Intelligent System for Ranking Big Data in Search Engine

M.M.El-Gayar [1], and Ibrahim M. EL-Hasnony[1]
[1] Faculty of Computers and Information, Mansoura University, Egypt
Emails: {< mostafa_elgayar; ibrahimhesin2005> @Mans.edu.eg}

Abstract: The spread of Internet sources has increased the volume of big data that is difficult to handle in traditional ways. So, most users need modern search systems to facilitate the search and retrieval of information in the presence of big data. However, the main challenge in the first and second conventional generations of search engines are linking different web data based on the syntax of keywords not on the semantic meaning and without a knowledge base. This manuscript proposes a framework based on modern technologies such as ETI processes, ontology graphs, and indexing RDF using wide column NoSQL technique. The main contribution of our work is introducing a mathematical model that is used to calculate the similarity score between a query and stored RDF documents based on semantic relations. Various operations were carried out to measure the proposed model's efficiency using data sources such as DBpedia, YAGO dataset. According to experimental results, the proposed model is achieving high precision compared to other related systems.

Keywords: Search Engine, Big Data, Ontology, Semantic Web, NoSQL

1. Introduction

Search engines passed through many generations. In the first generation, the search engine relied on displaying information that matched the input query only. In the current second-generation, search engines must analysis the text firstly using machine learning techniques, and then extract the valuable keywords based on the syntax form of words not semantic [1].

Today it is considering a part of many enterprises to provide access to information. It has become the largest computing technology or intelligent technology.

Internet is the largest syntax source of information ever. Static web pages as HTML and other online resources described the syntax structure of information, not the semantic relations. So, a traditional search engine can understand syntax form only, but it cannot infer what we need. Today's, the third generation of search engines uses semantic relations between different things to build a knowledge graph, as shown in Figure 1 [2].

Ontology described the elements that exist or may exist in any field or area and frequently applied to represent semantic relations. The classes of ontology describe the verbs, word senses, or notion and association types. Effective tools for any semantic search engine based on the ontology are OWL and RDF that makes the information readable. The Resource Description Framework (RDF) is a language for interpreting information about sources on the internet. The subject term is about the thing we're talking about. The predicate is a term used to describe or modify some aspect of the subject [3] [10] [11].

DOI: 10.5281/zenodo.5202917

Received: March 17, 2021 Accepted: July 14, 2021

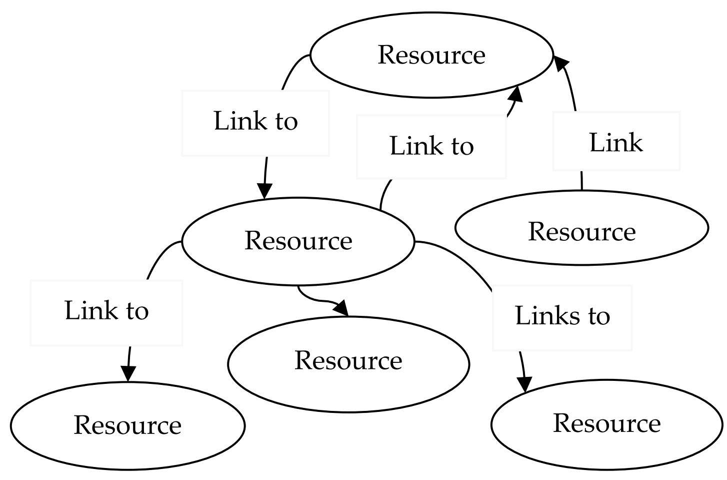

Figure 1. The General Structure of Semantic Web

In RDF, the predicate is used to denote relationships between the subject and the object. The object is the value. It can be another resource or just a literal value such as a number or word. There are many operational challenges and research points related to search engines, which will be listed in this paragraph [4-7]. These challenges will be summarized as follows:

a) Most current traditional or standard search engines scale well, but they have relied on syntax relationships between keywords or topics, not semantic relations.
b) Most modern semantic search engines do not scale well and focus on only one field in the search area.
c) If we need to enhance the knowledge base of any search engine, big data challenges such as time complexity, scalability, and availability must be addressed.

Our contributions in this article fall into three major categories. These contributions will be reviewed as follows:

a) Semantic search system framework is recommended.
b) Enhancing knowledge base using more than one source using proposed ETI (Extracting, Transforming and Integrated) processes.
c) A novel mathematical representation for determining semantic relations between different subjects is proposed and evaluated with an example.
d) Different testing factors are hold for evaluating recommended framework.

The remainder of this article will be divided as follows: The background of different information retrieval models and similarity measurement techniques are presented in section 2. Next, the related works of semantic search systems are presented in section 3. Then, the proposed framework and a novel algorithm will be presented with a mathematical model in section 4. Next, the experimental outcomes and evaluation criteria of the proposed framework are handled in section 5. Lastly, the conclusion is described in section 6.

DOI: 10.5281/zenodo.5202917
Received: March 17, 2021 Accepted: July 14, 2021

2. Background

In this section, we will briefly review and summarize a simple survey of the most common forms of IR models. Also, different conventional methods of measuring the degree of similarity at the level of characters or words will be briefly reviewed.

2.1 Different Models of Information Retrieval Systems

An IR (Information Retrieval) model governs how a document and a query are represented. There are two common main models: Boolean and vector space model. The model that we use in this paper is the second model (Vector Space Model) [6].

I. Boolean Model: This model is the simplest and oldest model used in search engines. The concept of exact match and the laws of Boolean algebra are used to match the search results with the query entered by the user. The weight of keyword $\boldsymbol{k_i}$ in page $\boldsymbol{p_j}$ is calculated according the following crisp equation:

$$\boldsymbol{W_{ij}} = \begin{cases} \boldsymbol{1}, & \boldsymbol{if\ k_i\ appears\ in\ p_i} \\ \boldsymbol{0}, & \boldsymbol{otherwise} \end{cases} \quad (1)$$

II. Vector Space Model: In this model, a document is represented as a weight vector. The weight of keyword $\boldsymbol{k_i}$ in page $\boldsymbol{p_j}$ is no longer {0,1} as in the previous model, but can be any number. We can use KF-IPF (Keyword Frequency - Inverse Page Frequency) scheme to assign weight for each component. Let M be the total number of pages in the system and $\boldsymbol{pf_i}$ be the number of pages in which keyword $\boldsymbol{k_i}$ appears at least once. Let $\boldsymbol{f_{ij}}$ be the frequency count of keyword $\boldsymbol{k_i}$ in page $\boldsymbol{p_j}$. Then keyword frequency ($\boldsymbol{kf_{ij}}$) is given by the equation (2). The inverse document frequency ($\boldsymbol{ipf_{ij}}$) is given by the equation (3). Finally, we can get component's weight using the equation (4).

$$\boldsymbol{kf_{ij}} = \frac{\boldsymbol{f_{ij}}}{\boldsymbol{max\,\{f_1, f_2, \dots, f_m\}}} \quad (2)$$

$$\boldsymbol{ipf_{ij}} = \boldsymbol{\log \frac{M}{pf_i}} \quad (3)$$

$$\boldsymbol{w_{ij}} = \boldsymbol{kf_{ij}} \times \boldsymbol{ipf_{ij}} \quad (4)$$

2.2 Character-Based Similarity Measures

Needleman-Wunsch method is a dynamic model commonly used in biology and bioinformatics to compare genome sequencing as it has recently been used in some standard search engines. DNA consists of a large sequence of a specific set of string characters (e.g. ACGT) where this method measures the degree of similarity by aligning two entire sequences. The usage of this method can be logical and correct if the two sequences are of the same length and share a high degree of similarity [7].

Smith-Waterman method is also another example of dynamic programming model and is also used in the field of bioinformatics, which can compare two sequences by aligning each character with the other to obtain the highest degree of similarity. This method does not require that the two sequences have the same length and have a high degree of similarity. But the good thing about this method is that we can extract similar areas in any two sequences and don't necessarily have a complete similarity [8].

Longest Common SubString (LCS) method is a sequential comparison between two sequences and the degree of similarity depends on the length of similar the contiguous chain of characters. That is, the longer

DOI: 10.5281/zenodo.5202917

Received: March 17, 2021 Accepted: July 14, 2021

the length of the string containing similar characters, the greater the proportion of similarity even if the meaning is different.
Levenshtein method, unlike the last two methods, calculates the degree of similarity between the two sequences based on the number of operations that match these two sequences. These operations such as deletions, substitutions, or insertions can be calculated on a sequence of characters.
That is, the degree of similarity here reflects the distance between the two sequences. This method is used in some search engines to fix errors in the user-submitted query in accordance with one of the terms in the database used [2] [3].

2.3 Term-Based Similarity Measures

Cosine method is a similarity score mathematical measurement between two vectors that calculates the cosine of the angle between them. The weight of each Wikipedia article is calculated by this method.
Euclidean distance (ED) method is the distance measurement between two strings or two vectors. The similarity score between two elements is one minus of the distance of these two elements. Distance is calculated as the square root of the sum of squared differences between related elements.
Dice's Coefficient is described as twice the amount of common terms in the associated strings divided by the total number of words in both strands [12] [17] [18].

2.4 Semantic-Based Similarity Measures

Explicit Semantic Analysis method (ESA) is a measure used to estimate the semantic relatedness among two arbitrary subjects. The relatedness of pair documents in the same language is evaluated by this method.
Latent Semantic Analysis (LSA) is a standard popular method of estimating semantic-based similarity. LSA assumes that strings are close in meaning will happen in similar parts of the text. LSA includes a matrix which holds word counts per paragraph. In the matrix, rows describe distinct words and columns represent a specific paragraph.
Second-order Ocuurence Pointwise Mutual Information (SOC-PMI) is a semantic score measurement method using pointwise information to sort lists of significant neighbor terms of two strings. The relation score between two strings that do not occur regularly can be calculated.

3. Related Works

Standard search systems are organized into search engines such as Google, Yahoo, Bing, and directories such as DMOZ, and Meta Search systems such as Dogpile, Mamma. Most standard search mechanisms are very popular, but their results are sometimes inaccurate, with lower precision and high recall. Modern and intelligent search systems, like Swoogle, SWSE, Falcons [19] Object Search, etc…, are designed based on the semantic approach.
Swoogle [20] is a system that relies on semantic crawling and indexing of web documents. It is divided into four main components: data discovery, metadata creation, data analysis, and retrieval. It allows metadata classification using the rational surfer model. But, these systems had limitations, such as limited indexing of massively large data and time-consuming query response. Also, Swoogle doesn't contain a ranking method to sort the more appropriate outcomes.
Hogan et al. [21] have represented the semantic method named Semantic Web Search Engine (SWSE). It is a complete search system that gives services comparable to standard search engines based on RDF and

DOI: 10.5281/zenodo.5202917
Received: March 17, 2021 Accepted: July 14, 2021

link-related data. It consists of the crawling method, indexing, ranking, argumentation, and retrieval stage. But, SWSE has some weaknesses, such as (i) the system doesn't scale adequately, and (ii) the system doesn't look robust in the face of heterogeneous, noisy, brazen, and perhaps conflicting data collected.
Hakia [24] is another system that acts as a comprehensive semantic engine that works for specific purposes. This system is another type of search engine that results in getting more accurate and reliable results from regular search engines. It consists of many components like query processor, ontology analyzer, QDex storage, and ranking approach. This system, like other systems, relies on linking results in terms of meaning rather than in terms of statistical methods, which gives strength and credibility to the results.
Fatima et al. [23] proposed a system that also depends on the semantic web. The proposed search engine in this system relies on a powerful query language processor and an easy user interface to sorting and retrieving data. Regardless of the technical methods used in this system, it lacks new algorithms that improve system performance.

4. Proposed Framework

The suggested framework called SEEK (Semantic Engine based on Enhancing Knowledge) is designed in a modular way and reasonably composed of two separated stages. Firstly, the offline stage is a back-end where the server runs solely away from users. Offline phase is involving ETI and indexing based on semantic relation processes. Secondly, the knowledge source phase used to convert JSON format into RDF schema and store related subjects into cache table for faster retrieval. Thirdly, the online phase is a front-end stage where a user can operate directly with the knowledge source at a real time. As presented in Figure 2, Each bold boxes of this figure focused on the contributions of this paper.

4.1 Offline Phase

This phase consists of three integrated stages. These stages are ETI (extract, transform, and integrate) processes, indexing process, and similarity score calculation process based on semantic relation.

A. ETI Stage

This stage is responsible for ETI processes that extract data from different data sources and convert it into uniform JSON format. The ETI job manager on a master thread invokes schema analyzer to identify the schema of the API data sources (e.g., Facebook, Twitter, DBpedia APIs). This stage outputs a uniform integrated schema in JavaScript Object Notation (JSON) format. This part is one of the paper contributions. ETI processes are initiated on the master thread and distribute ETI processes on several slave threads using the MapReduce technique for parallel processing. The extract process involves fetching data from the source, which should be correct and accurate. The transform process performs a series of complicated data cleaning and conversion it into JSON format. The integration process used to bind extracted data into other JOSN format.
Algorithm 1 represents the overall steps of how the schema is being created. In the schema creator procedure, the master thread parallelly assigns collections of different data sources (as input) to slave threads for creating the schema (as output). After that, Algorithm 2 shows how data can be extracted. In the extraction procedure, the inputs are the data source, the number of threads assigned to this data source (thread counter) and schema design from the previous method. The output of this procedure is Row data in batches.

DOI: 10.5281/zenodo.5202917

Received: March 17, 2021 Accepted: July 14, 2021

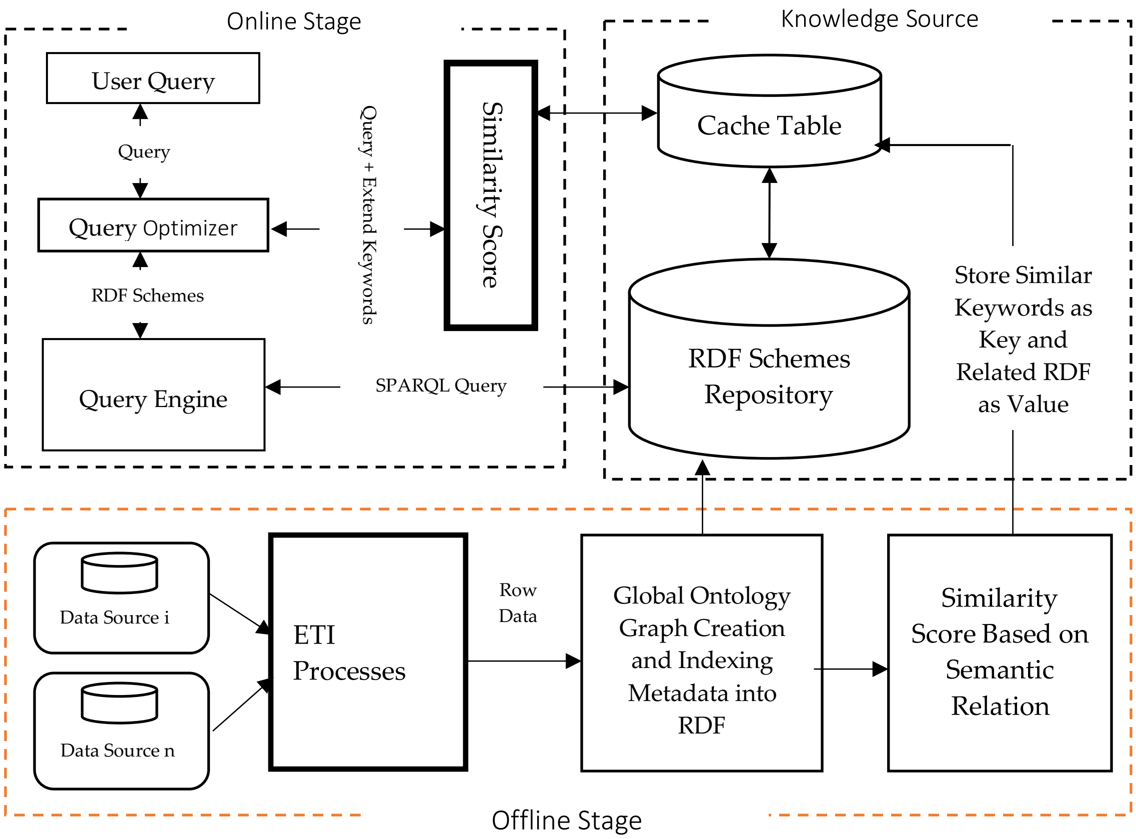

Figure 2. Proposed SEEK Architecture

```
Algorithm#1 Schema Creator Procedure
Input ← Data Source (DS_i)
Output ← Schema
Begin
  dataset ← DS_i.open connection
  count ←dataset. collection. Names ()
  Schema ← null
  For i ← 0 to count do
    Foreach record ∈ dataset.collection(i) do
      Foreach column_name ∈ record do
        Schema ← schema ∪ column_name
      End Foreach
```

DOI: 10.5281/zenodo.5202917
Received: March 17, 2021 Accepted: July 14, 2021

```
      End Foreach
    End For
    Return Schema
End Schema Creator Procedure
```

```
Algorithm#2 Extraction Procedure
Input ← Data Source (DS_i) , thread counter (TC),
            Schema
Output ← Row Data
-------------------------------------------------
Begin
    dataset ← DS_i.open connection
    count ←dataset. collection. Names ()
    Limit ← Count / TC
    For i ← 0 to count do
        Start ← limit * i
        Batch ← Read data (batch size)
        Row Data ← Extract (process id, Batch,
                       DS_i, start, limit, schema
    End for
    Return Row Data
End Extract Procedure
```

Finally, transformation procedure can record all the distinctions in the data source and compile the schema into JSON format as shown in Algorithm 3. In the transformation procedure, the inputs are the data source, extracted row data, and schema design. The output of this procedure is integrated data in JSON format.

```
Algorithm#3 Transformation Procedure
Input ← Data Source (DS_i) , Row Data, Schema
Output ← Data in JSON format
-------------------------------------------------
Begin
    Initialize schema ()
    Data Collection ← Row Data. Length
    Limit ← Data Collection .Count / PC
    Start ← 1
    While Start < Limit do
        Clean (Row Data)
        Remove Duplicates (Row Data)
        Schema ← Bind Row Data
        Start ++
    End While
    JSON ← Schema
    Return JSON
```


DOI: 10.5281/zenodo.5202917
Received: March 17, 2021 Accepted: July 14, 2021

14. End Transform Procedure

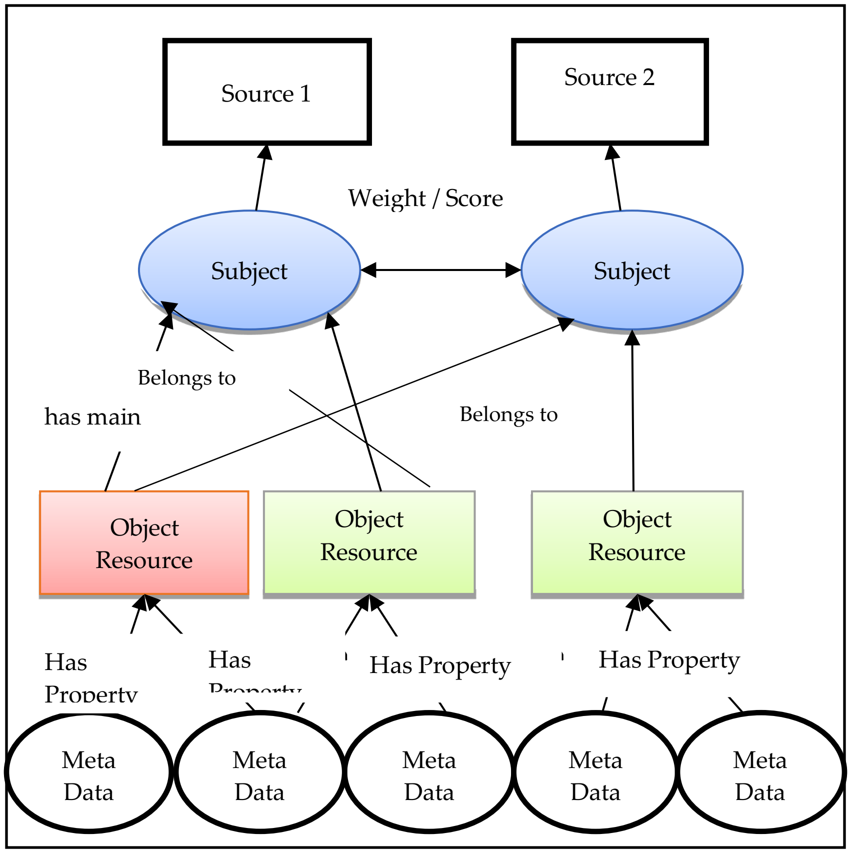

Figure 3. Structure of the Semantic Ontology Graph [5]

This stage is responsible for creating ontology graph, create RDF schema and store RDF files. The subject of ontology is the classes reflection of objects that exist or may exist in some field. The first step is the root node creation process. The root node is the source of the domain, then the related subjects with the corresponding objects are added in the given hierarchy, as shown in figure 3. The second step is the creation process of children nodes.

B. Similarity Score Stage

The similarity score measurement based on semantic relation is also one of the main contributions of this paper. Semantic Score (SS) equation used to measure weighting score between two subjects (I and J) in ontology graph [5].

$$SS_{in}(S_i, S_j) = 1 - \left(\frac{\max(\log S_i, \log S_j) - \log(S_i \cap S_j)}{\log T - \min(\log S_i, \log S_j)} \right) \quad (5)$$

$$SS_{in}(S_i, S_j) = \begin{cases} 0, & S_i \neq S_j \\ 1, & S_i = S_j \\ otherwise, & S_i \ related\ to\ S_j \end{cases} \quad (6)$$

Where:

DOI: 10.5281/zenodo.5202917

Received: March 17, 2021 Accepted: July 14, 2021

- Si is the count of input relations of subject I from different objects.
- Sj is the count of input relations of subject J from different objects.
- T is the total number of subjects in the graph.

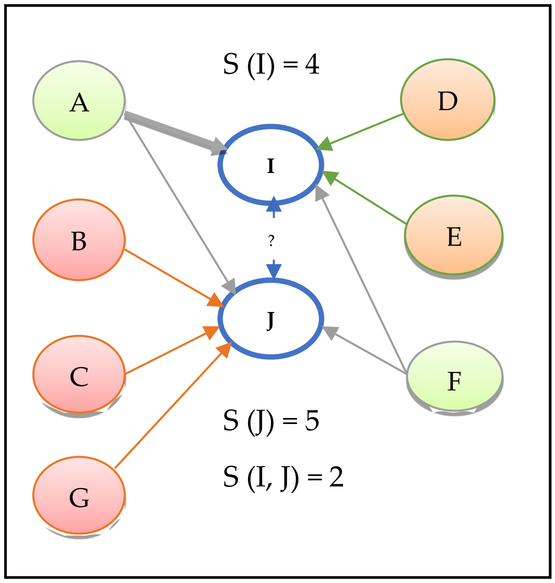

Figure 5. Example of Calculating Semantic Score between two subjects (I and J)

Figure 5 illustrates an example of calculating semantic score between two subjects (I and J). if the subject i has four incoming links from different objects (A, D, E, and F). Then the semantic relation of subject (I) was four. Also, if the subject (J) has five incoming links from different objects (A, B, C, F, and G).
Then the semantic relation of subject (J) was five. However, there are two objects (A and F) belonged to both subjects. So, semantic relation of both subjects (I and J) was two. If the ontology graph has 1000 subjects, then the semantic score by applying equation 5 equal 0.83 that means high semantic relation between two subjects (I and J).

$$SS_{in}(\mathrm{i},\mathrm{j}) = 1 - \left(\frac{\max(\log 4, \log 5) - \log 2}{\log 1000 - \min(\log 4, \log 5)}\right)$$
$$= 1 - \left(\frac{\log 5 - \log 2}{\log 1000 - \log 4}\right) = 0.83$$

C. Knowledge Source Phase

Hadoop is also used. It consists of one master machine and six slaves' machines. The integration between HBase and Hadoop archives many advantages as follows [11]:

- Hadoop runs batch jobs using MAP - Reduce technique on distributed machines to achieve maximum throughout and overall performance [13] [15].
- HBase [14] provides good latency, scalability and fault tolerance. Fault tolerance is handled by replication process. Scalability is handled by splitting each table horizontally into regions.
- The integration is done successfully because they use the same underlying HDFS file systems.

DOI: 10.5281/zenodo.5202917
Received: March 17, 2021 Accepted: July 14, 2021

D. Online Phase

This phase consists of two main stages. The first stage is optimizing input query. The second stage is the actual retrieving process by query engine. Query Optimizer phase is implemented. In this stage, stop words removal, stemming and keyword expansion processes are applied to the input query from the user. Wordnet tool is used to expand keywords to extend search domains. The semantic score between extended keywords and actual query's keywords is calculated to generate a key of the input subject. After that, the value of the corresponding key is returned from caching table. Then the query engine is implemented. In this stage, the actual retrieval process is applied by using SPARQL query to return the objects of the corresponding subject value. Query engine is used to generate SPARQL query to retrieve related information from RDF and corresponding source.

5. Experimental Results

A. Environmental Setup

The proposed framework is evaluated using cluster of 1 (head machine node) and 6 (slave machines). The head machine has Intel corei-7 processor@2.6GHz with 16GB RAM. The slave machines have Intel corei-3 processor@2.2GHz with 4GB RAM for each. Hadoop version 2.9.2 was used as open source tool. Hadoop was run with three workers per MAP operation and two workers per REDUCE operation. Also, Apache HBase version 2.2.0 was used. Multiple parallel slave workers are used for the execution of ETI jobs. Figure 6 shows the average ETI execution time for different workers. According to this figure, seven threads were used because they achieved the minimum execution time.

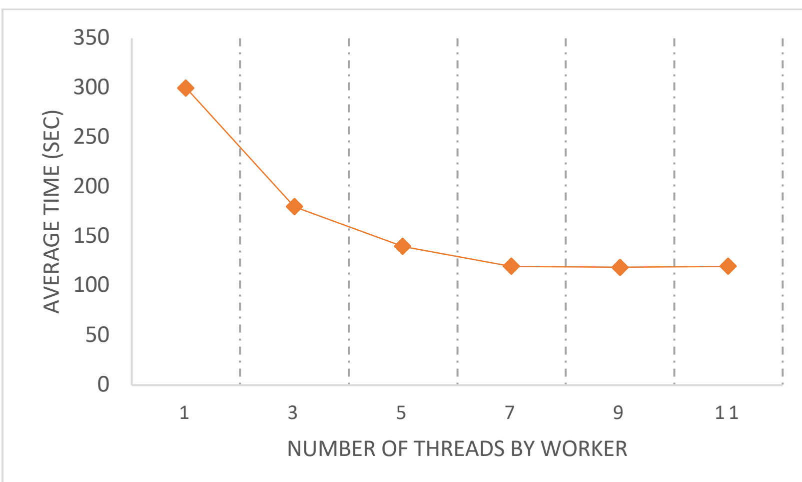

Figure 6. Average Execution Time of ETI Job from Different Source to JSON format

B. Datasets Description

The proposed framework was evaluated using two real world datasets. In table 1 shows some statistical information of these datasets. The two real-world datasets are:

- DBpedia is the most prominent real-world dataset that used in semantic web research community.

DOI: 10.5281/zenodo.5202917

Received: March 17, 2021 Accepted: July 14, 2021

The version of YAGO that used to evaluate our framework is 2016-04.

- YAGO is another great real-world dataset that has been developed at Max Planck institute. The version of YAGO that used to evaluate our framework is 2015-03.

Table 1. Brief statistical information of DBpedia and YAGO datasets.

	DBpedia	YAGO
# Entities in persons Domain	1.5 M	1.3 M
# Entities in work Domain	490 K	510 K
# Entities in organization Domain	275 K	350 K
# Entities in places Domain	810 K	1.2 M
# Entities in biology Domain	301 K	150 K
# N-Quads used	150 M	120 M

C. *Offline Phase Performance Measurements*

The proposed framework is evaluated using many nodes per cluster as shown in figure 7. According to figure 7, if the number of nodes per cluster was increased, the Loading time was decreased for each dataset. So, six nodes per cluster in addition to one master node for that cluster was used in the evaluation process.

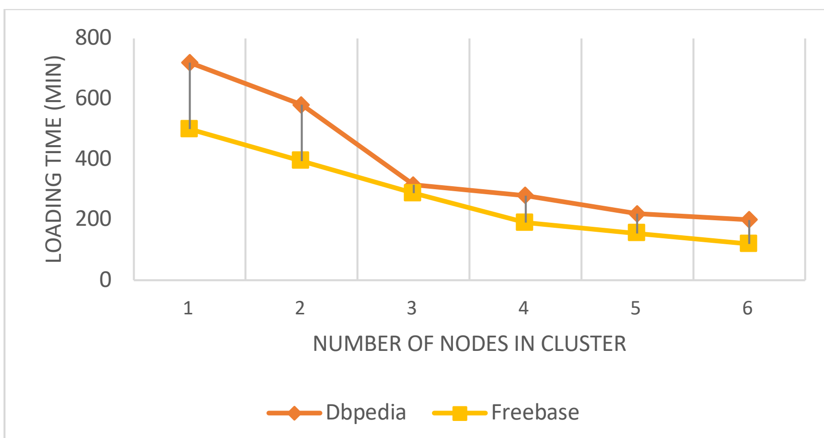

Figure 7. Loading time per each cluster.

D. *Online Phase and Query Performance Measurements*

To see if the proposed methods improve the results, we should test different queries in length using different technologies. The implemented four different queries will be conducted as follows:

DOI: 10.5281/zenodo.5202917
Received: March 17, 2021 Accepted: July 14, 2021

1) Query I: In this test case, the system is triggered by a simple unclear single word as query (e.g. network).

2) Query II: In this query, the system is triggered by the same query as in query I (e.g. networks). But this query expanded using word net to extend search area.

3) Query III: In this query, the system is triggered by complex multiple words as query (e.g. computer networks and protocols). Also, in this test case, we will apply all different proposed techniques.

4) Query IV: In this query, the system is triggered by the same query as in query III (e.g. computer networks and protocols). But this query expanded using word net to extend search area.

Table 2. Result of different online queries.

	Query I	Query II	Query III	Query IV
Input Keywords	1	1	1	1
Extended Keywords	8	0	8	0
True Positive	8690	6481	8630	4991
False Positive	302	430	362	1120
Precision	0.966	0.937	0.959	0.889
Recall	0.938	0.943	0.921	0.957
F-score	0.952	0.935	0.94	0.921
Response Time at online Phase	2.57 Sec	1.22 Sec	3.49 Sec	5.4 Sec

E. Comparisons between proposed system and other related systems

Table 3. Compassion between proposed system and other related systems

	Swoogle	Falcon	Proposed
Support Ontology	√	√	√
Semantic Relations	√	√	√
Support Ranking	×	√	√
Handle Big Data	×	×	√
High Precision	√	×	√

DOI: 10.5281/zenodo.5202917
Received: March 17, 2021 Accepted: July 14, 2021

High Recall	×	√	×
Response Time	Slow	Slow	Medium

6. Conclusion

This manuscript proposes semantic retrieval framework for enhancing knowledge and search area using the integration between more than one semantic source. Propose framework are implemented using some modern technologies such as ontology graph, RDF, MapReduce technique implemented in Hadoop, NoSQL model using Hbase and proposed mathematical model for calculating semantic relations between different subjects. Six nodes with one master node are used to evaluate framework. Two datasets (e.g. DBpedia and YAGO) are used to build knowledge. Experiments with four queries are conducted to evaluate the proposed framework. Experimental results showed high precision and accuracy of our tested queries in good response time comparing to related systems.

References

1. C. Gavankar, T. Bhosale, D. Gunda, A. Chavan and S. Hassan, "A Comparative Study of Semantic Search Systems," 2020 International Conference on Computer Communication and Informatics (ICCCI), 2020, pp. 1-7
2. Z. Pan, "Optimization of Information Retrieval Algorithm for Digital Library Based on Semantic Search Engine," 2020 International Conference on Computer Engineering and Application (ICCEA), 2020, pp. 364-367
3. A. Dramilio, C. Faustine, S. Sanjaya and B. Soewito, "The Effect and Technique in Search Engine Optimization," 2020 International Conference on Information Management and Technology (ICIMTech), 2020, pp. 348-353
4. A. Nadeem, M. Hussain and A. Iftikhar, "New Technique to Rank Without Off Page Search Engine Optimization," 2020 IEEE 23rd International Multitopic Conference (INMIC), 2020, pp. 1-6
5. M. M. El-Gayar, N. E. Mekky, A. Atwan and H. Soliman, "Enhanced Search Engine Using Proposed Framework and Ranking Algorithm Based on Semantic Relations," in International journal of simulation: systems, science & technology, vol. 10, 2019
6. M. M. El-Gayar, N. E. Mekky, A. Atwan and H. Soliman, " A Novel Knowledge-based Semantic Search Engine," in IEEE Access, vol. 7, pp. 139337-139349, 2019
7. M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R.Wald, and E. Muharemagic, “Deep learning applications and challenges in big data analytics,” J. Big Data, vol. 2, no. 1, Dec.2015.
8. Y. S. Negi and S. Kumar, “A comparative analysis of keywordand semantic-based search engines,” in Advances in Intelligent Systems and Computing, 2014, vol. 243, pp. 727–736.

DOI: 10.5281/zenodo.5202917
Received: March 17, 2021 Accepted: July 14, 2021

9. J. M. Kassim and M. Rahmany, "Introduction to semantic search engine," in Proceedings of the 2009 International Conference on Electrical Engineering and Informatics, ICEEI 2009, 2009, vol. 2, pp. 380–386.
10. A. Begdouri, O. Chergui, and D. Leclet-Groux, "A knowledge-based approach for keywords modeling into a semantic graph," 2018. "OWL," in Semantic Web: Concepts, Technologies and Applications, London: Springer London, pp. 81–103.
11. M. N. Asim, M. Wasim, M. U. G. Khan, N. Mahmood, and W. Mahmood, "The Use of Ontology in Retrieval: A Study on Textual, Multilingual, and Multimedia Retrieval," IEEE Access, vol. 7, pp. 21662–21686, 2019.
12. B. R. Prasad and S. Agarwal, "Comparative Study of Big Data Computing and Storage Tools: A Review," Int. J. Database Theory Appl., vol. 9, no. 1, pp. 45–66, Jan. 2016.
13. A. Oussous, F. Z. Benjelloun, A. Ait Lahcen, and S. Belfkih, "Big Data technologies: A survey," Journal of King Saud University - Computer and Information Sciences, vol. 30, no. 4. King Saud bin Abdulaziz University, pp. 431–448, 01-Oct-2018.
14. D. Vohra and D. Vohra, "Using Apache HBase," in Pro Docker, Apress, 2016, pp. 141–150.
15. K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The Hadoop distributed file system," in 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies, MSST2010, 2010.
16. S. Shahrivari, "Beyond batch processing: Towards real-time and streaming big data," Computers, vol. 3, no. 4. MDPI AG, pp. 117– 129, 01-Dec-2014.
17. D. Singh and C. K. Reddy, "A survey on platforms for big data analytics," J. Big Data, vol. 2, no. 1, Dec. 2015.
18. V. Bevilacqua, V. Santarcangelo, A. Magarelli, A. Bianco, G. Mastronardi, and E. Cascini, "A semantic search framework for document retrievals (literature, art and history) based on Thesaurus multi wordnet," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2011, vol. 6838 LNCS, pp. 456–463.
19. Y. Qu and G. Cheng, "Falcons concept search: A practical search engine for web ontologies," IEEE Trans. Syst. Man, Cybern. Part Systems Humans, vol. 41, no. 4, pp. 810–816, Jul. 2011.
20. L. Ding et al., "Swoogle Bibliographic Search," Proceedings of the Thirteenth ACM conference on Information and knowledge management - CIKM '04. ACM Press, New York, New York, USA, p. 652, 2004.
21. 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," J. Web Semant., vol. 9, no. 4, pp. 365–401, Dec. 2011.
22. S. S. Laddha and P. M. Jawandhiya, "Semantic tourism information retrieval interface," in 2017 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2017, 2017, vol. 2017–January, pp. 694–697.
23. A. Fatima, C. Luca, and G. Wilson, "New framework for semantic search engine," in Proceedings - UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, UKSim 2014, 2014, pp. 446–451.
24. M. Färber, F. Bartscherer, C. Menne, and A. Rettinger, "Linked data quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO," Semant. Web, vol. 9, no. 1, pp. 77–129, 2018.

DOI: 10.5281/zenodo.5202917

Received: March 17, 2021 Accepted: July 14, 2021

DOI: 10.5281/zenodo.5202917
Received: March 17, 2021 Accepted: July 14, 2021