently answered through facet hierarchy, selection query type, which requires deeper processing and asks for count or list of items with a particular feature, and path query type, in which a path of properties is followed to retrieve answer, disjunction category which requires union (or "OR" conditions). Two different datasets namely the Mooney’s Geography ontology [42] and a Quran structure ontology [43] are used as preliminary assessment. Table 1 shows the examples of NL to SPARQL translation.

Table 1. Examples of automatic generated SPARQL queries for the Geography and Quran ontology dataset.

Ontology	Natural Language Query	Generated SPARQL
Geography	What is the lowest point in kansas?	SELECT ?c0 WHERE { ?c0 ?p0 ?i0 . ?c0 a geo:LoPoint . filter (?i0 = geo:kansas) . filter (?p0 = geo:isLowestPointOf) . }
	What is	SELECT ?i0

	the area of idaho?	WHERE { ?c0 ?p0 ?i0 . filter (?c0 = geo:idaho) . filter (?p0 = geo:stateArea) . }
Quran	Which Surahs were revealed in Medina?	SELECT ?c0 WHERE { ?c0 ?p0 ?i0 . ?c0 a quran:Surah . filter (?i0 = quran:Medina) . filter (?p0 = quran:revealedIn) . }
	Which verses belongs to surah 101?	SELECT ?c0 WHERE { ?c0 ?p0 ?i0 . ?c0 a quran:Verse . filter (?i0 = quran:101) . filter (?p0 = quran:belongToSura h) . }

There are 106 questions used in the Geography dataset while 30 are used in the Quran dataset. Table 2 shows the number of correct concepts identified in each dataset. In this experiment, the maximum number of concepts that exist in the questions in the Geography dataset is 3 while, in the Quran dataset up to 5 concepts can be utilized to answer the questions. The results indicate that SANLI has been able to identify the concepts for all questions in both dataset.

Table 2. Correct Concepts Identified.

Correct # concepts identified	Geography	Quran
0	0	0
1	18	2
2	88	11
3	N/A	13
4	N/A	3
5	N/A	1
Total	106	30

Table 3 shows the number of correct SPARQL constructed; which is determined based on the combination of correct concept names and type assignment. The results indicate that SANLI has better performance in constructing the SPARQL for the geography ontology compared to the Quran's dataset.

Table 3. Correct Concepts Names and Type Assignment in the Constructed SPARQL.

# Correct SPARQL	Geography	%	Quran	%
0	23	21.70	6	20.00
1	23	21.70	11	36.67
2	60	56.60	13	43.33
Total	106		30	

The SANLI average precision in the geography dataset is also better compared to the Quran dataset, as shown in Table 4. This is because the queries in the Quran dataset are more complex.

Table 4. Average SANLI Precision in Concepts Identification and SPARQL Construction.

Dataset	Concepts Identification	SPARQL Construction
Geography	0.9191	0.6990
Quran	0.8889	0.6167

The SANLI model is the continuation of semantic concept ambiguity resolution from our previous works such as MyAutoSPARQL [38], [39]. The previous approaches used are solely based on linguistic rules which demand many customizations for new semantic KB to be queried. Therefore, SANLI model is introduced which allows self-adaptive semantic resolution. However, the performance in terms of precision by the SANLI model is 4% lower as compared to MyAutoSPARQL model [38], as shown in Table 5. This is most probably due to SANLI's lacking in identifying the answer type of the queries which leads to inaccurate concepts incorporation in the translated SPARQL. In SANLI, the concepts are identified by continuous resolving of concepts which have active relationships in the ontology. Nevertheless, the SANLI performance is better compared to FREyA [38] which depends on user's manual resolution.

Table 5. Comparison of Semantic Concept Ambiguity Resolution.

System	Precision
MyAutoSPARQL[38]	0.7453
SANLI	0.6990
FREyA[38]	0.6887

5. Conclusion

The SW leverages the sophisticated analytics in big data by allowing data to be linked which can then be integrated for aggregative results. Applications like NLI are then introduced to assist the big data semantic querying. However, limited works have addressed the ambiguity problem in the NLI. This is crucial so that high precision of results can be generated. This paper introduces the extended SANLI, a self-adaptive based model for disambiguation in BDI scenario which exploits the triples identification in the natural language question and semantic KB structure through iterative querying to construct the final SPARQL queries. This is useful for application such as question answering and semantic search. SANLI improves the previous work on semantic search [11], [38], [39] by offering a more flexible SPARQL construction technique. The planned future work is to improve the SANLI model to include translation of questions containing arithmetic expressions and more complex questions in the BDI setting.

Acknowledgment

This study is sponsored by the Ministry of Higher Education Malaysia. The authors would like to thank the colleagues and researchers whom have contributed in this research such as Professor Dr. Shahrul Azman Mohd Noah, Assoc. Professor Dr. Masrah Azrifah, Dr. Rabiah Abdul Kadir, Dr. Danica Damljanovic, Mrs. Hazrina Sofian, Mr. Aliyu Elwa and Mr. Mutee Rahman. Your brilliant ideas and fruitful discussions are highly appreciated.

References

[1] V. Lopez, A. Nikolov, M. Sabou, and V. Uren, “Scaling Up Question-Answering to Linked Data,” in *Knowledge Engineering and Management by the Masses*, 2010, pp. 193–210.

[2] O. Gorlitz and S. Staab, “Federated Data Management and Query Optimization for Linked Open Data,” in *New Directions inWeb DataManagement 1*, 2011, pp. 109–137.

[3] P. Hoefler, “Linked Data Interfaces for Non-expert Users,” in *2013 European Semantic Web Conference*, 2013, pp. 702–706.

[4] J. Xu and R. Pottinger, “Integrating domain heterogeneous data sources using decomposition aggregation queries,” *Inf. Syst.*, vol. 39, pp. 80–107, Jan. 2014.

[5] E. Kaufmann and A. Bernstein, “How useful are natural language interfaces to the semantic web for casual end-users?,” in *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference*, 2007, pp. 281–294.

[6] D. Damljanovic, M. Agatonovic, and H. Cunningham, “FREyA: an Interactive Way of Querying Linked Data Using Natural Language,” in *Proceedings of the 8th international conference on The Semantic Web*, 2011, pp. 125–138.

[7] G. Correndo, M. Salvadores, I. Millard, H. Glaser, and N. Shadbolt, “SPARQL query rewriting for implementing data integration over linked data,” *Proc. 1st Int. Work. Data Semant. - DataSem ’10*, p. 1, 2010.

[8] A. Hogan, A. Harth, J. Umbrich, S. Kinsella, A. Polleres, and S. Decker, “Searching and browsing Linked Data with SWSE: The Semantic Web Search Engine,” *Web Semant. Sci. Serv. Agents World Wide Web*, vol. 9, no. 4, pp. 365–401, Dec. 2011.

[9] S. Hakimov, H. Tunc, M. Akimaliev, and E. Dogdu, “Semantic question answering system over linked data using relational patterns,” *Proc. Jt. EDBT/ICDT 2013 Work. - EDBT ’13*, p. 83, 2013.

[10] A. Hogan, A. Zimmermann, J. Umbrich, A. Polleres, and S. Decker, “Scalable and distributed methods for entity matching, consolidation and disambiguation over linked data corpora,” *Web Semant. Sci. Serv. Agents World Wide Web*, vol. 10, pp. 76–110, Jan. 2012.

[11] Nurfadhlina Mohd Sharef and M. Y. Shafazand, “Self-Adaptive Based Natural Language Interface for Disambiguation of Semantic Search,” in *Malaysian National Conference of Databases 2014 (MaNCoD 2014)*, 2014, pp. 87–92.

[12] M. Chen, S. Mao, and Y. Liu, “Big Data: A Survey,” *Mob. Networks Appl.*, vol. 19, no. 2, pp. 171–209, Jan. 2014.

[13] C. L. Philip Chen and C.-Y. Zhang, “Data-intensive applications, challenges, techniques and technologies: A survey on Big Data,” *Inf. Sci. (Ny).*, vol. 275, pp. 314–347, Aug. 2014.

[14] A. Anjomshoaa and A. M. Tjoa, “Towards Semantic Mashup Tools for Big Data Analysis,” in *ICT-EurAsia*, 2014, pp. 129–138.

[15] E. Kaufmann and A. Bernstein, “Evaluating the usability of natural language query languages and interfaces to Semantic Web knowledge bases,” *Web Semant. Sci. Serv. Agents World Wide Web*, vol. 8, no. 4, pp. 377–393, Nov. 2010.

[16] D. Damljanovic, M. Agatonovic, and H. Cunningham, “Natural Language Interfaces to Ontologies : Combining Syntactic Analysis and Ontology-based Lookup through the User Interaction,” in *Proceedings of the 7th Extended Semantic Web Conference (ESWC 2010)*, 2010, pp. 1–15.

[17] C. Unger, P. Cimiano, S. Computing, G. Citec, and U. Bielefeld, “Pythia: Compositional meaning construction for question answering,” in *16th International Conference on Applications of Natural Language to Information Systems, NLDB 2011*, 2011, pp. 153–160.

[18] M. Yahya, K. Berberich, S. Elbassuoni, M. Ramanath, V. Tresp, and G. Weikum, “Deep answers for naturally asked questions on the web of data,” *Proc. 21st Int. Conf. Companion World Wide Web - WWW ’12 Companion*, pp. 445–448, 2012.

[19] X. L. Dong and D. Srivastava, “Big data integration,” *Proc. VLDB Endow.*, vol. 6, no. 11, pp. 1188–1189, Aug. 2013.

[20] A. Freitas, J. G. Oliveira, S. O’Riain, E. Curry, and J. C. P. da Silva, “Treo : Best-Effort Natural Language Queries over Linked Data,” in *International Conference on Applications of Natural Language to Information Systems*, 2011, pp. 286–289.

[21] M. Gärtner, A. Rauber, and H. Berger, “Bridging structured and unstructured data via hybrid semantic search and interactive ontology-enhanced query formulation,” *Knowl. Inf. Syst.*, Sep. 2013.

[22] W. Lehner and K.-U. Sattler, *Web-Scale Data Management for the Cloud*. New York, NY: Springer

New York, 2013.

[23] U. Data and I. Patterns, “Data Modeling Approaches for Big Data and Analytics Solutions,” pp. 155–195.

[24] K. T. Sridhar and M. A. Sakkeer, “Optimizing Database Load and Extract for Big,” pp. 503–512, 2014.

[25] R. Gupta, H. Gupta, and S. Gupta, “A Middleware for Managing Big-Data Flows,” pp. 410–424, 2013.

[26] C. Sauer and T. Härder, “Compilation of Query Languages into MapReduce,” *Datenbank-Spektrum*, vol. 13, no. 1, pp. 5–15, Jan. 2013.

[27] D. F. Barbieri, D. Braga, S. Ceri, E. Della Valle, and M. Grossniklaus, “C-SPARQL: a continuous query language for RDF data streams,” *Int. J. Semant. Comput.*, vol. 4, no. 1, pp. 3–25, 2010.

[28] S. Madria, K. Passi, and S. Bhowmick, “An XML Schema integration and query mechanism system,” *Data Knowl. Eng.*, vol. 65, no. 2, pp. 266–303, May 2008.

[29] W. Fan, “Querying Big Social Data,” pp. 14–28, 2013.

[30] M. a. Martínez-Prieto, C. E. Cuesta, M. Arias, and J. D. Fernández, “The Solid architecture for real-time management of big semantic data,” *Futur. Gener. Comput. Syst.*, Oct. 2014.

[31] K. Makris, N. Bikakis, N. Gioldasis, and S. Christodoulakis, “S PARQL-RW: Transparent Query Access over Mapped RDF Data Sources,” in *15th International Conference on Extending Database Technology*, 2012, no. c, pp. 610–613.

[32] A. Margara, J. Urbani, F. van Harmelen, and H. Bal, “Streaming the Web: Reasoning over dynamic data,” *Web Semant. Sci. Serv. Agents World Wide Web*, vol. 25, pp. 24–44, Mar. 2014.

[33] F. Xhafa and L. Barolli, “Semantics, intelligent processing and services for big data,” *Futur. Gener. Comput. Syst.*, vol. 37, pp. 201–202, Jul. 2014.

[34] Z. Qu, “AISC 129 - Semantic Processing on Big Data,” vol. 2, pp. 43–48, 2011.

[35] A. I. Torre-bastida, “Incremental SPARQL Query Processing,” pp. 712–716, 2013.

[36] A. Lausch, A. Schmidt, and L. Tischendorf, “Data mining and linked open data – New perspectives for data analysis in environmental research,” *Ecol. Modell.*, no. 2014, Sep. 2014.

[37] A. Gyrard, “An Architecture to Aggregate Heterogeneous State of the Art,” in *2013*, 2013, no. 1, pp. 697–701.

[38] N. M. Sharef and S. A. Mohd Noah, “Natural Language Query Translation for Semantic Search,” *J. Digit. Content, Technol. Appl.*, vol. 7, no. 13, pp. 53–63, 2013.

[39] N. M. Sharef and S. A. M. Noah, “Linguistic Patterns-Based Translation for Natural Language Interface,” in *2014 International Conference on Information Science and Applications (ICISA)*, 2014, pp. 1–5.

[40] S. Ferre and A. Hermann, “Semantic Search: Reconciling Expressive Querying and Exploratory Search,” in *ISWC’11 Proceedings of the 10th International Conference on The Semantic Web*, 2011, pp. 177–192.

[41] N. M. Sharef and S. A. Mohd, “Soft Queries Processing In Natural Language,” in *International Conference on Ubiquitous Information Management and Communication*, 2012, pp. 1460–1466.

[42] L. R. Tang and R. J. Mooney, “Using Multiple Clause Constructors in Inductive Logic Programming for Semantic Parsing,” in *12th European Conference on Machine Learning (ECML-2001), Freiburg, Germany*, 2001, pp. 466–477.

[43] R. Iqbal, A. Mustapha, and Z. M. Yusoff, “An experience of developing Quran ontology with contextual information support,” *Multicult. Educ. Technol. J.*, vol. 7, no. 4, pp. 333–343, 2013.

.