International Journal of Advanced Smart Convergence Vol.12 No.2 34-46 (2023)
http://dx.doi.org/10.7236/IJASC.2023.12.2.34

IJASC 23-2-4

Semantic-based Mashup Platform for Contents Convergence

Yongju Lee[1], Hongzhou Duan[2], and Yuxiang Sun[3]

[1]*Professor, School of Computer Science and Engineering, Kyungpook National University, Korea*
[2]*PhD Student, School of Computer Science and Engineering, Kyungpook National University, Korea*
[3]*Doctor, Software Technology Research Center, Kyungpook National University, Korea*
E-mail: [1]yongju@knu.ac.kr, [2]caixiuming1984@163.com, [3]syx921120@gmail.com

Abstract

A growing number of large scale knowledge graphs raises several issues how knowledge graph data can be organized, discovered, and integrated efficiently. We present a novel semantic-based mashup platform for contents convergence which consists of acquisition, RDF storage, ontology learning, and mashup subsystems. This platform servers a basis for developing other more sophisticated applications required in the area of knowledge big data. Moreover, this paper proposes an entity matching method using graph convolutional network techniques as a preliminary work for automatic classification and discovery on knowledge big data. Using real DBP15K and SRPRS datasets, the performance of our method is compared with some existing entity matching methods. The experimental results show that the proposed method outperforms existing methods due to its ability to increase accuracy and reduce training time.

Keywords: *Semantic-based Mashup Platform, Knowledge Graphs, Entity Matching, Graph Convolutional Network, Realign Canberra Distance*

1. Introduction

The evolution of large scale knowledge graphs has made a new wave of big data researches [1]. The resource description framework (RDF) can be the data model for knowledge graphs, and SPARQL is the standard query language for this model. All data items in RDF are represented in the form of subject, predicate, and object triples. Spurred by efforts like WordNet, Freebase, and Wikidata projects, a huge number of semantic data are available in the RDF format in many fields such as science, business, social networks, and government. These large volumes of RDF data motivate the need for a scalable RDF content convergence system capable of efficiently organizing, discovering, and integrating RDF big data. This paper proposes a novel semantic-based mashup platform for contents convergence. Our platform has a life-cycle workflow architecture that has been implemented using an ontology learning method, a graph-based RDF storage, and open source components from the semantic web community. Components in the life-cycle do not exist in isolation but mutually fertilize themselves. The workflow architecture consists of four components; acquisition, RDF storage, ontology

Manuscript Received: March. 14, 2023 / Revised: March. 17, 2023 / Accepted: March. 21, 2023
Corresponding Author: yongju@knu.ac.kr
Tel:*** - **** - ****   Fax: +82-53-957-4846
Professor, School of Computer Science and Engineering, Kyungpook National University, Korea

learning, and mashup subsystems.

The main contributions of this paper are followings. First, we present a semantic-based mashup platform for contents convergence that can support everyone freely publish, link, and use knowledge graph data. This provides the next generation web environment that can easy development of creative ideas and share them. Second, we develop a compact storage and an efficient index structure for large scale RDF big data. We employ optimization techniques using the index structure and present an extension of the SPARQL query language for knowledge graph data. Third, we propose a new ontology learning method to generate ontologies automatically. The ontology learning method is based on entity matching using graph convolutional network (GCN) techniques. Fourth, we adopt the knowledge graph mechanism as a unified data model. Because all knowledge graph data share the RDF data model, which is based on a single mechanism for representing information, it is easy to realize the syntactic and semantic integration of different knowledge datasets. The advanced semantic integration can be achieved by our entity matching method based on GCN techniques. Unlike traditional mashup systems against a fixed set of data resources, our system can discover new data resources at runtime and deliver more complete answers as new data resources appear.

This paper is organized as follows. In Section 2, we describe an overview of our platform architecture. Section 3 describes detail description of our system. Section 4 proposes a new entity matching method based on graph convolutional networks. Section 5 describes the performance evaluation. Section 6 contains conclusions.

2. Architecture

This section presents a system architecture for the proposed platform. The platform is composed of four subsystems; acquisition, RDF storage, ontology learning, and mashup subsystems (see Figure 1). We give an overview over our system before we investigate each component in detail.

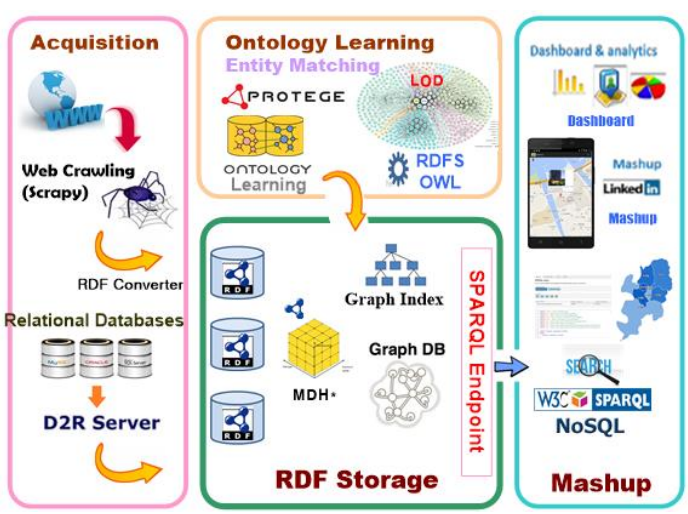


Figure 1. Semantic-based mashup platform for contents convergence.

(1) Acquisition: Unstructured or structured data must be converted to the RDF data. We crawl web sites and extract unstructured data using web crawling tool such as Scrapy [2]. The extracted properties are transformed into RDF triples. Structure data (e.g., relational databases) are transformed into RDF triples using D2R Server

[3]. D2R Server is a tool for publishing the content of relational databases on the semantic web.
(2) RDF Storage: Once there is a critical mass of RDF data, mechanisms have to be in place to store and index this data efficiently. Our system uses the graph-based RDF store. With this store, the system provides a SPARQL endpoint that allows any user to access the stored RDF triples. Our system indexes the triples stored in the RDF store. These triples are mapped into multi-dimensional histograms (i.e., MDH*) stored in the hybrid indexing system [4].

(3) Ontology Learning: RDF schema (RDFS) and web ontology language (OWL) are key semantic web technologies that give you a way to write down rich descriptions of your RDF data. Protégé [5] is a leading ontological engineering tool. In spite of using this ontological engineering tool, the construction of ontologies is a very expensive task which hinges on the availability of domain experts. In the next section, we investigate an ontology learning method to generate ontologies automatically.

(4) Mashup: A number of applications provided the browsing knowledge graph data (e.g., Tabulator, Marbles, Magpie), advanced searching facilities (e.g., Sindice, Sig.ma, Watson), and mashup Linked Data (e.g., DBpediaMobile, LinkedGeoData, ActiveHiring). Nevertheless, these applications hardly go beyond presenting together data gathered from different resources. Our system provides search capabilities such as SPARQL and NoSQL over the RDF triples. In addition, we can advance to build various applications based on knowledge graph data. For instance, we implement an aggregation of dashboards that presents various business analytics computed on knowledge graph data [6].

3. Detail Description of Our System

3.1 RDF Storage Subsystem

We adapt multi-dimensional histogram techniques for storage and indexing of knowledge graph data. The goal of our index structure is to support efficient join query processing without significant storage demand. In order to scale the query processor, we should design a compact storage structure and minimize the number of indexes used in the query evaluation. We develop a two-step index structure [4] comprising an approximate disk-based index structure (3D R*-tree) and flash memory-based *k*-d trees. We also present a hot-cold segment identification algorithm to determine ways for relocating data to disk and flash memory. For efficient query processing and join operations, we store data in the separated filter and refinement index structure [7]. The purpose of this structure is to filter unnecessary data and further refine the immediate result to improve the join query performance.

Each resource in RDF triple file is extended with two additional occurrences rather than a single RDF triple in order to speed up join queries. The occurrences specify s# and o#, where s# indicates the number of subjects in which an object occurs as subjects in the RDF dataset, and similarly, o# indicates the number of objects in which a subject occurs as objects. We observe that a fair number of triples in many real RDF datasets are used as subjects of a triple and objects of another triple.

We also propose a novel learned index structure using knowledge graph embedding and spatial clustering techniques for semantic search [8]. Knowledge graph data stored in the form of RDF can be automatically classified using the embedded model and density-based spatial clustering algorithm. Embedded vectors can quickly be matched to semantically similar clusters by comparing vector similarity between a given query and cluster centroids, thereby significantly reducing irrelevant traversal for complex semantic search. Figure 2 shows the layered architecture for the RDF storage subsystem.

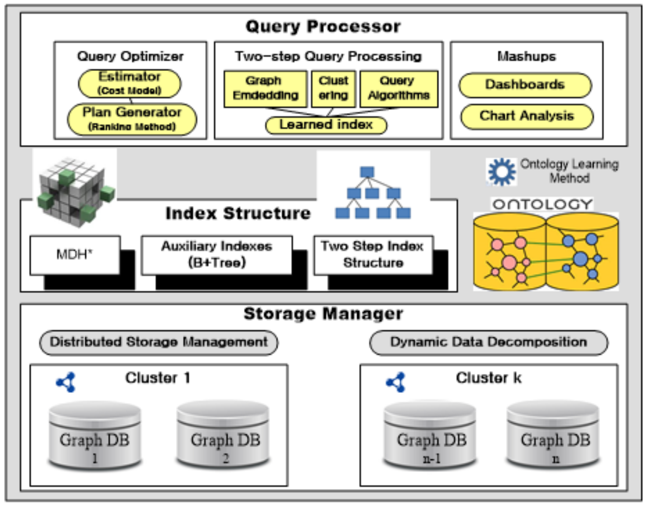


Figure 2. Architecture of RDF storage subsystem.

3.2 Ontology Learning Subsystem

The successful employment of knowledge graph data is dependent on the availability of high quality ontologies. Building such ontologies is difficult and costly, thus hampering knowledge graph data deployment. This research automatically generates ontologies from RDF big data and their underlying semantics. We focus on adapting knowledge graph data mining techniques to the syntactic descriptions of RDF triples. Since RDF was not designed for the ontology, it does not provide placeholders for high level syntax of the resources. We propose a ontology learning method to semantically describe knowledge graph data (see Figure 3).

We have developed a clustering technique [9] to derive several semantically meaningful concepts from Web APIs. We consider the syntactic information that resides in input/output parameters and apply a mining algorithm to obtain their underlying semantics. The main idea is to measure the co-occurrence of terms and then cluster the terms into a set of concepts. The pattern analysis technique [9] also captures relationships between terms contained in input/output parameters and matches items if both terms are similar and the relationships are equivalent. The ontology is generated from the set of parameters created in accordance with the pattern analysis rules.

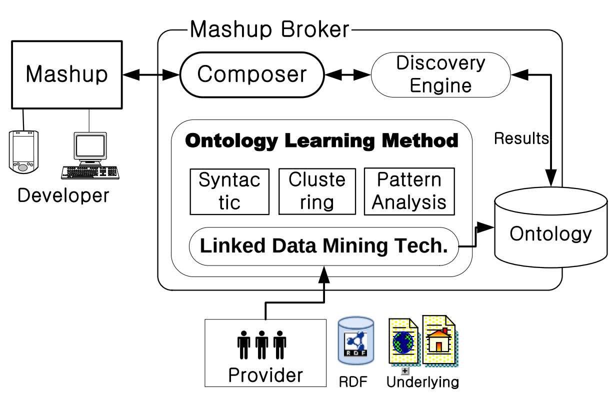


Figure 3. Ontology learning method and mashup broker.

We are currently developing an ontology matching system for RDF big data. More specifically, we are working on a semantic matching system for automatic classification and discovery on knowledge graph big data. As a preliminary work for implementing such a system, this paper proposes an entity matching method using GCN techniques. We describe this method in detail in Section 4.

3.3 Mashup Subsystem

With large scale knowledge graph data being published on the web, a number of efforts are under way to develop applications in regard to the semantic web. A key challenge now is to develop applications based on knowledge graph data. The emergence of knowledge graph data has been making an excellent revolution of semantic web applications. For instance, mashup is a web application that combines content from two or more services to create a new service. Although mashup has emerged as a very popular method of integrating web services, it is still suffer the data heterogeneity coming from various sources having different formats. To solve this problem, we adopt the knowledge graph data based on RDF as a unified data model. Unlike existing mashup applications against a fixed set of data sources, knowledge graph data applications can discover new data sources at runtime and deliver more complete answers as new data sources appear.

Figure 4 shows a semantic-based mashup application that uses smartphone capabilities. Multimedia, sensing, and communication capabilities of the smartphone provide the best environment in which to create easily new and superior mashup contents combined with knowledge graph data. The developed semantic-based mobile mashup can be able to find the desired place of your current location through GPS and access semantic information easily in a self-developed browser of knowledge graph data. We also develop a responsive web application for location-based semantic search (see Figure 5). These applications mashup DBpedia, a kind of Linked Data, and GoogleMap API provided by Google, and provide a semantic browser function to confirm detail information. These applications can be used to various access environments such as PC and mobile by applying responsive web design idea.

Figure 4. Semantic-based mobile mashup

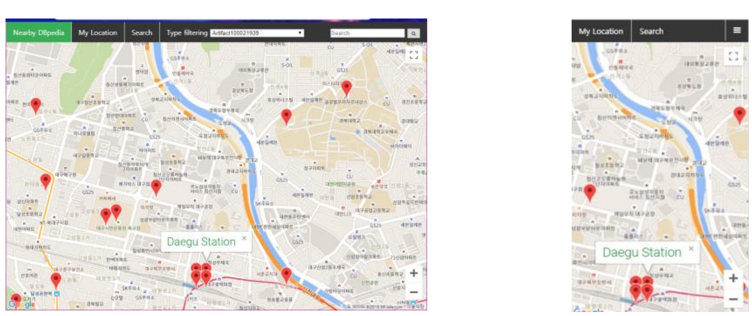


Figure 5. Responsible web application for PC and smartphone

We implement a visual analysis system that shows various chart results [6]. For RDF data analysis, we first write a SPARQL query statement for chart representation. It then queries the RDF storage where knowledge graph data are stored. By this process, we can receive results from SPARQL endpoint. Apache Jena framework is used to process this programming. We store results to HashMap and convert to the JSON format. Finally, draw the results. Figure 6 shows results for querying the number of restaurants by administrative district. This figure depicts the map of Daegu city. In addition, the number of restaurants by the food category can be indicated by a pie chart.

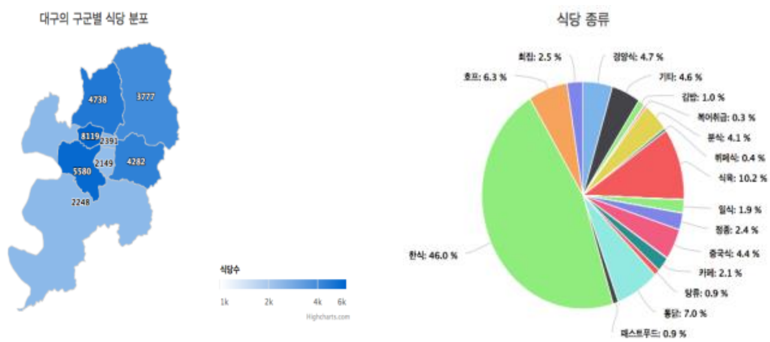


Figure 6. Visual analysis system for the number of restaurants

A mashup application provided by the semantic-based IoT system will provide new high value-added products that are completely different from what we have known and experienced. Our developed system collects IoT sensor data from the cloud computer, converts them into RDF data, and annotates them with semantics [10]. The converted semantic data are shared and utilized through the ontology repository. We use IoTMakers of KT (Korea Telecom) as a cloud computing environment, and the ontology repository uses Jena's Fuseki server. For mashup services, we provide SPARQL query results for the responsive web page using Map APIs of Daum and Open APIs provided by the platform. This gives people the opportunity to access the semantic IoT mashup service easily and has various application possibilities [10]. For IoT ontology modelling, we add new entities in the Fiesta IoT ontology model and build a new IoT ontology by establishing the relationship between each entity. As shown in Figure 7, our ontology model is visualized by using OntoGraf of Protégé.

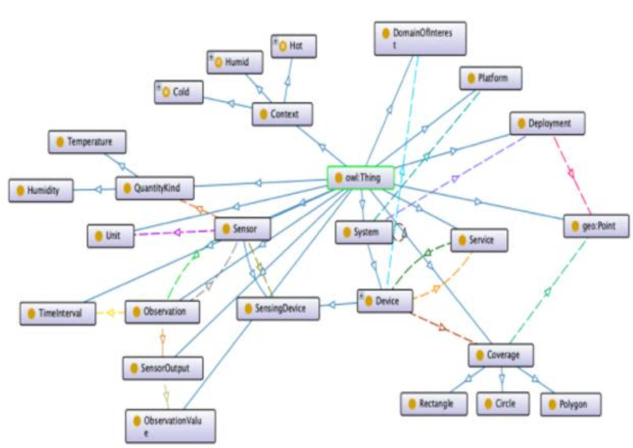
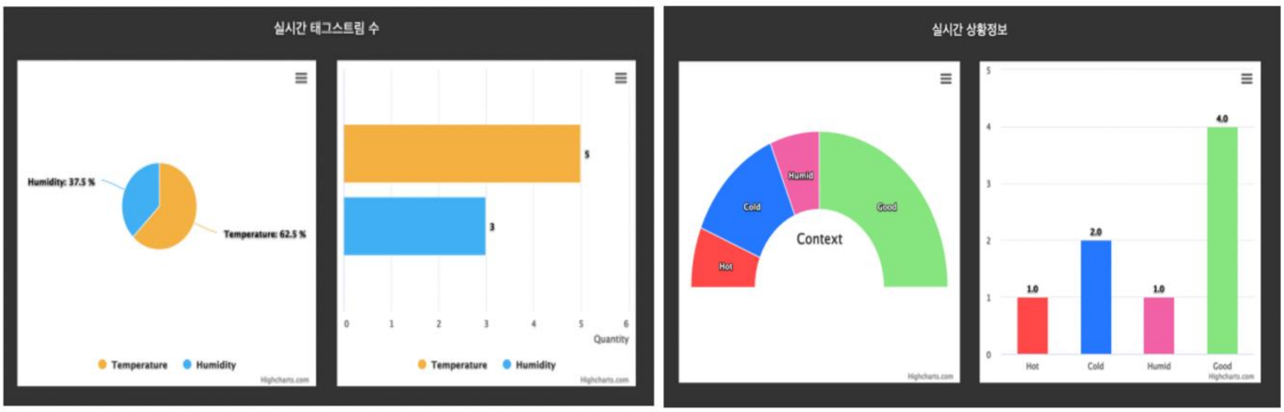


Figure 7. IoT ontology modelling

The dashboard page is implemented so that users can see sensor and context information in real time as shown in Figure 8. Figure 8 is implemented by using HighCharts [11] that is an open source which displays charts in JavaScript. Figure 8 (a) shows the entire tag stream through pie and bar graphs. The pie graph shows the percentage of each sensor, and the bar graph shows the number of sensors. Figure 8 (b) shows context information by measured values of sensors. If current space is set to Hot, Cold, Humid, and Good, you can see the context information by the semi-circle graph and the context information number by the bar graph.



(a) (b)

Figure 8. Sensor and context IoT information

4. GCN-based Entity Matching Method

Due to the recent rapid development in the field of neural networks, researches such as deep learning or data prediction for knowledge graph data are being actively conducted. However, there are relatively few studies on how to embed RDF triples into large-scale knowledge graph big data and apply it to train neural network models for entity matching. One of the most difficult issues to solve these problems is the lexical heterogeneity using different labeling. For instance, “beverage” and “fruit-juice” have the same semantic but cannot be matched since they do not belong to the same synonym set. Recently, considerable progress has been made in research applying embedding techniques to the field of automatic completion, but the accuracy of knowledge graph entity matching based on these techniques is still insufficient. Therefore, we propose an extended graph convolutional network (EGCN) model combining graph convolutional networks and realign Canberra distance structure to solve the lexical heterogeneity problem. In the first stage, we convert words into vectors using word embedding modes such as Word2Vec. Then, semantic similarity can be calculated from the spatial distance between vectors.

In the second stage, we use a dual graph convolutional neural network that better expresses complex edge structures and relationships, and consider word similarity scores together to improve matching accuracy. In this model, sorting is performed based on the distance of two knowledge graph entities created from the deep learning technique of the double graph convolution network, and the entity with the closest distance becomes the matching value found by deep learning. Based on this matching value found by deep learning, we measure the remaining entities that have not yet been precisely found with a similarity method, and finally match the entities found by the model.

Dual graphs are constructed from original graphs and then dual relations of these graphs are computed, which are learned through mutual interactions with dual graphs and original graphs. The output form at a vertex of a dual graph is as shown in equations (1), where N_i^r is a set of neighboring indices, $(D\_Att\_Sc)_{ij}^r$ is a dual attention score, G_j^r is a dual graph vertex value, and ELU is the activation function. The performance evaluation in Section 5 shows that ELU is the best choice for matching accuracy.

$$\widetilde{G_i^r} = \mathrm{ELU}(\textstyle\sum_{j \in N_i^r} (D\_Att\_Sc)_{ij}^r G_j^r) \tag{1}$$

Here, the dual attention score before normalization is as shown in equation (2), where $(c_i | c_j)$ is a formula that concatenates the vertex matrices related to the relation in the original data, FCL is a fully connected layer that maps inputs to scalars, and w_{ij} denotes a weight.

$$\mathrm{D\_Att\_Sc} = (c_i | c_j)(\mathrm{FCL}) w_{ij} \tag{2}$$

Finally, when this is normalized, it becomes the form shown in equation (3).

$$(\mathrm{D\_Att\_Sc})_{ij}^r = \frac{\exp\,(ELU((c_i | c_j)(\mathrm{FCL}) w_{ij}))}{\sum_{k \in N_i^r} \exp\,(ELU((c_i | c_j)(\mathrm{FCL}) w_{ij}))} \tag{3}$$

On the other hand, the dual attention score for a primal graph can be calculated similarly to the dual attention score, but the difference is that, $\widetilde{G_{ij}^r}$, which is the output form of the dual graph, is used for mutual interactions. The output form at the vertex of the primal graph is added to the initial matrix to clearly preserve the basis.

After several rounds of interaction between the dual graph and the primal graph, a relationship-aware entity representation can be obtained. Finally, GCN with highway gate is applied to the integrated adjacent information structure. The output expression $H^{(l+1)}$ generated by the GCN layer is as follows:

$$H^{(l+1)} = ELU(G^{(l)} W^{(l)} \widetilde{D}^{-\frac{1}{2}} \tilde{A} \widetilde{D}^{-\frac{1}{2}}) \tag{4}$$

Where $\widetilde{\mathrm{A}} = \mathrm{A} + \mathrm{I}$ is the adjacency matrix of the primal graph G^e, I is the identity matrix, $\widetilde{\mathrm{D}}$ is the degree matrix of $\widetilde{\mathrm{A}}$, and W is weights from convolution layer. The final entity representation generated from the output of the GCN layer is sorted through the distance between the two entities. To increase accuracy of entity matching, we present a realign structure based on the Canberra distance. First, entities are extracted from the GCN model. We then adopt the Canberra distance to evaluate entity similarity shown in equation (5). Finally, the minimum Canberra distance is added to the final alignment result.

$$\text{Canberra Distance} = \sum_{i=1}^{k} \frac{|G_{entity}^1 - G_{entity}^2|}{|G_{entity}^1| + |G_{entity}^2|} \tag{5}$$

5. Performance Evaluation

Experimental benchmark datasets (i.e., DBP15K and SRPRS) are used to evaluate the performance of our entity matching method using GCN techniques. The DBP15K dataset contains four language-specific knowledge graphs extracted from English (En), Chinese (Zh), Japanese (Ja), and French (Fr). Three sets are constructed to align entities between the other three language (Zh, Ja, and Fr) and En. The SRPRS dataset consists of cross-lingual (Fr-En and De-En) and mono-lingual (DBP-WD and DBP-YG) knowledge graph pairs, where En, Fr, De, DBP, WD, and YG describe English, French, German, DBPedia, Wikidata, and Yago 3, respectively. Table 1 and 2 present the statistics for the DBP15K and SRPRS datasets.

Table 1. DBP15K dataset

No.	DBP15K					
	Zn-En		Ja-En		Fr-En	
No. of relations	2,830	2,317	2,043	2,096	1,379	2,209
No. of attributes	379,684	567,755	354,619	497,230	528,665	576,543
No. of relation tuples	153,929	237,674	164,373	233,319	192,191	278,590

Table 2. SRPRS dataset

No.	SRPRS							
	Fr-En		De-En		DBP-WD		DBP-YG	
No. of relations	177	221	222	120	253	144	323	30
No. of attributes	53,045	60,800	55,580	73,753	64,021	133,371	58,853	18,241
No. of relation tuples	33,532	36,508	38,363	37,377	38,421	40,159	33,748	36,569

The experiment consists of the following. First, using the DBP15K and SRPRS datasets, entity matching of our method is performed to evaluate whether entities in two knowledge graphs match exactly. For experimental results, we use the Hit@K rate, which is commonly used in entity matching studies. Hit@K indicates whether the correct answer was found in the K-th among matching candidates.

Table 3 shows comparison results for embedding-based methods (i.e., MTransE [12] and BootEA [13]), RDGCN [14], and our method. Overall, the accuracy based on the GCN model (i.e., RDGCN and our method) is much higher than those based on embedding-based methods. Comparing the performance of the RDGCN and our method, Hit@1, which found entity matching at once, showed a performance improvement of about 5.1% from 69.99% to 75.1%. Hit@10 showed a slight improvement of 77.68% from the previous 76.5%. However, Hit@50 and Hit@100 did not show significant improvement. Figure 9 is a chart of the results in Table 3. In this curve, the X-axis represents Hit@K, and Y-axis represents the percentage of accuracy (%).

Table 3. Comparison results

Model	MTransE	BootEA	RDGCN	Our Method
Hit@1	22.7	35.63	69.99	75.1
Hit@10	42.68	62.41	76.5	77.68
Hit@50	58.45	71.32	88.3	89.12
Hit@100	61.3	79.56	92.52	93.4

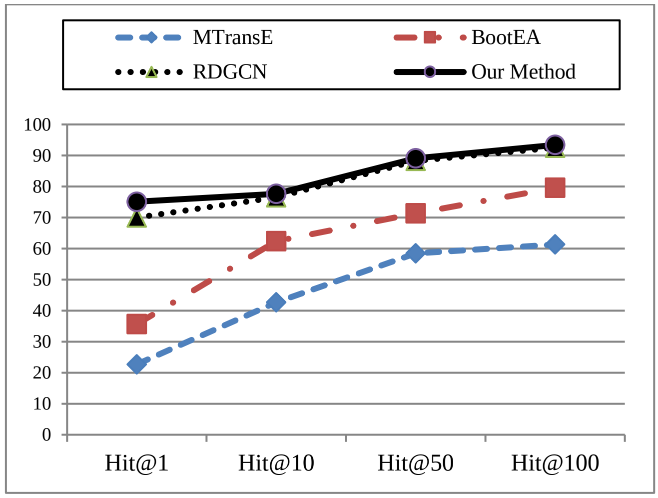


Figure 9. Chart of the results in Table 3

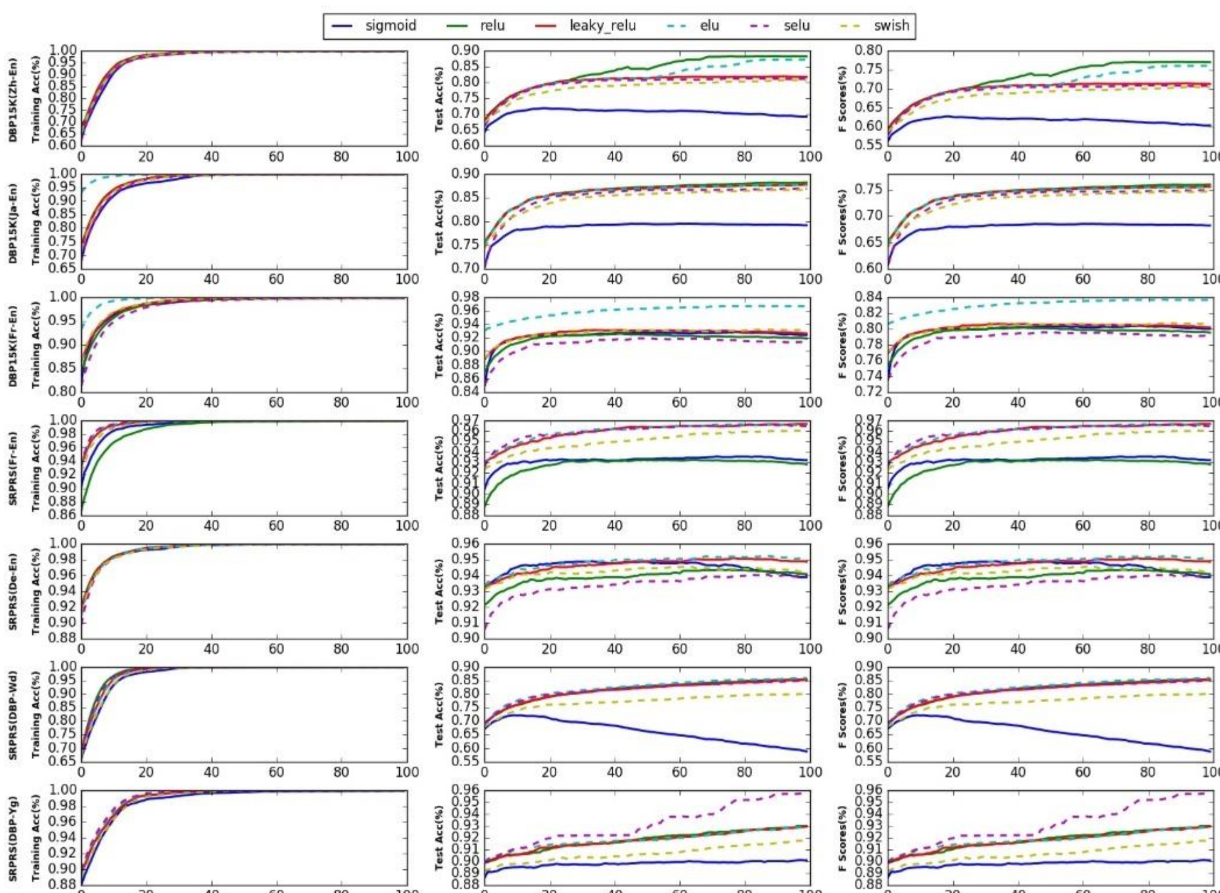


Figure 10. Alignment results for different activation functions

Figure 10 depicts alignment results for different activation functions based on our method for DBP15K and SRPRS datasets. We adopt recall, precision, and F-score as performance evaluation metrics. Our method does not overfit according to the training and test accuracy. The method produces matched entities similar to the number of benchmark matches with equal values for test accuracy because all entities in the SRPRS dataset are matchable. Regarding the DBP15K dataset, the number of benchmark matches is less than the number of training entities; therefore, the value of test accuracy is more than precision and equal to recall. The ELU activation function exhibits the best alignment performance in more than 50% of all cases, followed by SELU and ReLU. In terms of training time, ELU performs the best for over 80% of all cases. Considering the efficiency and performance, therefore, ELU is selected as the activation function.

Figure 11 presents the impact for different distance methods on the accuracy of entity alignment. Canberra exhibited excellent performance in DBP15K(Ja-En), DBP15K(Fr-En), SRPRS(De-En), and SRPRS(DBP-Wd) datasets. According to the training results in Figure 11, it can be inferred that Chebyshev's performance was the worst. Moreover, the training time of Canberra was the shortest for more than 85% of all cases. Therefore, the Canberra calculation method was adopted as the evaluation benchmark of entity alignment.

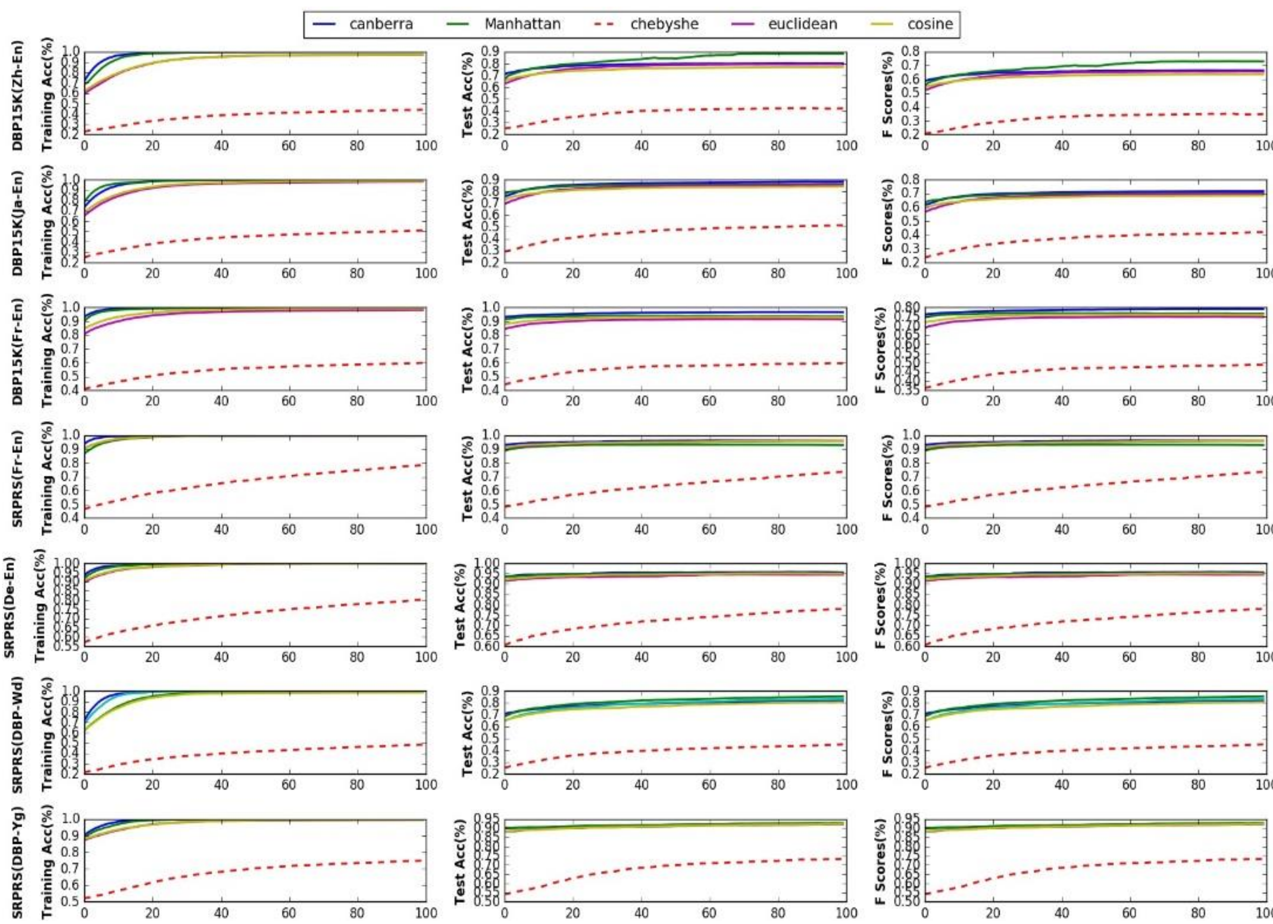


Figure 11. Alignment results for different distance methods

6. Conclusion

This paper presented a semantic-based mashup platform for contents convergence, which consists of acquisition, RDF storage, ontology learning, and mashup subsystems. This platform has an integrated hybrid architecture which supports the whole life-cycle of knowledge graph data from information acquisitions, building ontologies, knowledge storage and discovery, and developing applications. Our platform can be used by data publishers for the knowledge big data management ranging from contents authoring, dataset linking, and information enrichment to mashup exploration and searching. The main components of the system are open sources in order to facilitate wide usage and ease the scalability. We describe the overview of our ongoing project in this paper. This proposal will serve a bridge for research aiming at increasing extensibility of existing techniques and developing real applications of knowledge graph data.

We proposed an entity matching method using GCN techniques to improve alignment accuracy. Compared with existing matching methods, our method has the highest efficiency. The contribution of this paper is that it is possible to use the extended GCN model that expresses complex edge structures and relationships better than existing GCN and embedding-based models. Using real DBP15K and SRPRS datasets, we evaluated the performance of our method with some existing entity matching methods. The experimental results showed that our method significantly improves entity matching performance compared with existing methods.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 2016R1D1A1B 02008553). This study was supported by the BK21 FOUR project (AI-driven Convergence Software Education Research Program) funded by the Ministry of Education, School of Computer Science and Engineering, Kyungpook National University, Korea (4199990214394)

References

[1] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, “A Survey on Knowledge Graphs: Representation, Acquisition, and Applications,” IEEE Transactions on Neural Networks and Learning Systems, Vol. 33, No. 2, pp. 494-514, Feb. 2022. https://doi.org/10.1109/TNNLS.2021.3070843

[2] Scrapy: An Open Source Web Scraping Framework for Python. Available: http://scrapy.org/

[3] http://d2rq.org/d2r-server.

[4] Y. Sun, T. Zhao, S. Yoon, and Y. Lee, “A Hybrid Approach Combining R*-tree and k-d Trees to Imporve Linked Open Data Query Performance,” Applied Sciences, Vol. 11, No. 2405, Mar. 2021. https://doi.org/10.3390/app11052405

[5] http://protege.stanford.edu.

[6] E. Jung and Y. Lee, “Linked Data based Storage/Application Platform and Implementation of Analysis System Visualization,” Journal of KIIT, Vol. 16, No. 9, pp. 95-102, Sep. 2018. http://dx.doi.org/10.14801/jkiit.2018.16.9.95

[7] T. Zhao and Y. Lee, “Join Query Performance Optimizatin Based on Convergence Indexing Method,” Journal of KIECS, Vol. 16, No. 1, pp. 109-115, Feb. 2021. http://dx.doi.org/10.13067/jkiecs.2021.16.1.109

[8] Y. Sun, S. Chun, and Y. Lee, “Learned Semantic Index Structure using Knowledge Graph Embedding and Density-based Spatial Clustering Techniques,” Applied Sciences, Vol. 12, No. 6713, Jul. 2022. https://doi.org/10.3390/app12136713

[9] Y. Lee, “Semantic-based Data Mashups using Hierarchical Clustering and Pattern Analysis Methods,” Journal of Information Science and Engineering, Vol. 30, No. 5, pp. 1601-1618, Sep. 2014. https://doi.org/10.6688/JISE.2014.30.5.17

[10] H. Seok, S. Nam, and Y. Lee, “Implementing a Semantic-based IoT Mashup Service,” 2019 2nd Artificial Intelligence and Cloud Computing Conference, Dec. 2019, Kobe Japan. https://doi.org/10.1145/3375959.3375975

[11] Highsoft, HighCharts. DOI=https://www.highcharts.com, 2009

[12] M. Chen, Y. Tian, M. Yand, and C. Zaniolo, “Multilingual Knowledge Graph Embeddings for Cross-lingual Knowledge Alignment,” International Joint Conference on Artificial Intelligence, pp. 1511–1517, 2017. https://doi.org/10.48550/arXiv.1611.03954

[13] Z. Sun, W. Hu, Q. Zhang, and Y. Qu, “Boot-strapping Entity Alignment with Knowledge Graph Embedding,” International Joint Conference on Artificial Intelligence, pp. 4396–4402, 2018. https://doi.org/10.24963/ijcai.2018/611

[14] Y. Wu, X. Liu, Y. Feng, Z. Wang, R. Yan, and D. Zhao, “Relation-aware Entity Alignment for Heterogeneous Knowledge Graphs,” International Joint Conference on Artificial Intelligence (IJCAI), Macao, China, pp. 5278-5284, Aug. 2019. https://doi.org/10.24963/ijcai.2019/733