*Journal of Universal Computer Science, vol. 29, no. 3 (2023), 222-241*
*submitted: 14/5/2022, accepted: 15/12/2022, appeared: 28/3/2023 CC BY-ND 4.0*

# Leveraging Structural and Semantic Measures for JSON Document Clustering

**Uma Priya D**
(National Institute of Technology Karnataka, Mangalore, India
https://orcid.org/ 0000-0001-6090-3453, umapriya.dharmaraj@gmail.com)

**P. Santhi Thilagam**
(National Institute of Technology Karnataka, Mangalore, India
https://orcid.org/ 0000-0002-8359-1330, santhi@nitk.edu.in)

**Abstract:** In recent years, the increased use of smart devices and digital business opportunities has generated massive heterogeneous JSON data daily, making efficient data storage and management more difficult. Existing research uses different similarity metrics and clusters the documents to support the above tasks effectively. However, extant approaches have focused on either structural or semantic similarity of schemas. As JSON documents are application-specific, differently annotated JSON schemas are not only structurally heterogeneous but also differ by the context of the JSON attributes. Therefore, there is a need to consider the structural, semantic, and contextual properties of JSON schemas to perform meaningful clustering of JSON documents. This work proposes an approach to cluster heterogeneous JSON documents using the similarity fusion method. The similarity fusion matrix is constructed using structural, semantic, and contextual measures of JSON schemas. The experimental results demonstrate that the proposed approach outperforms the existing approaches significantly.

**Keywords:** JSON, Clustering, Data Mining, Similarity Measures
**Categories:** H.3.1, H.3.2, H.3.3, H.3.7, H.5.1
**DOI:** 10.3897/jucs.86563

## 1 Introduction

Over the past decade, Web applications have adopted JavaScript Object Notation (JSON) format as the primary standard for data interchange between servers and clients. Most e-commerce applications and Application Programming Interfaces (API) use JSON format to generate varied structured data. NoSQL document-oriented databases provide an efficient architecture in storing and managing the varied structured JSON data effectively. JSON documents are self-describing, i.e., the structure is embedded with the data. A JSON document or an object is represented as key-value pair where keys are of string type, and values are of simple and complex types such as string, number, Boolean, array, and object. The heterogeneous and dynamic nature of JSON documents increases the complexity of analysing the documents for efficient data retrieval, integration, and so on. Therefore, there is a demand for an efficient way of organizing the JSON documents to support the above tasks effectively.

Clustering is the process of dividing documents into groups or clusters that are similar to each other and different from other clusters. Successful clustering algorithms rely on data representations and similarity measures. In the case of hierarchical data like JSON, where the schema is embedded with the data, the similarity metrics to cluster the documents can be applied on: (i) structure-only, (ii) content-only, and (iii) both structure and content. This paper focuses on clustering the JSON documents based on schemas.

Most literature on clustering hierarchical data focuses on eXtensible Markup Language (XML) data [Piernik et al., 2016, Costa and Ortale, 2017]. Even though the JSON format received much attention for storing data from large-scale applications, the research on clustering JSON documents is still in its early stages. The heterogeneous nature of JSON format allows documents to have different schemas in a collection. Researchers have used structural similarity measures and identified the equivalent schemas to generate the global schema that support various tasks such as query formulation and data retrieval.

$D_1$

```
{
”author”:[{
”firstName”: ”John”,
”lastName” : ”Smith”
},
{
”firstName”: ”Noam”,
”lastName” : ”Zeilberger”
}],
”book”:{
”title”: ”Trademaker”,
”year” : 2014},
”publisher”: ”Science of Science”
}
```

$D_2$

```
{
”author”: ”Torwick”,
”booktitle”: ”Ethnea”,
”year” : 2016,
”issn”: 976−93−87654−01−1
}
```

$D_3$

```
{
”author”: ”Lotfi”,
”headline”: ”tomaso@xyz.com”,
”dated”: ”Journal of Computers”}
```

*Figure 1: Sample collection of JSON documents*

Extant approaches on JSON data identify similar schemas based on structural similarity [Wang et al., 2015, Gallinucci et al., 2019, Bawakid, 2019]. The semantic similarity of schemas is identified with the help of external knowledge bases such as WordNet [Miller, 1998]. The meaning of the JSON attributes is used to identify the semantic similarity of schemas. However, the limitation of this technique is that the knowledge bases must be updated to support current concepts. As JSON documents are application-specific and hierarchical, the JSON schemas generated by the applications must also preserve the ancestor-descendant (A-D) and parent-child (P-C) relationship of attributes. Therefore, apart from traditional structural and semantic similarity, finding the similarity based on the context hidden in the schemas is needed to perform clustering efficiently.

In order to address the issues mentioned above, various approaches have used deep neural language models that identify both the syntactic and semantic information of words in a document. The language models generate a low-dimensional dense representation of vectors for each word in a document. While the traditional models generate a unique vector for each word in a document without considering the hidden context, the deep neural language models generate vectors based on the context. Therefore, the vector representations or embeddings generated are not unique for a word. However, literature has shown that these models have been applied to unstructured data where the input is a paragraph or a sentence [Uma Priya et al., 2020]. Therefore, the research on using deep neural network models for hierarchical data is in the preliminary stage.

**Motivation:** Consider the sample collection of JSON documents $D = \{D_1, D_2, D_3\}$ shown in Figure 1. The documents are different in structure. The schema of these documents says $S = \{S_1, S_2, S_3\}$ has different attributes. JSON schemas are not only structurally heterogeneous but also semantically heterogeneous. Therefore, considering only structural similarity in these documents results in three different clusters. In addition to structural similarity, traditional semantic similarity using external knowledge bases also plays a primary role in improving the clustering quality. In Figure 1, the meaning of the attributes in $S_2$ and $S_3$ are similar, i.e., the *headline* is related to *booktitle*, *dated* is associated with the *year*. However, the attribute information in these schemas carries some context related to the article scenario. The article can be a *conference* or *journal*, or *news article*. Looking into the context of these schemas, $S_1$ and $S_2$ belong to the *journal* article, whereas $S_3$ belongs to the *news article*. Therefore, apart from structural similarity, finding the contextual and semantic similarity of schemas results in better clustering.

**Contributions:** While most literature focuses either on structural or semantic similarity measures to cluster the documents, this work partitions the JSON document collection into clusters based on the structural, contextual, and semantic similarity of schemas. This work uses a Robustly Optimized BERT Pretraining Approach (RoBERTa) [Liu et al., 2019] to generate schema embeddings for identifying contextual similarity.

The major contributions are:

1. Extracting JSON schemas from documents and analyzing them to determine the structural, semantic, and contextual similarities
2. Clustering JSON documents using the similarity fusion method
3. Evaluating the performance of the proposed approach using real and synthetic datasets

The rest of the paper is structured as follows. The following section reviews the related works of XML and JSON similarity approaches. We describe the proposed approach with an example in section 3. Section 4 presents and discusses the experimental results of the proposed and existing approaches. The findings and future work are summarized in section 5.

## 2 Related Work

Literature has seen numerous approaches for clustering XML data. However, JSON document clustering has received less attention from researchers. This section summarizes the related works on the structural, contextual, and semantic similarity of XML and JSON documents used in several uses like clustering, schema matching, etc. Figure 2 illustrates the related works on similarity approaches to XML and JSON data.

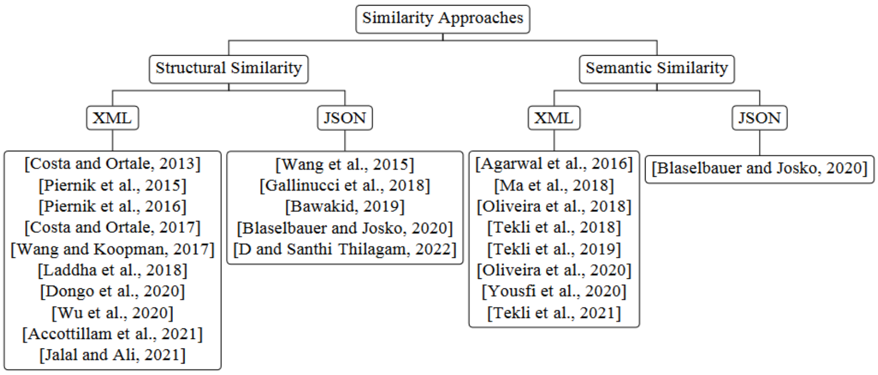

*Figure 2: Related works on Similarity Approaches for XML and JSON Documents*

## 2.1 XML Similarity Approaches

Wang et al. [Wang and Koopman, 2017] determined the similarity of journal articles based solely on the entities that were connected to those articles. But they use XML content rather than structure to determine the context. In addition, their models suffer from large dimensions as they use traditional semantic representations. Laddha et al. [Laddha et al., 2018] modeled the structure and content of the semi-structured documents in a shared vector space, which allowed them to capture the semantics of the texts. Hence, the semantics of the document as a whole has been extracted. Costa et al. [Costa and Ortale, 2019] suggested a method to separate the XML documents using their topical similarity. Dongo et al. [Dongo et al. 2020] presented an approach for determining semantically similar XML documents based not only on the structure of the documents but also on the content of the documents. Wu et al. [Wu et al. 2020] used dense clustering to group semantically similar structures in the registers such as Novel, News, and Interview. This allowed them to find groups of structures that were semantically related. Jalal et al. [Jalal and Ali, 2021] group similar papers from the same magazine that are related based on the content. However, they clustered the articles using the Term Frequency-Inverse Document Frequency (TF-IDF) and the cosine similarity of XML attributes.

## 2.2 JSON Similarity Approaches

The majority of studies on JSON data investigate the similarities of JSON schemas by comparing the names and types of attributes. The skeleton schema concept was suggested by Wang et al. [Wang et al., 2015] to group together identical schemas and produce a summarized representation of different schema types. Gallinucci et al. [Gallinucci et al., 2018] presented Build Schema Profile (BSP), which uses association rules to classify the schema variants. JSONGlue [Blaselbauer and Josko, 2020] calculates the degree of similarity between schemas by combining semantic, linguistic, and instance-level approaches. D, U. P., and Santhi Thilagam, P. (2022) [D and Santhi Thilagam, 2022] used the TF-IDF approach to cluster the JSON documents.

### 2.3 Advanced Language Models

Neural language models that generate word and document embeddings have become increasingly popular in recent years. Numerous research efforts have suggested enhancing distributed representations of documents/sentences [Le and Mikolov, 2014, Dai et al., 2015, Kiros et al., 2015, Hill et al., 2016, Pagliardini et al., 2017, Logeswaran and Lee, 2018, Gupta et al., 2019, Cer et al., 2018, Sinoara et al., 2019]. These methods efficiently calculate the semantic similarity of words by capturing the similarity of the respective vector representations. Very few research works [Hammad et al., 2020, Farouk, 2020] focused on combining traditional semantic models with word embedding techniques to capture more semantics on data. However, using a vector space to analyze JSON data is in its early stages.

**Research Gaps:** A substantial amount of research works for clustering XML documents have been presented in the literature. Yet, extant approaches fall short in identifying semantic as well as contextual similarities in JSON documents. Similarly, the existing literature on JSON data focused on structural or semantic measures alone to identify the similarity of schemas. As a result, there is a need to determine the hidden semantic relatedness in the schema by focusing not only on semantic similarity but also on contextual similarity as well.

## 3 Proposed Approach

Given a JSON documents collection $D = \{D_1, D_2, ..., D_n\}$ and $D_i=\{A_1, A_2, ..., A_m\}$ where $n$ and $m$ denote the number of documents in a collection $D$ and the attributes in a document $i$, the goal of this work is to divide $D$ into mutually exclusive $K$ clusters, say $C_{k,}$ where each $C_t \in C$ contains both structurally and semantically similar JSON documents.

To address the above issue, this paper focuses on meeting the following objectives:

1. To explore the contextually similar JSON schemas available in a collection
2. To design the similarity fusion method that captures the structural, semantic, and contextual properties of JSON schemas
3. To cluster the JSON documents using the similarity fusion matrix

The proposed approach is depicted diagrammatically in Figure 3. Algorithm 1 on page 9 describes the proposed workflow in detail.

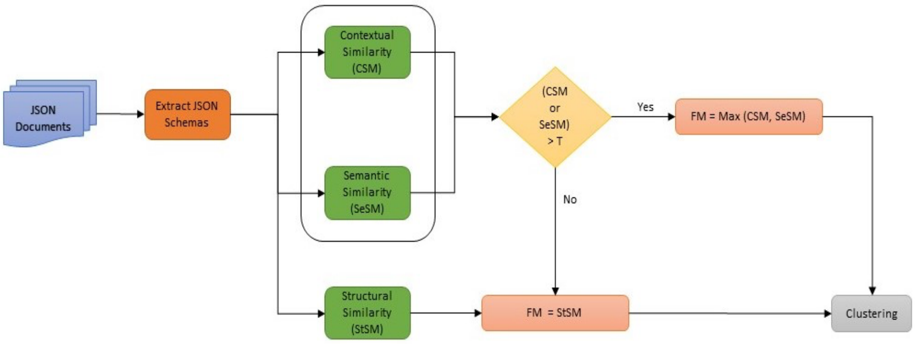

*Figure 3: Flow description of the proposed approach*

### 3.1 Extraction of JSON Schemas

A JSON document can be expressed in a tree format where the JSON tree is defined as a 3-tuple $T = (R, V, E)$. $R$ represents the root of a tree; the attributes are represented as vertices (nodes) $V$, and the P-C relationship between the attributes is defined using edges $E$. The attribute value denotes the data types such as number, string, Boolean, arrays, and nesting objects. Arrays and object types are represented as sub-trees in T.

In general, when the schema is embedded with the data, the reverse engineering method is used to extract the schema. Therefore, the JSON document is parsed in depth-first order, i.e., the parsing process starts from a root node and then the first child of a root node, followed by its children in left-to-right order. This way of parsing the document preserves the P-C relationship of JSON tree nodes, i.e., the attribute names and their parent names are concatenated. To distinguish the representation of arrays and nesting objects, different concatenating operators such as "#" and "." are used because arrays are ordered lists of values, and nesting objects are unordered. Arrays may include a sequence of ordered objects whose order must be retained to capture the array's whole structure. This work represents JSON schemas as root-to-leaf paths.

### 3.2 Calculation of Similarity

As JSON schemas are dynamic and heterogeneous, using a single similarity metric to determine the clusters is not informative and complete. Clustering quality can be improved by using the semantic and syntactic properties of JSON schemas while identifying similar documents. The flow of similarity computation is briefly described as follows:

#### 3.2.1 Contextual Similarity

The proposed work identifies the contextual similarity of JSON schemas using the RoBERTa model. The vector associated with each schema describes how it occurs in context with other schemas. For example, the vectors of $S_1$ and $S_2$ are similar not because the attributes *"author"* and *"year"* are present in them. The reason is that the surrounding attributes of an attribute *"year"* in $S_1$ and $S_2$ share similar meanings. Since RoBERTa is a distributed language model, the vectors generated are low-dimensional and dense vectors. After generating the schema embeddings or vectors for each schema, the cosine similarity measure is used to determine the contextual similarity.

Initially, the schemas are tokenized by the WordPiece model. The text representation obtained for a schema with *m* attributes is *m* numeric vectors of length 1024 (dimension). Since this work uses the pre-trained RoBERTa model to generate schema embeddings, this work retains the dimension of the model used for training. Therefore, the output vector *P* of all the attributes in a schema is placed into a matrix of size $m$*1024.

Given a contextual embedding *P* of D, the cosine similarity measure for any two schemas $S_i$ and $S_j$ are formally measured as follows:

$$\text{cosine}\left(S_i, S_j\right) = \frac{s_i^T s_j}{||s_i||\,||s_j||} \quad (1)$$

The contextual similarity matrix *CSM is* calculated as

$$CSM[S_i]\left[S_j\right] = \text{cosine}\left(\, S_{i,} S_j\right) \;\; \forall i, j \leq n \quad (2)$$

### 3.2.2 Semantic Similarity

The semantic heterogeneity of schemas describes that they are structurally different but carry the same semantic information. This semantic information can be obtained by external knowledge bases such as DBPedia and WordNet, which construct the semantic network for each attribute in a schema. In addition to contextual similarity, semantic similarity also plays a significant role in determining the performance of clustering because the synonyms of attributes enrich the meaning of attributes and identifies the closest schemas using the semantic network. The semantic similarity for any two schemas $S_i$ and $S_j$ is formally measured using the Wu-Palmer similarity measure [Wu and Palmer, 1994] as follows:

$$SeSM[S_i]\left[S_j\right] = wup\_similarity\ (S_i, S_j) \quad (3)$$

If the similarity score between two schemas is below the threshold, they are dissimilar.

### 3.2.3 Structural Similarity

Structural similarity is a metric to determine similar documents in terms of their syntactic properties. In this work, the structural similarity of the two schemas is identified by comparing their P-C and A-D relationship. Since the flattened attributes are considered to measure the similarity, the frequency of flattened attributes determines the similarity score. In order to accomplish this task, this work calculates the similarity between any two schemas using cosine measure. The cosine similarity for any two schemas is calculated as given in equation 2. However, the input for finding the cosine similarity in this section differs from section 3.2.1.

Given the set of schemas *S*, this section aims to construct the schema-attribute matrix for each attribute from schemas. The schema-attribute matrix gives the frequency of each attribute in a schema which helps to identify the cosine similarity. Because the cosine similarity score varies depends on the frequency of attributes in each schema. The reason behind choosing the cosine similarity measure is that JSON schemas may have the same attributes present in different levels of schemas which contributes to the structural heterogeneity of schemas, i.e., although the attribute name is the same in different levels of a schema, if the parent or child of the attribute is different, they are heterogeneous by structure. In order to preserve this nature, cosine similarity is preferred over other similarity measures like Jaccard similarity because the cosine similarity measure considers the duplication of attributes in a schema and updates the score, which is not the case with Jaccard similarity.

### 3.2.4 Fusion of Similarity Measures

Similarity fusion aims to integrate the high contextual, semantic and structural scores to enrich the similarity score of the schemas. This is achieved in the following steps:

1. Compare the contextual and semantic scores of any two schemas, say $S_i$ and $S_j$, and get the maximum score, say $m_s$.
2. If $m_s > T$, then include them in Fusion Matrix FM.
3. If $m_s < T$, then find the structural similarity for the pair $(S_i, S_j)$.
   Therefore, FM is calculated using the following equation

$$FM[S_i][S_j] = \begin{cases} Max\left(CSM[S_i][S_j], SeSM[S_i][S_j]\right), if\ (CSM[S_i][S_j]\ or\ SeSM[S_i][S_j]) > T \\ StSM[S_i][S_j], otherwise \end{cases}$$

The major reason behind this kind of similarity fusion method is that if the structural similarity is calculated before semantic and contextual similarities, the search space for contextual and semantic similarities will be reduced. In addition, the clusters based on structural similarity lose the meaning of schemas in a whole JSON document collection. Therefore, identifying contextual and semantic similarity prior to structural similarity helps capture more semantics and preserve the structural properties.

## 3.3 Clustering

Given the schemas $S = \{S_1, S_2, ..., S_n\}$ and the similarity fusion matrix *FM*, this work aims to cluster the JSON documents $D = \{D_1, D_2, ..., D_n\}$ using the clustering algorithm. In this work, the similarity matrix *FM* determines the clustering quality.

This work uses a spectral clustering algorithm [Von Luxburg, 2007] to cluster the documents. Now, the problem of clustering can be restated as a graph U$G = (V, E)$ where $V = \{V_1, V_2, ..., V_l\}$, and *E* denotes the edge between any two vertices in *V* and the weight for an edge is assigned using $S_{ij} \in FM$. The Degree matrix $D'$ is a diagonal matrix designated as termed as $\{d'_1, d'_2, ..., d'_n\}$ with degree $d'_i$ defined as

$$d'_i = \sum_{j=1}^{n} S_{ij} \qquad (4)$$

The normalized Laplacian matrix $L$ is defined as

$$L = D'^{\frac{-1}{2}}(D' - FM)D'^{\frac{-1}{2}} \qquad (5)$$

**Algorithm 1:** Clustering JSON Documents using Similarity Fusion Matrix

```
Input: JSON data collection $D = \{D_1, D_2, ..., D_n\}$, Output: $C = \{C_1, C_2, ..., C_k\}$
Start
    1.  Extract the schemas S = $\{S_1, S_2, ..., S_n\}$ from $D$
    2.  schema_embed = RoBERTa ($S$, dimension)
    3.  for each $(S_i, S_j) \in$ schema_embed do
    4.  $CSM[S_i][S_j] = cosine(S_i, S_j)$
    5.  end for
    6.  find synsets for each $A_i \in A$
    7.  for each $(S_i, S_j) \in S$ do
    8.  $SeSM[S_i][S_j] = wup\_similarity(S_i, S_j)$
    9.  $StSM[S_i][S_j] = cosine(S_i, S_j)$
    10. end for
    11. if $(CSM[S_i][S_j]$ or $SeSM[S_i][S_j]) > T$ then
    12. $FM[S_i][S_j] = Max(CSM[S_i][S_j], SeSM[S_i][S_j])$
    13. else
    14. $FM[S_i][S_j] = StSM[S_i][S_j]$
    15. end if
    16. Cluster the documents $D$ into $K$ clusters with the pair-wise similarity
        matrix $FM$
End
```

Calculate the first k's eigenvectors $U = \{U_1, U_2, ..., U_k\}$ associated with $L$ given that $U \in R^k$ and the columns of $U$ are denoted by vectors. The ith row of $U$ is denoted as $y_i \in R^k$. K-Means clustering is used to cluster the data points $\{y_1, y_2, ..., y_n\}$.

Algorithm 1 is explained with an example. Considering the JSON sample collection given in Figure 1, the schemas $S = \{S_1, S_2, S_3\}$ = *{(author.firstName author.lastName book.title book.year publisher), (author booktitle year issn), (author headline dated)}* are extracted (line 1). The contextual similarity matrix (line 4) $CSM \in R^{3\times3}$ for the schemas $S = \{S_1, S_2, S_3\}$ is calculated as

$$CSM = \begin{pmatrix} 1.0 & 0.57 & 0.54 \\ 0.57 & 1.0 & 0.49 \\ 0.54 & 0.49 & 1.0 \end{pmatrix}$$

The semantic similarity matrix (line 8) $SeSM \in R^{3\times3}$ for the schemas $S$ is calculated as

$$\text{SeSM} = \begin{pmatrix} 1.0 & 0.54 & 0.08 \\ 0.54 & 1.0 & 0.1 \\ 0.08 & 0.1 & 1.0 \end{pmatrix}$$

The structural similarity matrix (line 9) $StSM \in R^{3\times3}$ for the schemas $S$ is calculated as

$$\text{StSM} = \begin{pmatrix} 1.0 & 0.41 & 0.32 \\ 0.41 & 1.0 & 0.28 \\ 0.32 & 0.28 & 1.0 \end{pmatrix}$$

The fusion similarity matrix *FM* (lines 11 to 15) from all the above similarity matrices is calculated $FM \in R^{3\times3}$ for the schemas $S = \{S_1, S_2, S_3\}$ using *CSM, SeSM*, and *StSM* are calculated as

$$\text{FM} = \begin{pmatrix} 1.0 & 0.57 & 0.54 \\ 0.57 & 1.0 & 0.28 \\ 0.54 & 0.28 & 1.0 \end{pmatrix}$$

In this example, the threshold value is set as 0.5 because of fewer documents. After applying the fusion matrix to the clustering algorithm, the clusters are identified as [0,0,1]. $D_1$ and $D_2$ belong to $cluster_0$, and $D_3$ belongs to $cluster_1$. This result shows that the proposed approach efficiently calculates the similarities, and the clusters are formed efficiently.

## 4 Experimental Evaluation

The proposed approach has been evaluated for two datasets, DBLP [Mohamed L. Chouder and Stefano Rizzi and Rachid Chalal, 2017] and the synthetic dataset (SD). The DBLP dataset comprises 2,00,000 documents randomly picked from 20,00,000 documents of publication scenario. The SD[1] is generated with 50,000 documents for the publication scenario. Both datasets may have some common attributes as they are part of the publication scenario. Both datasets together have 76 attributes and 13 classes such as conference, journal, book, and so on. Hence, the number of clusters is decided as 13 for evaluating the existing and proposed approaches.

### 4.1 Evaluation Measures

Typically, intrinsic and extrinsic measures are employed to evaluate the effectiveness of document clustering algorithms. This paper uses Silhouette Co-efficient (SC) to

[1] https://github.com/umagourish/Synthetic-Datasets

validate the cluster. The most common extrinsic measures, including recall and precision, rely on the order of cluster labels to ground truth labels that are problematic for various labels. In this instance, the measures such as Normalized Mutual Information (NMI) score, Adjusted Rand Index (ARI) score, and Adjusted Mutual Information (AMI) scores are preferable as they are not affected by the absolute label values [Vinh et al., 2010].

#### 4.1.1 External Metrics

*NMI:*

Two labels on the same dataset can be compared using a metric called mutual information (MI). With known ground truth labels, MI describes the drop in the class labels entropy. The MI of class labels A and cluster labels B, MI (A, B) is computed as

$$MI\ (A, B) = \sum_{a\, \in A} \sum_{b \in B} \log\left(\frac{p(a,b)}{p(a)p(b)}\right) \qquad (6)$$

NMI normalizes MI to a range of 0 to 1, with 0 indicating no mutual information and 1 indicating agreement. NMI is computed as follows:

$$NMI\ (A, B) = \frac{MI\ (A,B)}{\sqrt{E(A)E(B)}} \qquad (7)$$

where E(A) and E(B) are marginal entropies, and they are computed as:

$$E(A) = -\sum_{i=1}^{n} p(a_i) \log p(a_i) \qquad (8)$$

*ARI:*

ARI is computed by adjusting the Rand Index (RI) by its expected value as follows:

$$ARI\ (A, B) = \frac{RI(A,B) - E\{RI(A,B)\}}{\max\{RI(A,B)\} - E\{RI(A,B)\}} \qquad (9)$$

where RI (A, B) is the random index for any two clusters A, and B [Yeung et al., 2001], which takes a value ranging from 0 to 1. 1 represents identical clusters, and 0 is non-identical. V {RI (A, B)} represents the expected value of RI.

*AMI:*

AMI changes the MI based on its expected value, similar to ARI. AMI values range from 0 to 1, with 1 signifying an identical cluster and adjusting for cluster count. AMI is determined by

$$AMI\ (A,B) = \frac{MI(A,B) - E\{MI(A,B)\}}{\max\{E(A),E(B)\} - V\{MI(A,B)\}} \tag{10}$$

### 4.1.2 Internal Metrics

Silhouette Co-efficient is ideal for estimating the clustering efficiency for unlabelled data, which is obtained by

$$SC = \frac{n-m}{\max\{m,n\}} \tag{11}$$

where *m* and *n* represent the mean distance for a document from other documents in the same cluster and in a different cluster, respectively. The value of *SC* ranges from -1 to 1. If samples from the same cluster are closer and those from different clusters are far apart, the score will be high.

## 4.2 Existing Approaches for Comparison

The proposed approach is compared with contextual, structure-only, and semantic-only approaches to demonstrate the impact of merging all similarities. The proposed approach has been compared with state-of-the-art language models such as InferSent [Conneau et al., 2017], Universal Sentence Encoder (USE) [Cer et al., 2018], and Embeddings for Language Models (ELMo) [Peters et al., 2018].

**Structure-only approach:** Bawakid [Bawakid, 2019] discovered the schema variants in a collection by clustering the documents based on how similar their structures were. TF-IDF approach was used to determine the frequency of attributes and performed clustering.

**Semantic-only approach:** The JSONGlue [Blaselbauer and Josko, 2020] approach used WordNet to determine the synonyms of the attributes. The semantic similarity of the schemas is identified to support JSON schema matching. In order to have a fair comparison, JSONGlue has been optimized to compare JSON schemas with a threshold of 0.6 to group the documents.

**Neural language models for Contextual Similarity:** In the case of contextual similarity, pre-trained sentence encoders like USE, InferSent, and ELMo encode the attributes into deep contextualized sentence embeddings. The cosine similarity of these embeddings is calculated and clustered in the documents.

### 4.3 Results and Discussion

To examine the benefit of combining all the similarities in the proposed work, the result outcomes have been studied across multiple dimensions. The similarities of JSON schemas are identified using node-based similarity and path-based similarity. The sample schemas given in Tables 1 and 2 are part of a dataset. The similarity scores mentioned in the respective tables depend on the whole dataset. The sample schemas and their scores are given to better understand the different cases considered to evaluate the proposed approach. For node-based similarity, the JSON schemas are represented as a set of nodes where the nodes are parsed in the depth-first order of a JSON tree. For path-based similarity, the JSON schemas are represented as a set of root-to-leaf paths. The threshold to determine the semantic scores is fixed as 0.6 based on the preliminary results.

*Node-based Similarity:* There are five schemas considered for evaluation in Table 1. The pair *($S_1$, $S_2$)* are not similar schemas, whereas the pairs *($S_2$, $S_3$)* and *($S_4$, $S_5$)* are similar schemas and hence placed in the same cluster. Table 1 illustrates the node-based similarity of different schemas for the existing and proposed approaches. The score of USE is high in two pairs such as *($S_1$, $S_2$)* and *($S_4$, $S_5$)*. However, *($S_1$, $S_2$)* are not similar schemas. The high value of USE indicates that the pair may belong to the same cluster, whereas other models, such as ELMo and InferSent, have got fewer scores. Out of all the approaches, the proposed approach has got 0.28. Although it is higher than TF-IDF and JSONGlue, 0.28 is less to be considered for clustering. Therefore, the proposed approach is able to identify dissimilar schemas efficiently. Considering the other pairs, the proposed work yielded better results than other approaches and highlighted the importance of fusing all the similarity scores. It is noted from the results that the use of the cosine similarity measure to find the structural similarity in the proposed work is more suitable for node-based similarity because the attributes may be repeated when it is present in more than one nesting level of a document. However, this is reduced in path-based similarity because the attributes are concatenated, and the flattened attributes act as features to find the similarity.

*Table 1: Similarity comparison of the existing approaches and proposed approach for JSON schemas based on attribute nodes*

| **S.No** | **Schemas** | | **TF-IDF [Bawakid, 2019]** | **JSONGlue [Blaselbauer and Josko, 2020]** | **USE [Cer et al., 2018]** | **ELMo [Peters et al., 2018]** | **InferSent [Conneau et al., 2017]** | **Proposed Approach** |
|---|---|---|---|---|---|---|---|---|
| 1 | $S_1$: *paper headline reporter company date* | $S_2$: *paper title authors conferencename year* | 0.18 | 0.14 | 0.48 | 0.31 | 0.35 | 0.28 |
| 2 | $S_2$: *paper title authors conferencename year* | $S_3$: *article title authors author conference name conference year* | 0.16 | 1 | 0.53 | 0.95 | 0.89 | 0.96 |
| 3 | $S_4$: *book author title year url timestamp biburl bibsource* | $S_5$: *book author booktitle year volume number pages doi* | 0.44 | 0.6 | 0.57 | 0.71 | 0.67 | 0.79 |

*Table 2: Similarity comparison of the existing approaches and proposed approach for JSON schemas based on attribute paths*

| S. No | Schemas | | TF-IDF [Bawakid, 2019] | JSONGlue [Blaselbauer and Josko, 2020] | USE [Cer et al., 2018] | ELMo [Peters et al., 2018] | InferSent [Conneau et al., 2017] | Proposed Approach |
|---|---|---|---|---|---|---|---|---|
| 1 | $S_6$:<br>*paper.headline*<br>*paper.reporter*<br>*paper.company*<br>*paper.date* | $S_7$: *paper.title*<br>*paper.authors*<br>*paper.conferencename*<br>*paper.year* | 0 | NA | 0.48 | 0.19 | 0.24 | 0.18 |
| 2 | $S_7$: *paper.title*<br>*paper.authors*<br>*paper.conferencename*<br>*paper.year* | $S_8$: *article.title*<br>*article.authors*<br>*article.author*<br>*article.conference.name*<br>*article.conference.year* | 0 | NA | 0.46 | 0.94 | 0.87 | 0.97 |
| 3 | $S_9$: *book.author*<br>*book.title*<br>*book.year*<br>*book.url*<br>*book.timestamp*<br>*book.biburl*<br>*book.bibsource* | $S_{10}$:*book.author*<br>*book.booktitle*<br>*book.year*<br>*book.volume*<br>*book.number*<br>*book.pages*<br>*book.doi* | NA | NA | 0.57 | 0.69 | 0.64 | 0.81 |

*Path-based Similarity:* Table 2 illustrates the path-based similarity of different pairs of schemas for the existing approaches and proposed approach. It is observed from the results that the TF-IDF score is comparatively less for path-based similarity than node-based similarity because there is less chance for frequent paths in schemas. When the common paths are high, TF-IDF retrieves better results. Although neural models such as ELMo, USE, and InferSent yield a reasonable score, the proposed work gives maximum importance to contextual similarity. Hence, we achieve a better score than all other existing models. When comparing tables 1 and 2, it is observed that structure-only and semantic-only approaches for path-based similarity yield better results than node-based similarity. This is based on the presence of common paths present in a schema. This case illustrates the power of contextual similarity before structural similarity in the proposed work. Hence, finding the similarity of heterogeneous hierarchical structures like JSON documents shows better results when considering structural, semantic, and contextual properties of schemas.

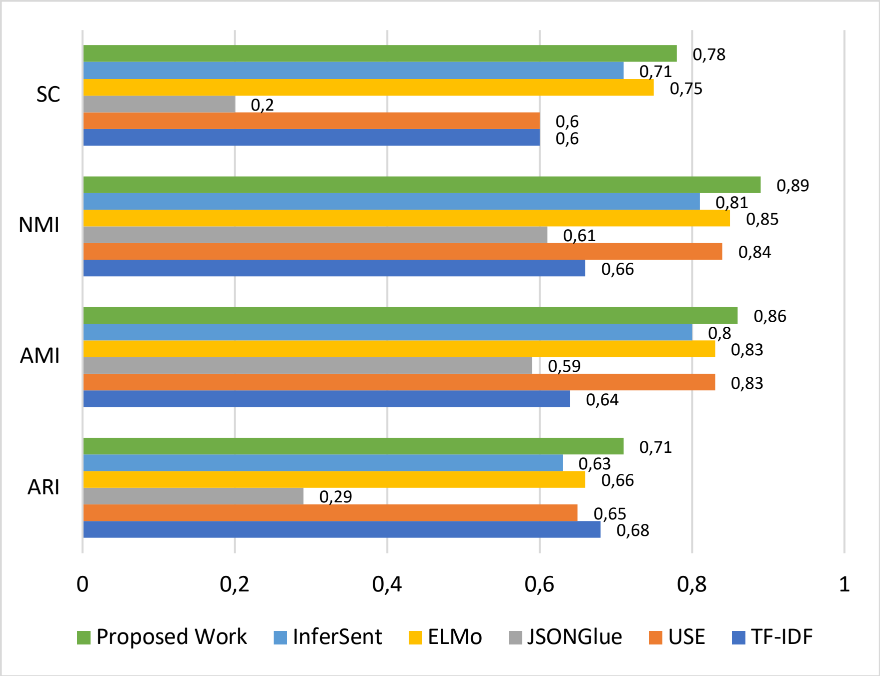

*Figure 4: Comparison of the clustering performance of the proposed work using existing structural, semantic, and contextual similarity approaches*

*Clustering Performance:* It is observed from Figure 4 that the schemas are correlated well and give meaningful clusters compared to existing approaches. Although the neural models achieve good performance on NMI, AMI, and ARI scores, JSONGlue has a significant drop in the performance of ARI because of the path-based similarity scores. It is noted that the intrinsic evaluation scores of InferSent, ELMo, and USE are extremely similar, but the proposed approach has shown a significant improvement over the other works. Also, the experiments show that exploiting contextual information of schemas improves the overall score.

It is noted from Tables 1, 2, and Figure 4 that the proposed work shows a substantial improvement in finding both the structurally and semantically equivalent JSON documents in comparison with existing approaches. The differences in the results appear to come from the way of handling the similarity approaches. The clustering performance reveals the ability of the proposed work to find contextual and semantic similarity before structural similarity. It is also observed from the results that the proposed approach also preserves the structure of attributes without losing essential information, such as repeated attributes. Therefore, it is evident that the proposed approach works exceptionally well on JSON datasets, irrespective of the nesting depth.

## 5 Conclusions

This paper proposed an approach to cluster JSON documents using the contextual, semantic, and structural similarity of JSON schemas. The proposed approach captures more semantics by merging semantic and contextual similarities scores. The results show that the proposed approach is superior to the state-of-the-art approaches in similarity calculation. In the future, we plan to extend this study for incremental clustering of JSON documents by updating the fusion similarity matrix for new documents and comparing its performance in a real-world scenario.

## References

[Accottillam et al., 2021] Accottillam, T., Remya, K., and Raju, G. (2021). Treexp—an instantiation of xpattern framework. In Data Science and Security, pages 61–69. Springer.

[Agarwal et al., 2016] Agarwal, M. K., Ramamritham, K., and Agarwal, P. (2016). Generic keyword search over XML data. In EDBT, pages 149–160.

[Bawakid, 2019] Bawakid, F. (2019). A schema exploration approach for document-oriented data using unsupervised techniques. Ph.D. thesis, University of Southampton.

[Blaselbauer and Josko, 2020] Blaselbauer, V. M. and Josko, J. M. B. (2020). Jsonglue: A hybrid matcher for JSON schema matching. In Proceedings of the Brazilian Symposium on Databases.

[Cer et al., 2018] Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Céspedes, M., Yuan, S., Tar, C., et al. (2018). Universal sentence encoder. arXiv preprint arXiv:1803.11175.

[Conneau et al., 2017] Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

[Costa and Ortale, 2013] Costa, G. and Ortale, R. (2013). A latent semantic approach to XML clustering by content and structure based on non-negative matrix factorization. In 2013 12th International Conference on Machine Learning and Applications, volume 1, pages 179–184. IEEE.

[Costa and Ortale, 2017] Costa, G. and Ortale, R. (2017). XML clustering by structure-constrained phrases: A fully-automatic approach using contextualized n-grams. International Journal on Artificial Intelligence Tools, 26(01):1760002.

[Costa and Ortale, 2019] Costa, G. and Ortale, R. (2019). Mining cluster patterns in XML corpora via latent topic models of content and structure. In Yang, Q., Zhou, Z.-H., Gong, Z., Zhang, M.-L., and Huang, S.-J., editors, Advances in Knowledge Discovery and Data Mining, pages 237–248, Cham. Springer International Publishing.

[D and Santhi Thilagam, 2022] D, U. P. and Santhi Thilagam, P. (2022). ClustVariants: An Approach for Schema Variants Extraction from JSON Document Collections. In 2022 IEEE IAS Global Conference on Emerging Technologies (GlobConET), pages 515–520.

[Dai et al., 2015] Dai, A. M., Olah, C., and Le, Q. V. (2015). Document embedding with paragraph vectors. arXiv preprint arXiv:1507.07998.

[Dongo et al., 2020] Dongo, I., Ticona-Herrera, R., Cadinale, Y., and Guzmán, R. (2020). Semantic similarity of XML documents based on structural and content analysis. In Proceedings of the 2020 4th International Symposium on Computer Science and Intelligent Control, ISCSIC 2020, New York, NY, USA. Association for Computing Machinery.

[Farouk, 2020] Farouk, M. (2020). Measuring text similarity based on structure and word embedding. Cognitive Systems Research, 63:1–10.

[Gallinucci et al., 2018] Gallinucci, E., Golfarelli, M., and Rizzi, S. (2018). Schema profiling of document-oriented databases. Information Systems, 75:13–25.

[Gallinucci et al., 2019] Gallinucci, E., Golfarelli, M., and Rizzi, S. (2019). Approximate OLAP of document-oriented databases: A variety-aware approach. Information Systems, 85:114 – 130.

[Gupta et al., 2019] Gupta, P., Pagliardini, M., and Jaggi, M. (2019). Better word embeddings by disentangling contextual n-gram information. CoRR, abs/1904.05033.

[Hammad et al., 2020] Hammad, M. M., Al-Smadi, M., Baker, Q. B., asa D, M. A., Al-Khdour, N., Younes, M. B., and Khwaileh, E. (2020). Question to question similarity analysis using morphological, syntactic, semantic, and lexical features. JUCS - Journal of Universal Computer Science, 26(6):671–697.

[Hill et al., 2016] Hill, F., Cho, K., and Korhonen, A. (2016). Learning distributed representations of sentences from unlabelled data. CoRR, abs/1602.03483.

[Jalal and Ali, 2021] Jalal, A. A. and Ali, B. H. (2021). Text documents clustering using data mining techniques. International Journal of Electrical & Computer Engineering (2088-8708), 11(1).

[Kiros et al., 2015] Kiros, R., Zhu, Y., Salakhutdinov, R., Zemel, R. S., Torralba, A., Urtasun, R., and Fidler, S. (2015). Skip-thought vectors. arXiv preprint arXiv:1506.06726.

[Laddha et al., 2018] Laddha, A., Joshi, S., Shaikh, S., and Mehta, S. (2018). Joint distributed representation of text and structure of semi-structured documents. In Proceedings of the 29th on Hypertext and Social Media, pages 25–32.

[Le and Mikolov, 2014] Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In International conference on machine learning, pages 1188–1196. PMLR.

[Liu et al., 2019] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., & Stoyanov, V. (2019). Roberta: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

[Logeswaran and Lee, 2018] Logeswaran, L. and Lee, H. (2018). An efficient framework for learning sentence representations. CoRR, abs/1803.02893.

[Ma et al., 2018] Ma, Z., Zhao, Z., and Yan, L. (2018). Heterogeneous fuzzy XML data integration based on structural and semantic similarities. Fuzzy Sets and Systems, 351:64–89.

[Miller, 1998] Miller, G. A. (1998). WordNet: An electronic lexical database. MIT Press.

[Mohamed L. Chouder and Stefano Rizzi and Rachid Chalal , 2017] Mohamed L. Chouder and Stefano Rizzi and Rachid Chalal (2017). JSON datasets for exploratory OLAP. doi:10.17632/ct8f9skv97.1. [Last accessed on 21-12-2020].

[Oliveira et al., 2020] Oliveira, A., Kohwalter, T., Kalinowski, M., Murta, L., and Braganholo, V. (2020). Xchange: A semantic diff approach for XML documents. Information Systems, 94:101610.

[Oliveira et al., 2018] Oliveira, A., Tessarolli, G., Ghiotto, G., Pinto, B., Campello, F., Marques, M., Oliveira, C., Rodrigues, I., Kalinowski, M., Souza, U., et al. (2018). An efficient similarity-based approach for comparing XML documents. Information Systems, 78:40–57.

[Pagliardini et al., 2017] Pagliardini, M., Gupta, P., and Jaggi, M. (2017). Unsupervised learning of sentence embeddings using compositional n-gram features. CoRR, abs/1703.02507.

[Peters et al., 2018] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. arXiv preprint arXiv:1802.05365.

[Piernik et al., 2016] Piernik, M., Brzezinski, D., and Morzy, T. (2016). Clustering XML documents by patterns. Knowledge and Information Systems, 46(1):185–212.

[Piernik et al., 2015] Piernik, M., Brzezinski, D., Morzy, T., and Lesniewska, A. (2015). XML clustering: a review of structural approaches. The Knowledge Engineering Review, 30(3):297–323.

[Sinoara et al., 2019] Sinoara, R. A., Camacho-Collados, J., Rossi, R. G., Navigli, R., and Rezende, S. O. (2019). Knowledge-enhanced document embeddings for text classification. Knowledge-Based Systems, 163:955–971.

[Tekli et al., 2018] Tekli, J., Chbeir, R., Traina, A. J., Traina, C., Yetongnon, K., Ibanez, C. R., Al Assad, M., and Kallas, C. (2018). Full-fledged semantic indexing and querying model designed for seamless integration in legacy RDBMS. Data and Knowledge Engineering, 117:133–173.

[Tekli et al., 2019] Tekli, J., Chbeir, R., Traina, A. J., and Traina Jr, C. (2019). Semindex+: A semantic indexing scheme for structured, unstructured, and partly structured data. Knowledge-Based Systems, 164:378–403.

[Tekli et al., 2021] Tekli, J., Tekli, G., and Chbeir, R. (2021). Almost linear semantic XML keyword search. In Proceedings of the 13th International Conference on Management of Digital EcoSystems, pages 129–138.

[Uma Priya et al., 2020] Uma Priya D, Santhi Thilagam P. Dynamic data retrieval using incremental clustering and indexing. International Journal of Information Retrieval Research (IJIRR). 2020;10(3):74-91.

[Vinh et al., 2010] Vinh, N. X., Epps, J., and Bailey, J. (2010). Information theoretic measures for clusterings comparison: Variants, properties, normalization, and correction for chance. J. Mach. Learn. Res., 11:2837–2854.

[Von Luxburg, 2007] Von Luxburg, U. (2007). A tutorial on spectral clustering. Statistics and Computing, 17(4):395–416.

[Wang et al., 2015] Wang, L., Zhang, S., Shi, J., Jiao, L., Hassanzadeh, O., Zou, J., and Wangz, C. (2015). Schema management for document stores. Proc. VLDB Endow., 8(9):922–933.

[Wang and Koopman, 2017] Wang, S. and Koopman, R. (2017). Clustering articles based on semantic similarity. Scientometrics, 111(2):1017–1031.

[Wu et al., 2020] Wu, H., Liu, Y., and Wu, Q. (2020). Stylistic syntactic structure extraction and semantic clustering for different registers. In 2020 International Conference on Asian Language Processing (IALP), pages 66–74. IEEE.

[Wu and Palmer, 1994] Wu, Z. and Palmer, M. (1994). Verb semantics and lexical selection. arXiv preprint cmp-lg/9406033.

[Yeung et al., 2001] Yeung KY, Ruzzo WL. Details of the adjusted rand index and clustering algorithms, supplement to the paper an empirical study on principal component analysis for clustering gene expression data. Bioinformatics. 2001 May 3;17(9):763-74.

[Yousfi et al., 2020] Yousfi, A., El Yazidi, M. H., and Zellou, A. (2020). xmatcher: Matching extensible markup language schemas using semantic-based techniques. International Journal of Advanced Computer Science and Applications, 11(8):655–665.