

Article

Not peer-reviewed version

Deep Ontology Alignment using Natural Language Processing Approach for Automatic M2M Translation in IIoT

[Saleha Javed](#) ^{*}, Muhammad Usman Joyia , Fredrik Sandin , [Marcus Liwicki](#) , [Hamam Mokayed](#)

Posted Date: 21 September 2023

doi: [10.20944/preprints202309.1413.v1](https://doi.org/10.20944/preprints202309.1413.v1)

Keywords: Ontology Alignment; M2M Translation; Self-Attention, Deep Learning; Industry 4.0; Industry 5.0 IIoT; Knowledge Graph; Industrial Internet of Things; Smart City



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Deep Ontology Alignment Using Natural Language Processing Approach for Automatic M2M Translation in IIoT

Saleha Javed ^{1,*} , Muhammad Usman ² , Fredrik Sandin ¹ , Marcus Liwicki ¹ , and Hamam Mokayed ¹ 

¹ Lulea University of Technical, 97187 Lulea, Sweden

² Department of Computer Science, National University of Computer and Emerging Sciences, Islamabad, Chiniot-Faisalabad Campus, Chiniot 35400, Pakistan;

* Correspondence: saleha.haseeb@ltu.se

Abstract: The technical capabilities of modern Industry 4.0 and Industry 5.0 are rather vast and growing exponentially daily. The present-day Industrial Internet of Things (IIoT) combines manifold underlying technologies that require real-time interconnections and communications among heterogeneous devices. Smart cities are established with sophisticated designs and control for seamless Machine-to-Machine (M2M) communication to optimize resources, costs, performances, and energy distribution. All the sensory devices within a building interact to maintain a sustainable climate for residents and intuitively optimize the energy distribution to optimize energy production. However, it encompasses quite a few challenges for devices that lack compatible and interoperable designs. Conventional solutions are restricted to limited domains or rely on engineers to design and deploy translators for each pair of ontologies. This is a costly process in terms of engineering efforts and computational resources. The issue persists that a new device with a different ontology must be integrated into an existing IoT network. We propose a self-learning model that can determine the taxonomy of devices given their ontology meta-data and structural information. The model finds matches between two distinct ontologies using the Natural Language Processing (NLP) approach for learning linguistic contexts. Then, by visualizing the ontology network as a knowledge graph, it is possible to learn the structure of the meta-data and understand the device's message formulation. Finally, it can align entities of both ontology graphs similar in context and structure. Furthermore, the model performs dynamic M2M translation without requiring extra engineering or hardware efforts.

Keywords: Ontology Alignment; M2M Translation; Self-Attention, Deep Learning; Industry 4.0; Industry 5.0 IIoT; Knowledge Graph; Industrial Internet of Things; Smart City

1. Introduction

The speed of technology development is changing with automation and digitization efforts, bringing forward several challenges [1]. The backbone of Industry 4.0 and 5.0 was industrial automation systems that enabled sustainable development [1] and innovative functionalities access to the cyber world [2] known as Cyber-Physical Systems (CPS). CPS is a conjunction between physical systems and digital microsystems that feature a tight integration for modeling, computation, and communication. Both Cyber-physical systems and the IoT have been merging in the industrial digitization process, further known as the Industrial Internet of Things (IIoT). The focus of such merges has been reshaping society [2] by bridging the physical divides via digital connectivity through IIoT and digitization applications. The applications such as automation of manufacturing processes [3,4], agriculture for precision fertilization programs [5], smart farming, condition monitoring of wind turbines [6] and farms, smart factories [7], and smart building and cities [8], and many others. By digitizing physical processes, these applications have lowered overheads associated with human dependency, cost, time, and computations. While these solutions aim to achieve connectivity across their respective Service-Oriented Architectures (SOA) when it comes to developing a dynamically

scalable and enhanced Software-as-a-Service (SaaS) architecture that can incorporate machine learning models as a service (MLaaS) [9], such systems are still in their infancy. Additionally, this problem becomes more challenging and crucial in the environmental settings of Industry 5.0. This application domain is the hub of devices with different responsibilities working together for the same business objective. Despite these devices having homogeneous or heterogeneous underlying structures, these devices need to comprehend, translate, and interact with each other to converge toward the business goal. Thus, IIoT automation cannot be confined to the digitization of connections, and this development is subject to interoperability challenges. In particular, Machine Learning (ML) approaches are considered to automate costly engineering processes. For example, challenges related to the automatic translation of messages transmitted between heterogeneous devices are investigated using supervised and unsupervised machine learning approaches [10].

We conceive the IIoT device ontology as the device's language, corresponding to the language encoder component. The schema of the ontology graph contains all the information about classes and sub-classes hierarchy and their connection, which we convert into the structural encoder. Then, the names of classes and relations are considered labels mapped as side information in the ontology graph and as sentence tokens in the NLP paradigm. Finally, the relations indicate which classes are interconnected, and these constitute the structural question set. To the best of our knowledge, no other work in literature has proposed this mapping, and so there is a knowledge vacuum about the efficient use of such synergies. Existing techniques of entity alignment are based on different approaches for integrating structural information, which overlook that even if a node pair has similar entity labels, they may not belong to the same ontological context, and vice versa. To address these challenges, a model based on modifying the BERT-INTeraction model on graph triples is developed. The developed model is an iterative model for the alignment of heterogeneous IIoT ontologies, enabling alignments within nodes and relations. When compared to the state-of-the-art BERT INT, on the DBPK15 language dataset, the developed model exceeds the baseline model by an error rate of 2.1%. This work can be considered as a step towards enabling translation between heterogeneous IoT sensor devices; therefore, the proposed model can be extended to a translation module in which, based on the ontology graphs of any device, the model can interpret the messages transmitted from that device.

We focus on designing an ontology alignment model as a first step toward developing automatic dynamic translation between IIoT heterogeneous devices. The following summarizes our main contributions:

- Thoroughly investigate how to enable automatic alignment across heterogeneous IIoT sensor devices using NLP NLP-based learning model in conjunction with entity alignment for the ontology graph.
- Explore the use of an ontology graph as the main metric in a representation learning problem for interpreting the metadata of sensory devices.
- The first significant novelty herein is highlighting three knowledge gaps: 1) lack of research attention on modeling ontology alignment approaches for IIoT heterogeneous devices, 2) scarcity of literature on fusing NLP methodologies with IIoT domain, and limitations of datasets for IIoT ontology alignment.
- The second prime novelty of this work is synthesizing a model as a solution for the IIoT ontology alignment task. The model significantly exceeds the state-of-the-art results on the DBP15K languages dataset by a wide margin. This work is the first of many to conceptualize a mapping between NLP and IIoT domains by utilizing knowledge graph modeling for the device's ontology.

The paper is outlined into eight sections, first, a brief background is given in Section 2 of the various domains used in constituting the proposed solution. Section 3 presents the important state-of-the-art of each domain. Then, a detailed discussion on highlighted knowledge gaps is given in Sections 3.4. Section 4 elaborates on the problem formulation followed up by Section 5 with the complete architecture of the proposed solution followed by a use case explanation for the proposed system

discussed in section 6. Then, Section 7 states the used experimental setup and a proof of concept with results is presented in Section 8. Lastly, all the reflections and concluding remarks are discussed in Section 9.

2. Background

Numerous models with varying strengths and weaknesses have been established for cross-language translation, but none have been designed for the IIoT automatic ontology paradigm. This section outlines the different dimensions that go into synthesizing the proposed solution. The first dimension is about the IIoT ontology's constitution and role from an industrial perspective. The next dimension addresses the importance of interoperability in the context of ontologies and the popularity of ML for modernizing Industry 4.0 applications and leading to Industry 5.0 smart society applications.

2.1. Interoperability in Context of IIoT Ontologies

With the development of embedded CPSs and vast computational resources, IIoT has grown significantly, resulting in a massive increase in the number of IoT devices. According to recent figures, the number of linked IoT devices globally has reached 15.14 billion in 2023. This forecast is expected to quadruple to around 50 billion IoT devices by 2030 [11]. IIoT is a hub for heterogeneous and homogeneous devices that needs seamless integration and connectivity. The interoperability issue is the challenge of enabling communication to occur despite varying assumptions about the data model, message format, and device ontology [12]. Figure 1 presents an example scenario of ontology interoperability. In the past decade, researchers have shown keen interest in developing ML-based automatic translation models for solving interoperability problems, but the lack of datasets and complexity constraints of real-world applications have been hindering this synergy so far.

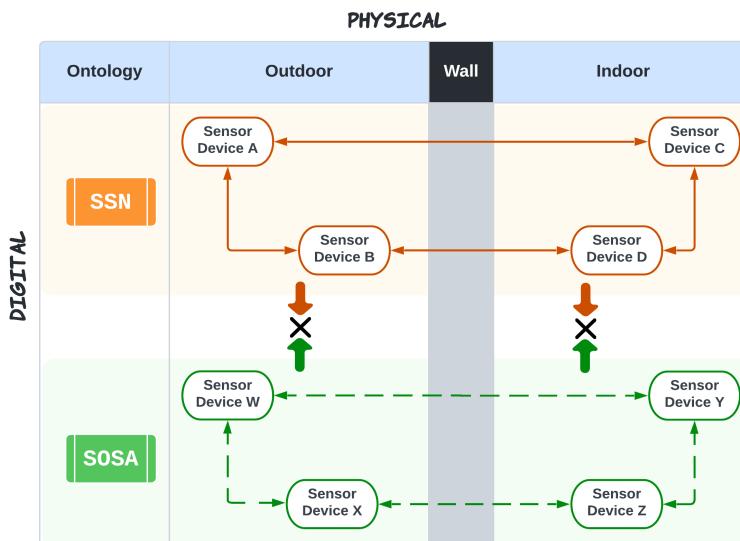


Figure 1. Explanation of heterogeneity in device ontology. The figure illustrates an example scenario of a smart building with multiple interconnected sensors installed outside and inside. Few devices follow the Semantic Sensor Network (SSN) ontology; the rest follow the Sensor-Observation-Sampling-Actuator (SOSA) ontology. All the devices that follow SSN ontology can intercommunicate, and similarly, devices that follow SOSA ontology can successfully intercommunicate. However, a device following SSN ontology can not communicate with the device following SOSA [13].

2.2. Representation Learning for Sensor Devices

The performance of ML algorithms is highly dependent on the type of data representation used. As a result, a major percentage of the effort is spent on feature engineering to execute ML algorithms and build data transformations that result in a representation of the data suited for effective learning. [14]. Data of sensor devices is conceptualized in several technical layers of SOAs. It includes the device's ontology and protocol, data format, message payload schema, message transmission protocol, and more. However, this work emphasizes the importance of device ontology for identifying and disentangling the messages received from a heterogeneous device. Given any device's messages and its ontology, the representation learning model can map vector representation of low-dimensional space for each entity in the ontology. The vectors of every unique entity are also unique, called embedding vectors. There are three major methods in which the model can perform representation learning: 1) supervised in which input labels and mapping of input X to output Y are given; 2) semi-supervised in which a mix of labeled data and unlabelled data is used; and 3) unsupervised in which no prior information of labels or mapping onto output is given. The present IoT sensor ontology domain literature has examples of supervised and semi-supervised approaches as discussed in Section 3 but lacks unsupervised learning-based methods.

3. Related Work

The work presented herein is primarily in the context of the industrial Internet of things paradigm. We address the translation problem amongst heterogeneous sensory devices with respect to the ontology followed for installing the network in a smart building. Here, all the devices are interconnected for regulating and optimizing energy consumption such as temperature control (heating or cooling), humidity, or climate. Each subsection presents the important state-of-the-art of various domains that contribute to hypothesizing the research question and its solution.

3.1. Sensor Ontologies

Sensors are a major source of data available on the Web today. While sensor data may be published as mere values, searching, reusing, integrating, and interpreting these data requires more than just the observation results. The captured information with its context is equally important for properly interpreting these values as information about the studied feature of interest, such as for a heater, the observed property, the specific locations and times at which the temperature was measured, and a variety of other information. This work takes into account only the ontology that is standardized, integrated by, and aligned with W3C semantic web technologies [15] and Linked Data [16], which are key drivers for creating and maintaining a global and densely interconnected graph of data. Intelligent sensors should be seamlessly, securely, and trustworthy interconnected to enable automated high-level smart applications. Smart interconnection of sensors, actuators, and devices enables the development of solutions required for smart city- and CPS industrial solutions [17].

Ontologies can enrich sensory data and ensure interoperability by providing an abstraction layer [18]. The ontology defines the semantic model and contextual information of the devices [19]. Figure 2 shows the essential components of an ontology design. W3C has developed several benchmark ontologies based on IoT standards, such as Smart Onto Sensor, SSN, SAN, IoT-Lite, SOSA, and others, adopted by industrial manufacturers globally. The authors present [18] a timeline of the evolution of all base-level ontologies developed from 2002 till 2018. The authors divide the timeline into *before* and *after* SSN ontology as it was the first ontology with complete design patterns for sensory devices network. Ontologies are continually evolving, compiling ever more space for reasoning and simplification.

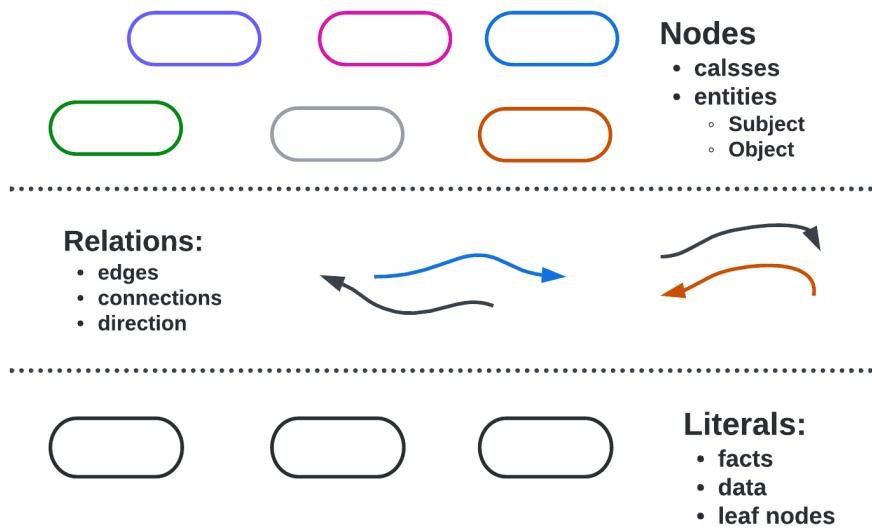


Figure 2. Basic Components of an Ontology. There are three types of nodes here: 1) subject node, 2) object node, and 3) literal node. Both subject and object nodes belong to a class of the knowledge domain for which the ontology is being developed. The edges between nodes represent the relations, and the third literal node has a data fact in them.

SOSA provides a lightweight core for SSN as shown in Figure 3, and aims at broadening the target audience and application areas that can make use of Semantic Web ontologies. At the same time, SOSA [20] acts as a minimal interoperability fall-back level, i.e., it defines those common classes and properties for which data can be safely exchanged across all uses of SSN [21], its modules, and SOSA.

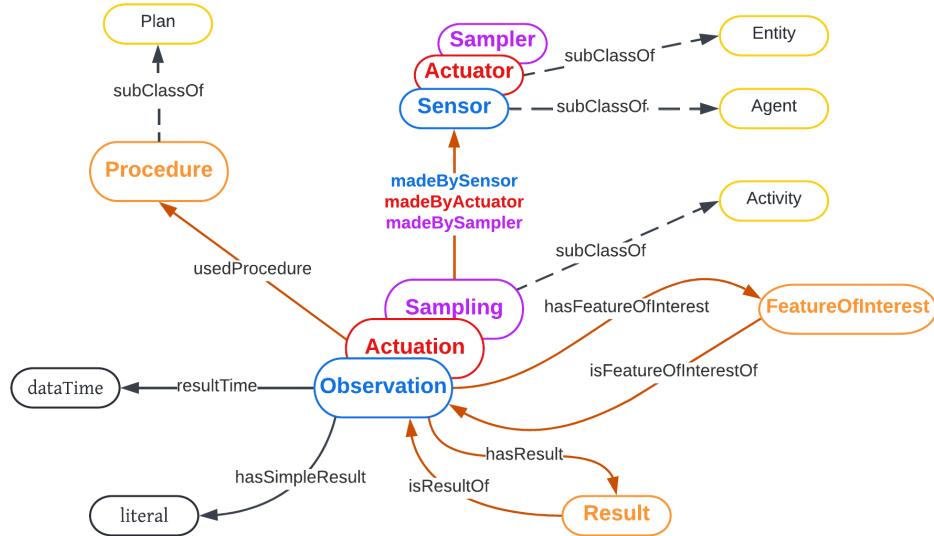


Figure 3. SSN and SOSA Ontology Core structure.

3.2. M2M Translation Problem in IIoT Domain

Devices often use different communication protocols, standards, and data representation languages, which create interoperability and M2M translation challenges. Existing literature contains different perspectives to address the M2M translation problem. Application protocol-level solutions focus on pre-defined functions or annotations as proxies and XML schemes to enable translation

between sender and receiver devices [22,23]. However, such solutions fail regarding automated CPSs that cannot rely on hand-crafted predefined schemes for every possible pair of devices. Moreover, protocol-level proxies exclude the possibility of utilizing data in the messages to draw intuitive interpretations about the device's protocol. Data-driven methods [24] exploit the data augmentation approach to analyze patterns and features in device data messages and infer important knowledge that can generate interpretations between heterogeneous devices. However, to the best of our knowledge, a successful automatic translation model based specifically on industrial IoT ontologies has not been developed. The major challenge in developing such learning-dependent solutions is the unavailability of large datasets, which is a massive hindrance.

3.3. Knowledge Graph Alignment

Knowledge graph alignment aims to link equivalent entities across different knowledge graphs. ML models in conjunction with data-driven methods for automatic semantic translations, have recently been trending among researchers [25]. Deep learning (DL) models such as deep alignment for ontology [26] design solutions among parallel ontologies by aligning entities of different ontologies have been developed independently but for the same domain. [26] introduced word vector-driven descriptions for defining the entities (nodes) and matching tasks on DBpedia dataset of ontologies and Schema.org. Recently, a large number of Knowledge Graphs (KGs) have been established for supporting AI applications, such as Freebase [27] and YAGO [28]. Entity alignment seeks to discover identical entities in different KGs, such as the English entity Thailand and its French counterpart Thailande. To tackle this important problem, literature has attempted with the embedding-based entity alignment methods [29–31]. These methods jointly embed different KGs and put similar entities at close positions in a vector space, as shown in Figure 4, where the nearest neighbor search can retrieve entity alignment.

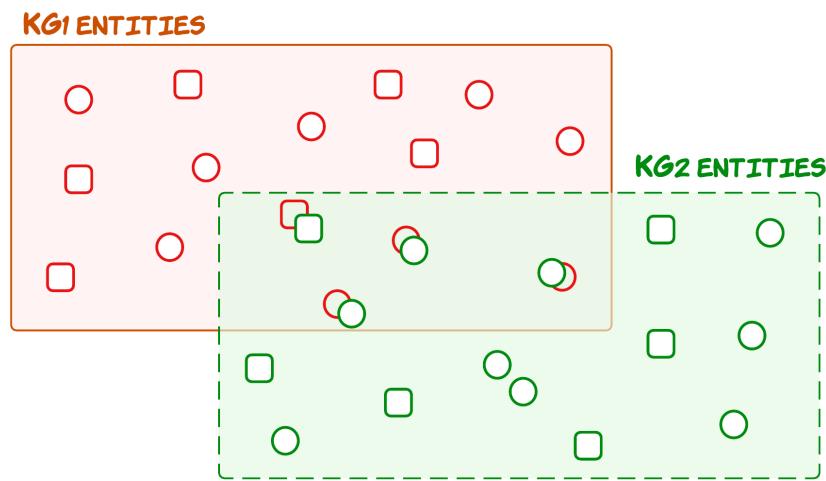


Figure 4. Illustration of entity alignment between two heterogeneous KGs. Each KG has its embedding vector space for its entities, i.e., circles represent nodes, and squares represent relations. The entities from both graphs that have similar embedding in the vector space overlap in the figure.

Due to its effectiveness, embedding-based entity alignment has drawn extensive attention recently. KGs have evolved to be the building blocks of many intelligent systems. They provide fundamental tools for NLP tasks [32] of language representation through BERT, knowledge reasoning [33], recommend systems using Knowledge Graph Convolutional Networks (KGCN) [34], Cross-lingual entity alignment (CEA) based on generative adversarial network (GAN) [29] with semi-supervised learning. Despite the importance, KGs are usually costly to construct and naturally suffer from incompleteness [35]. Table 1 summarizes a brief survey of recent renowned graph alignment methods

on the account for whether they are scalable for IIoT domain or not. The analysis is focused on the utilization of both language and structure information. It is evident that most of the models heavily rely on pre-aligned entities to be used during the training stage.

Table 1. Summarizing herein recent and renowned state-of-the-art methods for Graph Alignment task.

Ref.	Learning Approach	Entity Matching	Domain	Datasets Used	Structure Information Used?	Scalable for IIoT Domain?
[36]	supervised	MTransE based alignment	Languages	WK31-15k	✓	yes, but only if benchmark datasets is present
[37]	supervised	GCN based entity embeddings	Languages	DBP15K	✓	yes, but only if all pre-aligned entities are included in training data
[38]	supervised	Topic entity graph using GCN	Languages	DBP15K	✗	no, as each entity in ontology graph represents a topic and topic grouping will fail here
[39]	unsupervised	TransE based predicate alignment for attribute character embeddings	Locations	DBP, GEO, YAGO	✓	no, as structure information is used only to learn the relations labels and not the interconnections
[40]	supervised	BERT based Interaction Model	Languages	DBP15K	✗	yes, but only with unsupervised learning approach and inclusion of structure information
[41]	semi-supervised	BERT for Triadic KG	Languages	DBP15K	✗	no, as ontology graphs can not be realized as triadic KG with all independent entities

3.4. Challenges to Adaptation and Integration

For the foreseeable future, ML models will play a prime role in automating the current industrial applications into intelligent solutions. However, as the previous sections highlight, research in translation among IoT devices and automatic language translation is so far working as isolated areas, whereas their synergy can bring bigger benefits to both. The following sub-section presents the important gaps this work is based on and crisp indications for plausible merges to bridge those gaps.

3.4.1. State-of-The-Art Limitations

We conducted a query search in three renowned search engines, i.e., Google Scholar, SCOPUS, and Web of Science, to investigate all the existing research publications for the given problem. The main metric of this analysis is the number of publications per year.

The search queries are designed sequentially, in which the first search query is on publications for M2M translation but only within the Industry 4.0 paradigm. The second query is narrowed down to the same problem but specifically addressed by ML approaches. And, lastly, the third query investigates the number of publications that have focused on ML models for solving ontology alignment problems. Table 2 presents all the statistics of search results, and the numbers indicate the lack of attention towards ML approaches for solving M2M translation problems, specifically using alignment tasks.

Table 2. Details of the search queries in different search engines and search results in the number of publications in every year. Search in Google Scholar is on "Entire Article", and SCOPUS is on "Title, Abstract, Keywords" and Web of Science is only on "Abstract".

Year	Query 1: M2M Translation & Industry 4.0			Query 2: M2M Translation & Industry 4.0 & ML			Query 3: M2M Translation & Industry 4.0 & ML & Ontology Alignment		
	Google Scholar	SCOPUS	Web of Science	Google Scholar	SCOPUS	Web of Science	Google Scholar	SCOPUS	Web of Science
2010	10200	95	76	9180	0	0	232	0	0
2011	11600	89	85	10900	0	0	229	0	0
2012	12900	101	94	12800	0	0	231	0	0
2013	13400	116	93	14500	0	0	233	0	0
2014	15000	220	140	16100	0	2	202	0	1
2015	16000	344	306	17100	2	3	196	0	1
2016	18500	751	500	17500	21	8	211	0	0
2017	16600	1432	957	16600	49	24	201	1	0
2018	22900	2495	1501	19600	146	77	206	1	1
2019	16800	4886	2242	16700	329	128	210	1	1
2020	29700	5577	2514	22900	496	148	236	1	1
2021	37300	6706	3118	26700	791	225	143	1	0
2022	45120	8954	3118	22950	977	225	298	2	0

3.4.2. Lack of NLP fusion into IIoT Domain

Dynamic translation between machines has stressed the need to establish automated systems that enable effective real-time communication across heterogeneous devices. The literature is unquestionably packed with NLP solutions for various industrial applications, including language translation (chatbots); most focus on a pre or post-process analysis of processes and datasets. On the other hand, IIoT network activities are ongoing and greatly diverse, and there is a crucial need to deploy automatic translators for dynamic, seamless communication between heterogeneous devices. Using NLP models for that purpose is a considerable gap in the study. As seen in Table 2, researchers place premium attention on language datasets, even regarding graph alignment approaches. This work is the first of many efforts to conceptualize the mapping and validate the proposed solution as a proof of concept. To understand how mapping is implemented in this study, let us dissect the NLP domain into its main components: a language encoder, a structural encoder, language sentences and tokens, and a structural question set.

3.4.3. Limitations of Dataset for IIoT Ontology Alignment

Considerable efforts have been made, and will continue for the foreseeable future, to develop a variety of datasets for computer-based linguistic technology applications [42]. The research community recognizes that only data can pave the way for linguistic technology. Hence, the number of publicly accessible NLP datasets has grown significantly as researchers experiment on new tasks, larger models, and novel benchmarks [43]. Datasets are essential in empirical NLP studies since they are utilized to evaluate proposed models and their bench-marking. Supervised datasets with predefined annotations are required to train and fine-tune the models, and large unsupervised datasets are required for pre-training and language modeling. DBP15K [44], YAGO [45] and DWY100K [46] are the widely used massive benchmark datasets of knowledge bases for alignment tasks, with high alignment accuracy of existing embedding-based methods. Each consists of approximately millions of KG triplets with thousands of entities and relations.

Whereas there is plenty of research attention and datasets for cross-linguistic alignment tasks, both are increasingly scarce for industrial IoT ontology alignment. The IoT ontology graphs are concise since they are curated for specific industrial use cases and devices. As seen in Table 3, fewer nodes and graph triples than the language datasets' knowledge bases.

Table 3. Statistics of the empirical NLP datasets used for entity alignment for two domains: a) contemporary language-based b) IIoT domain utilizing both structure and language-based alignments

Dataset		Entities	Relations	Triples
<i>Domain: Language-based</i>				
DBP15K _{ZH-EN}	Chinese	66,469	2,830	153,929
	English	98,125	2,317	237,674
DBP15K _{JA-EN}	Japanese	65,744	2,043	164,373
	English	95,680	2,096	233,319
DBP15K _{FR-EN}	English	66,858	1,379	192,191
	French	105,889	2,209	278,590
<i>Domain: IIoT Language + Structure-based</i>				
Smart Appliance REference (SAREF)		37	20	1097
Semantic Actuator Network (SAN)		17	17	271
Semantic Sensor Network (SSN)		105	40	767
Sensor, Observation, Sample, & Actuator (SOSA)		70	23	487

4. Problem Formulation

This section contains two key definitions designed for the address problem domain. Then, we present the problem definition targeted in this work.

4.1. Definition 1: Knowledge Graph and Structure Encoding

We generate KGs of two forerunner ontologies by W3C regulations: SSN and SOSA as KG_1 and KG_2 . A graph is denoted as $KG = (H, T, R)$ where R is a set of all relations entities, H is a set of all head entities, and T is a set of all tail entities. Each edge represents a relation $r \in R$, a subject node represents $h \in H$, and an object node represents $t \in T$. In the structural encoder of the proposed model, there are four representation vectors: D_H^S , D_R^S , D_T^S , and D^S . Vector D_H^S represents the path length from a head entity, D_R^S represents the path length from the relation, D_T^S represents the path length from a tail entity, and D^S which encodes the structural information of the underlying KG. Entity pairs between KG_1 and KG_2 are denoted as:

- for pair of head entities $g(h, h')$ where $h \in H \in KG_1$ and $h' \in H' \in KG_2$
- for pair of relation entities $g(r, r')$ where $r \in R \in KG_1$ and $r' \in R' \in KG_2$
- for pair of tail entities $g(t, t')$ where $t \in T \in KG_1$ and $t' \in T' \in KG_2$

4.2. Definition 2: Mapping to BERT Language Model

The metadata, labels of nodes, and relations are conceived as the language of the IoT ontology. The language encoder of our proposed model is similar to the original BERT encoder [47]. Sets of H , R , and T along with D^S vectors are encoded into B^L encoder on which we apply concatenation to generate a final language representation vector as C^L . KG_1 will have a matching node in KG_2 if a node e_i has a similar embedding vector e'_i in the common latent space of both KGs.

4.3. Problem Definition: Ontology Graph Alignment

The problem herein is manifold. Given two ontology graphs KG_1 and KG_2 provided they both are designed for IIoT sensor devices, the prime task is to learn alignment across the heterogeneous ontology graphs. For which, we first use language BERT encoder (B^L) on the ontology dataset and further process it by a two-layer Multi-layer Perceptron (MLP) network that learns the final language representation vector as C^L . Next, we use the structural encoder to transform the language vectors into a binary vector D^S to capture the triplets and in-graph information with respect to neighboring nodes. Then, the interaction model is used to learn the alignment across the graphs with two baseline assumptions:

1. An entity from a KG_1 can only match with only one entity in KG_2 . The term $C_{ij}^{uniquemax}$ ensures this property in two different KGs.
2. If an entity e_i from KG_1 aligned with entity e_j of KG_2 then their neighbour will also have similar properties. The term S_i^{topsum} ensures this property in the neighbor of e_i and e_j .

Lastly, a $\text{Loss}_{interaction}$ function is defined to learn the maximal similarity based on the side and structural information of different entities from both KGs.

5. Proposed Conceptual Design

5.1. Overview of the proposed system

There are two forms of information available in a KG. The first is language information, and the second is structural information. The BERT-based encoders have already proved their effectiveness for language models [47]. Recently, the BERT-INT, a BERT encoder, has also been used for the entity alignment task in KGs [40]. But BERT-INT [40] only used language information with a BERT encoder to generate an encoded vector, which is further encoded by a multi-layered-perceptron (MLP) network to yield the final representative vector for a given query.

Indeed, the structural information is used in its interaction model at the last stage, but considerably, structural information is not covered effectively by BERT-INT. In this work, we present a model-based

solution for Ontology Alignment using a modified BERT-INT model on graph Triplets that encodes the available information in KGs with or without language pieces of information. Figure 5 illustrates the overview of the model starting from two heterogeneous sensor devices that have different ontologies.

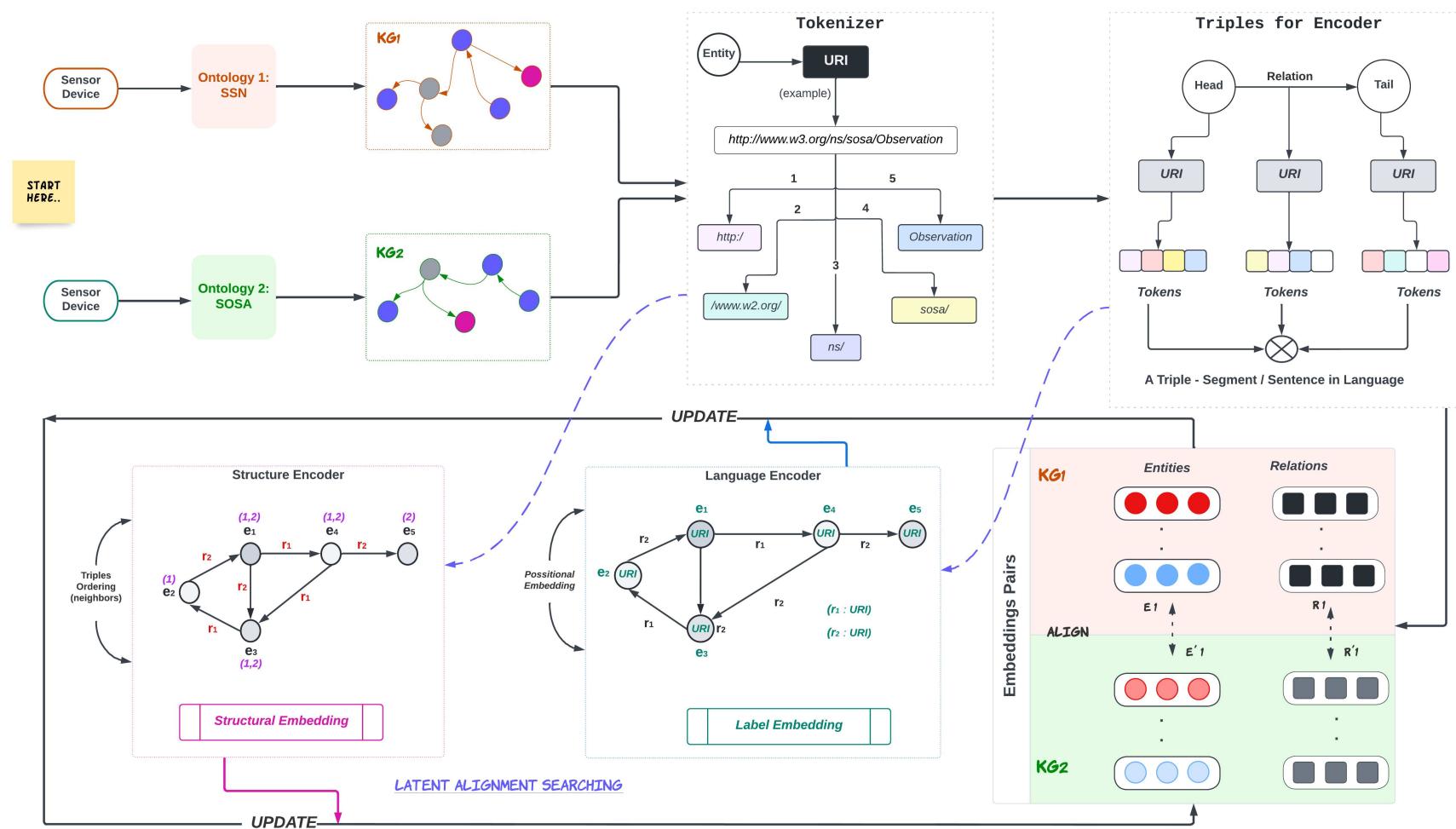


Figure 5. Complete overview of the proposed model with abstract components.

5.2. Improvements on BERT_INT Model

The following sections present in detail every component of the proposed model. However, here is a summary of proposed improvements to the state-of-the-art model:

- Modified input arrangement is used in this work to utilize the full potential of a pre-trained BERT model.
- The improved input arrangement can be used for experiments of aggregation models that are designed using both language and structural encoders.
- For integrating the structural encoder and incorporating side information with an improved BERT-INT model, the structural question-set reasoning block is designed and implemented with an in-graph approach.
- Interaction model is changed by proposing an iterative method of calculating similarities between entities over each iteration.
- Interaction model is designed for an unsupervised learning approach as in the case study used for the work, where no alignment pairs are available for KG1 and KG2.

6. Ontology Dataset Construction for System Use Case

We select two ontologies SOSA and SSN as discussed in subsection:3.1 as these are the forerunner ontology curated by W3C on the account of IoT sensor devices. For generating ontology instances strictly on SOSA and SSN ontology graphs we follow the W3C standardized examples of *Appartement 134* [48] and utilize the RDF (Resource Description Framework) files containing graphs with SOSA and SSN core terms. The example is designed for temperature sensor devices and an actuator in which the devices log their temperature values for corresponding time stamps. Although this gives us a complete graph of both ontology for sensor devices for the training of machine learning model we require a much bigger number of ontology instances.

Therefore, we refer to Kaggle's dataset of Smart Building Data [49] synthesized by Hong et.al [50]. This dataset was collected from 255 sensor time series, instrumented in 51 rooms on four floors of the Sutardja Dai Hall(SDH) at UC Berkeley. The dataset can be utilized for experiments relating to IoT, sensor fusion networks, or time-series tasks. It is also suitable for both supervised and unsupervised learning tasks. The building infrastructure is that each room includes five types of measurement sensor data as shown in Figure 6. In the following sections, we discuss the complete workflow the proposed system for the language encoding and ontologies structure construction.

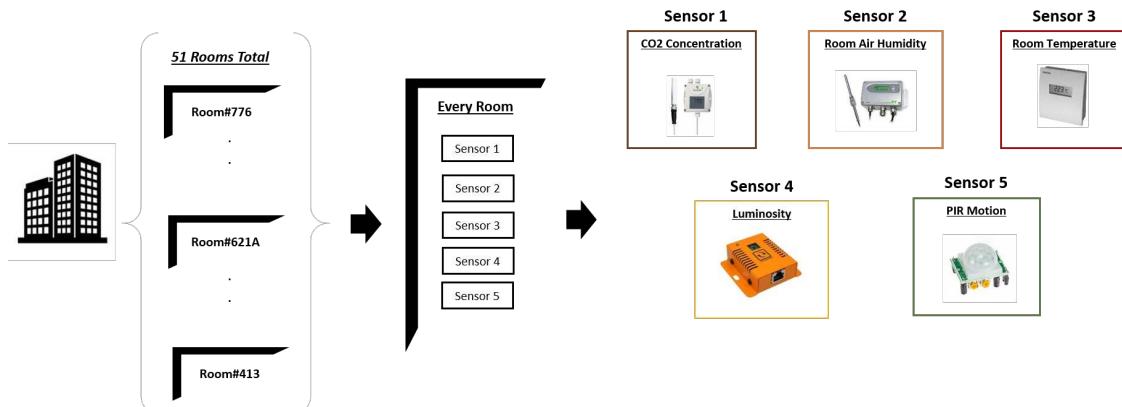


Figure 6. Smart Building System Dataset collected over a period of one week from Friday, August 23, 2013 to Saturday, August 31, 2013. The PIR motion sensor is sampled once every 10 seconds and the remaining sensors are sampled once every 5 seconds. Each file contains the timestamps (in Unix Epoch Time) and actual readings from the sensor.

6.1. Language Encoder

The language encoder of the proposed work is similar to the BERT-INT with modification as discussed in section 5.2 and generates the language representative vector for each entity and relation in the graph represented in Figure 7. Then, the language vectors of the head, relation, and tail of the triplet are concatenated to form the input vectors for the structural BERT encoder shown in Figure 7b. The corresponding embeddings generated by the structural BERT are further diverged into three separate vectors by another MLP network to yield the final representative vector for the respective triplet's head, relation, and tail.

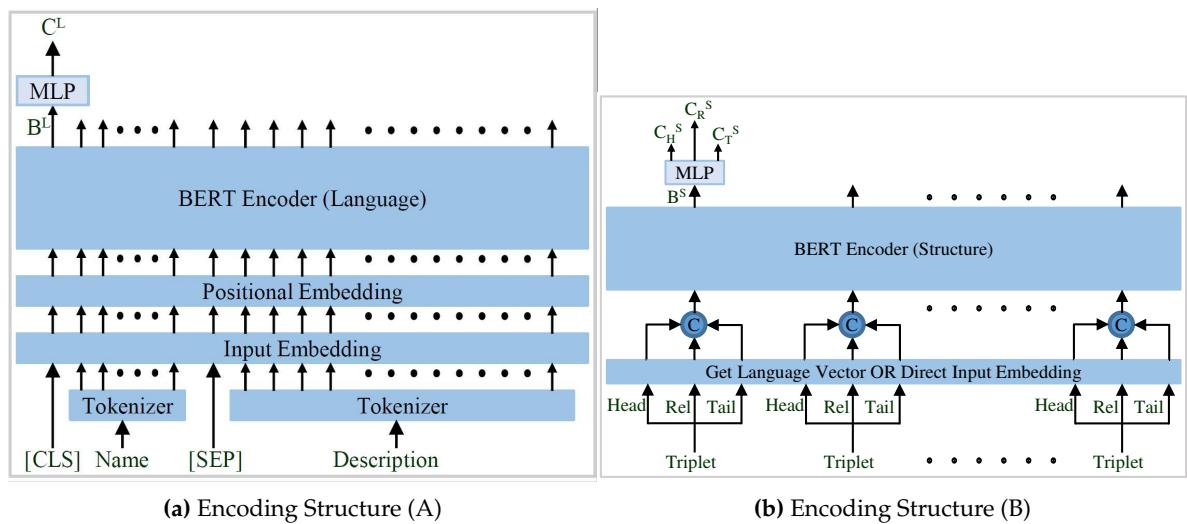


Figure 7. Different BERT Encoders used in the proposed model.

The original BERT encoder [47] uses sentence-1, and sentence-2 input arrangement as shown in Figure 8a. The same input arrangement is utilized by most of the methods that utilize the pre-trained BERT model [47]. But the BERT-INT [40] does not use this input organization and uses a very different arrangement shown in Figure 8b. Therefore the utilization of the full potential of a pre-trained BERT model is questionable. In contrast, the input arrangement of the proposed language encoder as represented in Figure 8c is very similar to the original BERT encoder. Here only the input arrangement is updated and everything else remains the same as BERT-INT. The representation generated by the language BERT encoder (B^L) is further processed by a two-layered MLP network which yields the final language representation vector as C^L .

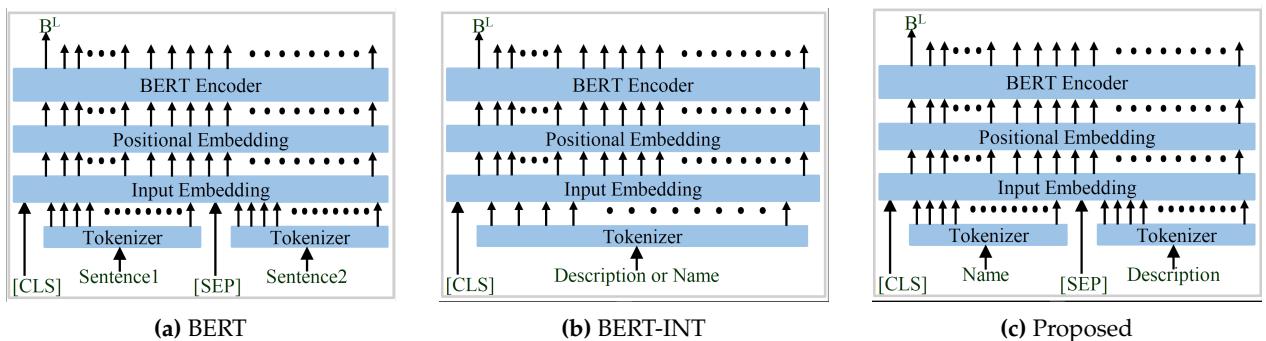


Figure 8. Different Input Arrangements for BERT Encoder.

6.2. Structural Encoder

Structural encoder yields output for a given KG as input such that the generated output can answer all questions related to the structure of the KG as shown in Figure 9. However, there are two issues with this structural encoder:

1. How to represent the complete KG as input?
2. What should be the questions set that capture all structural information of KG?

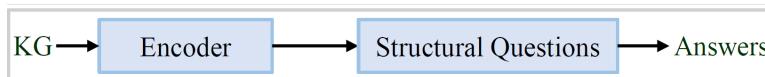


Figure 9. structural Encoder Block

Processing the complete KG as input for very large KGs is not computationally feasible, so initial work tries to generate the embedding vectors for the different components of KG (such as head [subject], relations, and tail [object]). Generating an embedding vector for a component of KG requires contextual information, but acquiring all the contextual information of a node or a relation is complex. Therefore, most existing works treat all neighbors within a specific path length as the context of the targeted node. Besides this, these embeddings should provide answers to structural questions. The most famous approaches are 1) continuous bag of words (CBOW) and 2) Skip-gram for encoding structural information.

6.2.1. Graph Representation for Structural Encoder

In this work, we represent a graph by its set of triplets. These triplets are passed to the structural encoder to incorporate the structural information. These triplets do not have any specific order, so we are not integrated with the positional encoder. Besides this, the set of triplets passed as input at a time is considered as the in-graph. The components of the original graph that are not part of the in-graph are considered as care for structural encoder processing. Therefore only the elements of the in-graph (nodes and relations) will participate, differentiating the entity from having different neighbors and weakening the issue of aggregating neighbors.

We require a cost function to train the structural encoder such that the generated representation vectors should incorporate the structural information of the underlying knowledge graph. We can ensure specific information is encoded into the representation vectors by getting the desired results from a linear transformation of the vector. The linear transformations shown in Figure 10 convert the representation vectors into vectors as D_H^S , D_R^S , D_T^S , and D^S which represent the structural information from the knowledge graph.

The vectors generated by the structural encoder should incorporate the structural information. Therefore a fully connected layer extracts these pieces of information from them. The Figure 10 and equation 1 explain the structural Question-set used in the proposed work. Here the vector C_R^S is transformed into binary vector D_R^S where its i^{th} element represents the connectivity of i^{th} entity with this relationship element. The vectors C_H^S , C_T^S are transformed into probability vector D_H^S , D_T^S respectively where its i^{th} element represents the connectivity score of i^{th} entity with this entity. The $^S D$ is the reference labeled ground truth for the corresponding vector as shown in Equation 1).

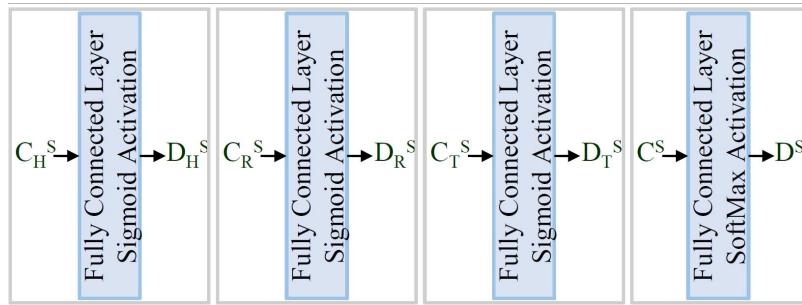


Figure 10. The Structural Question-set for encoding the structural information

$$\begin{aligned}
 {}^g_i D_H^S &= \begin{cases} e^{-spl}, & \text{spl is shortest path length} \\ & \text{between } i^{\text{th}} \text{ entity and this head} \\ 0, & \text{no connectivity} \end{cases} \\
 {}^g_i D_R^S &= \begin{cases} 1, & \text{if } i^{\text{th}} \text{ entity is connected with this relation} \\ 0, & \text{otherwise} \end{cases} \\
 {}^g_i D_T^S &= \begin{cases} e^{-spl}, & \text{spl is shortest path length} \\ & \text{between } i^{\text{th}} \text{ entity and this tail} \\ 0, & \text{no connectivity} \end{cases} \\
 {}^g D^S &= \text{one hot vector for corresponding entity}
 \end{aligned}$$

The cost function for the learning of the parameter of the structural encoder is based on the mean square error (MRR) function. As we have multiple questions set to encode the structural information, their corresponding losses are weighted to form the final cost (loss) of the encoder. The cost of the structural encoder (L^S) is given by Equation:1. Here the weight $s1, s2, s3$ are empirically set as $s1 = 0.3, s2 = 0.5, s3 = 0.3$.

$$\begin{aligned}
 L_{D_H^S} &= \frac{\sum_{i=0}^n ({}^g_i D_H^S - i D_H^S)^2}{n} \\
 L_{D_R^S} &= \frac{\sum_{i=0}^n ({}^g_i D_R^S - i D_R^S)^2}{n} \\
 L_{D_T^S} &= \frac{\sum_{i=0}^n ({}^g_i D_T^S - i D_T^S)^2}{n} \\
 L_{D^S} &= \frac{\sum_{i=0}^n ({}^g_i D^S - i D^S)^2}{n} \\
 L^S &= s1 \times L_{D_H^S} + s2 \times L_{D_R^S} + s3 \times L_{D_T^S} + L_{D^S}
 \end{aligned} \tag{1} \tag{2}$$

6.3. Interaction model

The proposed work utilizes the two interaction model learning schemes 1) supervised, and 2) unsupervised. The supervised interaction model learning scheme is used when we have labeled data available for training. Whereas the unsupervised interaction model learning does not have any label data. These two different learning approaches used different interaction models with some modifications.

$$S_{ij} = \frac{C_{ei}^F \cdot C_{ej}^F}{\|C_{ei}^F\| \cdot \|C_{ej}^F\|} \quad (3)$$

$$S_i^{max} = \max_{j=0}^n (s_{i0}, s_{i1}, s_{i2}, s_{in})$$

6.3.1. Supervised learning of interaction model

The interaction model used in the proposed work is similar to the BERT-INT (refer Figure 11).

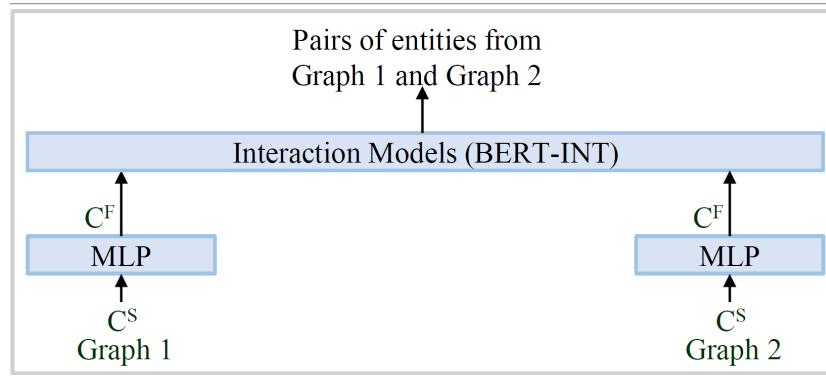


Figure 11. The interaction model for alignment of the entities of different graphs

All operations are the same except the calculation of the S_i^{max} (r BERT-INT). The original BERT-INT discarded the other similarities except the maximal one. The maximal similarity is given by Equation 3. Discarding other similarities is a waste of information, and we propose that they should be discarded after applying a softmax activation (refer Equation 4) across the row similarities. If we have similar entity pairs from graph 1 and graph 2 then we can maximize the corresponding $S_{ij}^{softmax}$ and then use $S_{ij}^{softmax}$ as the S_i^{max} for interaction model. But if the pair information is not available (i.e. we don't have the proper pairing between the entities) then the S_i^{max} should be replaced by S_i^{topsum} which is calculated by Equation:4. Here N is the number of top elements (having high $S_{ij}^{softmax}$). The value of N is dynamic in nature and decreases as the learning proceeds. We are decreasing the value of N by one after each epoch of learning till it becomes one.

$$S_{ij} = \frac{C_{ei}^F \cdot C_{ej}^F}{\|C_{ei}^F\| \cdot \|C_{ej}^F\|}$$

$$S_{ij}^{softmax} = \frac{e^{\alpha^2 s_{ij}}}{\sum_{j=0}^n e^{\alpha^2 s_{ij}}} \quad (4)$$

$$S_i^{TopN} = \{S_{ij}^{softmax} \mid S_{ij}^{softmax} \in \text{top N elements of } S_i \text{ row}\}$$

$$S_i^{topsum} = \sum_{S_{ij}^{softmax} \in S_i^{TopN}} (S_{ij}^{softmax})$$

$Loss_{interaction}$ = same as BERT-INT

6.3.2. Unsupervised learning of interaction model

The interaction model used for this scheme is different than the BERT-INT. Here we do not have the pair alignment information for the entities of KG 1 and KG 2. Therefore we need to reduce the trainable parameter of the interaction model as there is no validated gradient (corresponding to

the ground truth label) for parameter learning. The proposed work also does not utilize the dual aggregation technique for unsupervised learning as we don't want to use a trainable MLP for final classification. This new interaction model is defined by the Equation 5. Here as we do not have any alignment information available so we need to utilize only the implicit information of the different entities from different KGs. The two properties (assumption) we are exploiting for the learning are mentioned in 4.3.

$$\begin{aligned}
 Dist_{ij} &= C_i^F - C_j^F \\
 C_{ij}^{uniquemax} &= \frac{e^{\beta^2 Dist_{ij}}}{\sum_{i=0}^m e^{\beta^2 Dist_{ij}} + \sum_{j=0}^n e^{\beta^2 Dist_{ij}} - e^{\beta^2 Dist_{ij}}} \\
 S_{ij} &= C_{ei,ej}^{uniquemax} \\
 S_i^{TopN} &= \{S_{ij} \mid S_{ij} \in \text{top N elements of } S_i \text{ row}\} \\
 S_i^{topsum} &= \sum_{S_{ij} \in S_i^{TopN}} (S_{ij}) \\
 Loss_{interaction} &= \sum_{All} (1.0 - S_i^{topsum})
 \end{aligned} \tag{5}$$

7. Experimental Setup

7.1. Training Procedure

In this section, we elaborate on the training procedure used for experiments. We utilize Adam optimizer to train the proposed system with dynamic learning (exponentially and linear decreasing) rate setting. The learning rate is initialized to 0.001 and reduced to 10^{-4} in 25 thousand iterations with an exponentially decaying rate. After 25 thousand iterations, we operated a linearly decaying learning rate as equation 6. A total of one million iterations with 16 batch sizes is used to train the proposed system.

The learning stage also includes the L2 regularisation with a scale of 10^{-4} to limit the overfitting in the trained system.

$$lr = 10^{-4} \times (1.01 - \frac{iterationCount}{2500000}) \tag{6}$$

7.2. Evaluation Metric

Consistent with the previous works in literature, Hits@k (k=1, and 10) and mean reciprocal rank (MRR) are selected as the evaluation metrics in this paper. Hits@k calculates the proportion of correctly aligned entities ranked in the top-k list. Here, we focus on Hits@1 and Hits@10. MRR measures the average of the reciprocal ranks of the results. Outstanding methods should have higher Hits@k and MRR. Furthermore, during training 30-70% split of dataset is applied by consciously taking out the data of floor#4 to be used during validation.

7.3. Experiments Breakout

The empirical study for this work is designed on three different experiments shown in Figure 12.

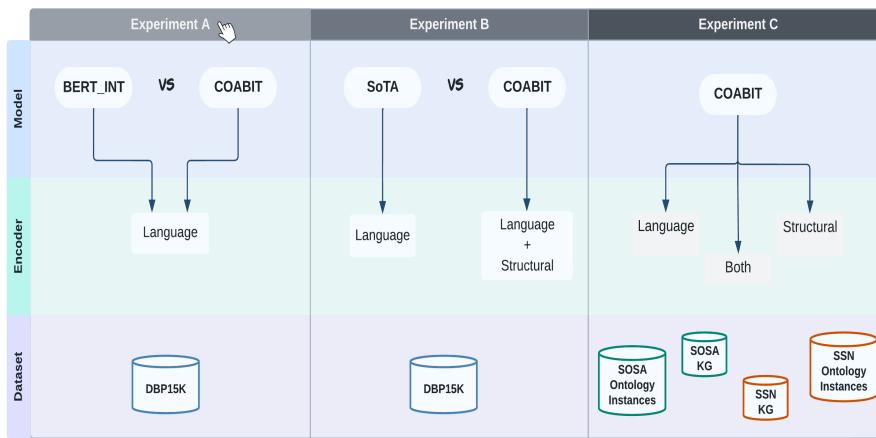


Figure 12. Layout of all experiments and the components used in them.

Experiments are designed from a systematically logical perspective. First, we conduct a comparative analysis of the baseline model with our proposed model. Next, we evaluate the performance of the proposed model in contrast to all state-of-the-art methods. Lastly, we conducted an ablation study on the proposed model to study its architecture's effectiveness. The selection of data set in each set of experiments is also mentioned in Figure 12.

8. Proof of Concept and Results

8.1. Improvement on SoTA (BERT_INT vs Proposed)

As discussed earlier, the proposed model is designed on a similar model of BERT-INT but with modifications explained in subsections 5.2, and 6.3. We extended the experiments of language encoder-based graph alignment conducted by Tang et al. [40] by using the same DBP15K dataset and similar BERT embedding setting and evaluate the results using the same parameters of HitRatio@K (K=1,10) and MRR. The modification of the language encoder is in updating the input arrangement shown in Figure 8b. The effectiveness of this input arrangement is also verified by incorporating it within BERT-INT as shown in Table 4). The table's first row exemplifies the BERT-INT model's performance improved when the proposed input arrangement is used. The second row states the results of the proposed model using only the proposed language encoder with the modified input arrangement. The results clearly show that even minor improvements beat the BERT-INT model. Moreover, we compare the complete proposed model (language + structural encoder) with all the state-of-the-art results presented in [40] in Table 5, and it is seen that the performance of the proposed model is highest among all by approximately 1.2-2.7%.

Table 4. Experiment A results of the performance of supervised entity alignment by BERT_INT method and its variant with proposed input arrangement on DBP15K dataset.

Method	DBP15K _{ZH-EN}			DBP15K _{JA-EN}			DBP15K _{FR-EN}		
	HR1	HR10	MRR	HR1	HR10	MRR	HR1	HR10	MRR
BERT-INT	96.8	99.0	97.7	96.4	99.1	97.5	99.2	99.8	99.5
Proposed	97.1	99.1	97.9	96.9	99.1	97.9	99.3	99.8	99.6

Table 5. Experiment B results on the overall performance of graph alignment on DBP15K dataset by all SoTA and proposed models.

Method	DBP15K _{ZH-EN}			DBP15K _{JA-EN}			DBP15K _{FR-EN}		
	HR1	HR10	MRR	HR1	HR10	MRR	HR1	HR10	MRR
Only use graph structures by variant TransE									
MTransE	30.8	61.4	36.4	27.9	57.5	34.9	24.4	55.6	33.5
IPTransE	40.6	73.5	51.6	36.7	69.3	47.4	33.3	68.5	45.1
BootEA	62.9	84.8	70.3	62.2	85.4	70.1	65.3	87.4	73.1
RSNs	50.8	74.5	59.1	50.7	73.7	59.0	51.6	76.8	60.5
TransEdge	73.5	91.9	80.1	71.9	93.2	79.5	71.0	94.1	79.6
MRPEA	68.1	86.7	74.8	65.5	85.9	72.7	67.7	89.0	75.5
Only use graph structures by variant TransE plus GCN									
MuGNN	49.4	84.4	61.1	50.1	85.7	62.1	49.5	87.0	62.1
NAEA	65.0	86.7	72.0	64.1	87.3	71.8	67.3	89.4	75.2
KECG	47.8	83.5	59.8	49.0	84.4	61.0	48.6	85.1	61.0
AliNet	53.9	82.6	62.8	54.9	83.1	64.5	55.2	85.2	65.7
Only use graph structures by variant TransE plus adversarial learning									
AKE	32.5	70.3	44.9	25.9	66.3	39.0	28.7	68.1	41.6
SEA	42.4	79.6	54.8	38.5	78.3	51.8	40.0	79.7	53.3
Combine graph structures and side information by variant GCN									
GCN-Align	41.3	74.4	54.9	39.9	74.5	54.6	37.3	74.5	53.2
GM-Align	67.9	78.5	-	74.0	87.2	-	89.4	95.2	-
RDGCN	70.8	84.6	74.6	76.7	89.5	81.2	88.6	95.7	91.1
HGCN	72.0	85.7	76.8	76.6	89.7	81.3	89.2	96.1	91.7
DGMC	77.2	89.7	-	77.4	90.7	-	89.1	96.7	-
Combine graph structures and side information by multi-view learning									
JAPE	41.2	74.5	49.0	36.3	68.5	47.6	32.4	66.7	43.0
MultiKE	50.9	57.6	53.2	39.3	48.9	42.6	63.9	71.2	66.5
JarKA	70.6	87.8	76.6	64.6	85.5	70.8	70.4	88.8	76.8
HMAN	87.1	98.7	-	93.5	99.4	-	97.3	99.8	-
CEAFF	79.5	-	-	86.0	-	-	96.4	-	-
BERT_INT	96.8	99.0	97.7	96.4	99.1	97.5	99.2	99.8	99.5
graph structural encoder in conjunction with language encoder									
Proposed	98.1	99.2	98.3	97.2	99.2	98.1	99.4	99.8	99.6

8.2. Quantitative Analysis with Ablation Study

To thoroughly investigate the effectiveness of proposed encoders, we conduct an ablation study on the proposed model. The dataset used for these experiments is the synthesized ontology dataset created from the Smart Building dataset of Kaggle as discussed in subsection 6 using SOSA and SSN ontology graphs. In Table 6, the first set of experiments are on *Synthetic SOSA - KG SSN* in which MMR score is highest when both encoders are used. For experiments of Only KG Structure, the Interaction Model is pre-trained on known ontology and uses the direct input embedding vector for the corresponding entity. However, the MRR score is lowest when only the structural encoder is used, which indicates that enforcing the graph structural information might have excluded all those alignment matches that were correct with respect to the language encoder but incorrect as per the ontology. A similar pattern is observed in other experiment sets as well. The last key observation is that the highest HRs' and MRR scores are achieved when *KG SOSA - Synthetic SSN* dataset was used. Our reflection from this is that SSN is a superset of SOSA so the model might have found all the correct alignments for every token of SOSA. Additionally, all alignment results had to be validated by annotations hand-picked by a human expert, as no bench-marking ontology alignment dataset is present. Although these results are subjective to the alignment annotations, they are significantly important because of their novelty.

8.3. Qualitative Analysis of proposed model

To visualize the alignments, we generate tsne plots of all the entities from both ontologies. First, we do indexing of all nodes and relations for both SOSA and SSN ontology. Then, lookup tables of entities are created. Next, we reduce the embedding vectors of all entities onto two-dimensional tsne plots as shown in Figure 14. Figure 13 demonstrates an alignment pair. Here, we magnify a pair of adjacent nodes from the alignment plot and follow their index in the lookup tables. We could see that both nodes were similar across the ontology; hence, they are aligned in the plot with the least Euclidean distance. Additionally, for further analysis of all the entities, the tsne plots are used to curate heat-maps by calculating the Euclidean distance map shown in Figures 15 and 16. These figures also show the learning of the model throughout iterations from 1000 to 62000th iteration. The heat maps show one-to-one mapping between pairs of SOSA and SSN nodes and relations, respectively. In the beginning, the model has almost learned no mapping, but the processing of loss functions continues; it starts identifying similar entities and those with lesser Euclidean distances between them are highlighted with lighter colors on the map.

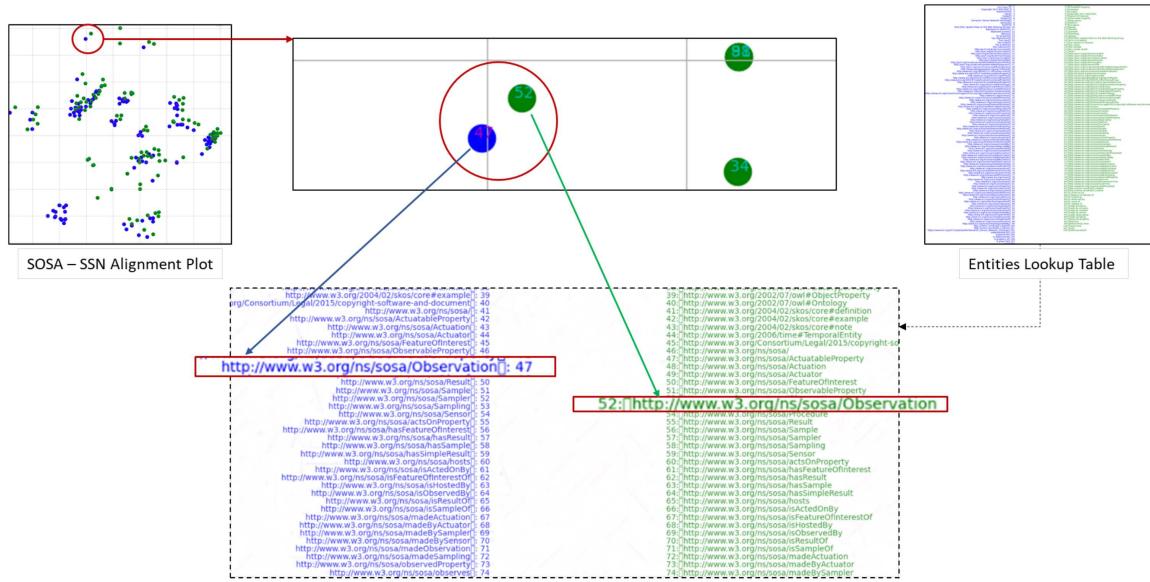


Figure 13. Ontology graph alignment pair demonstration. Entities in color blue represent SOSA graph nodes, and green represents SSN graph nodes. For clarity and ease in visualization, all SSN nodes in the alignment plot are shifted three spaces to the left.

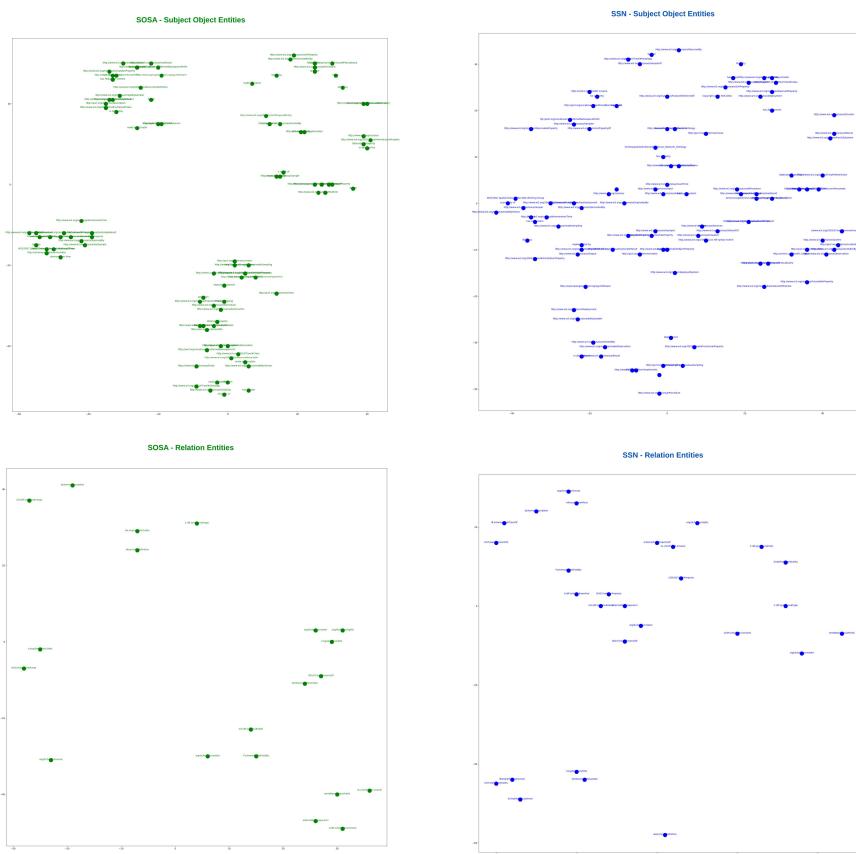


Figure 14. tsne plots generated from vectors of SOSA and SSN entities. An entity can be a node (subject or object) or a relation.

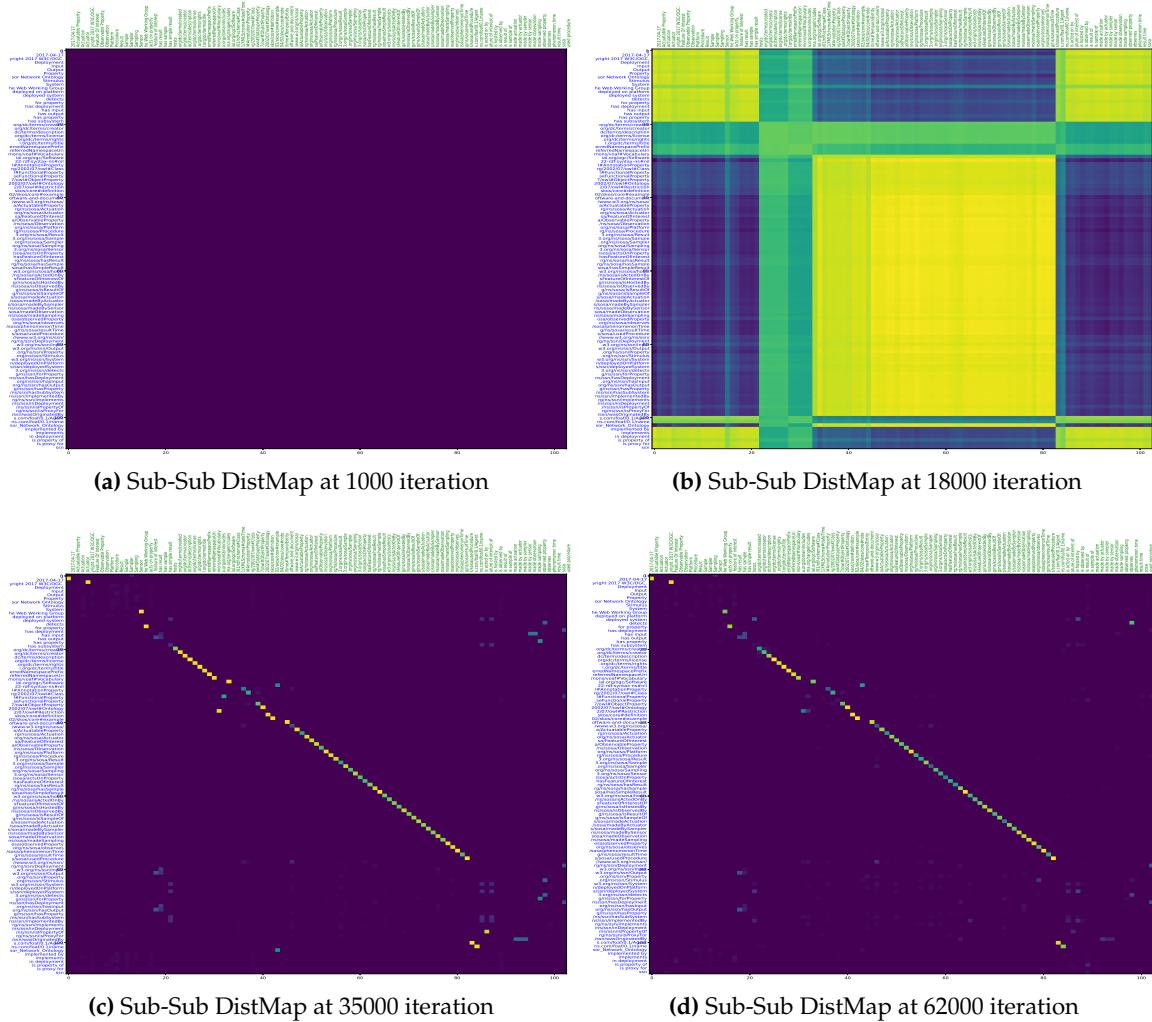


Figure 15. DistMap between Different Nodes of SOSA and SSN KGs

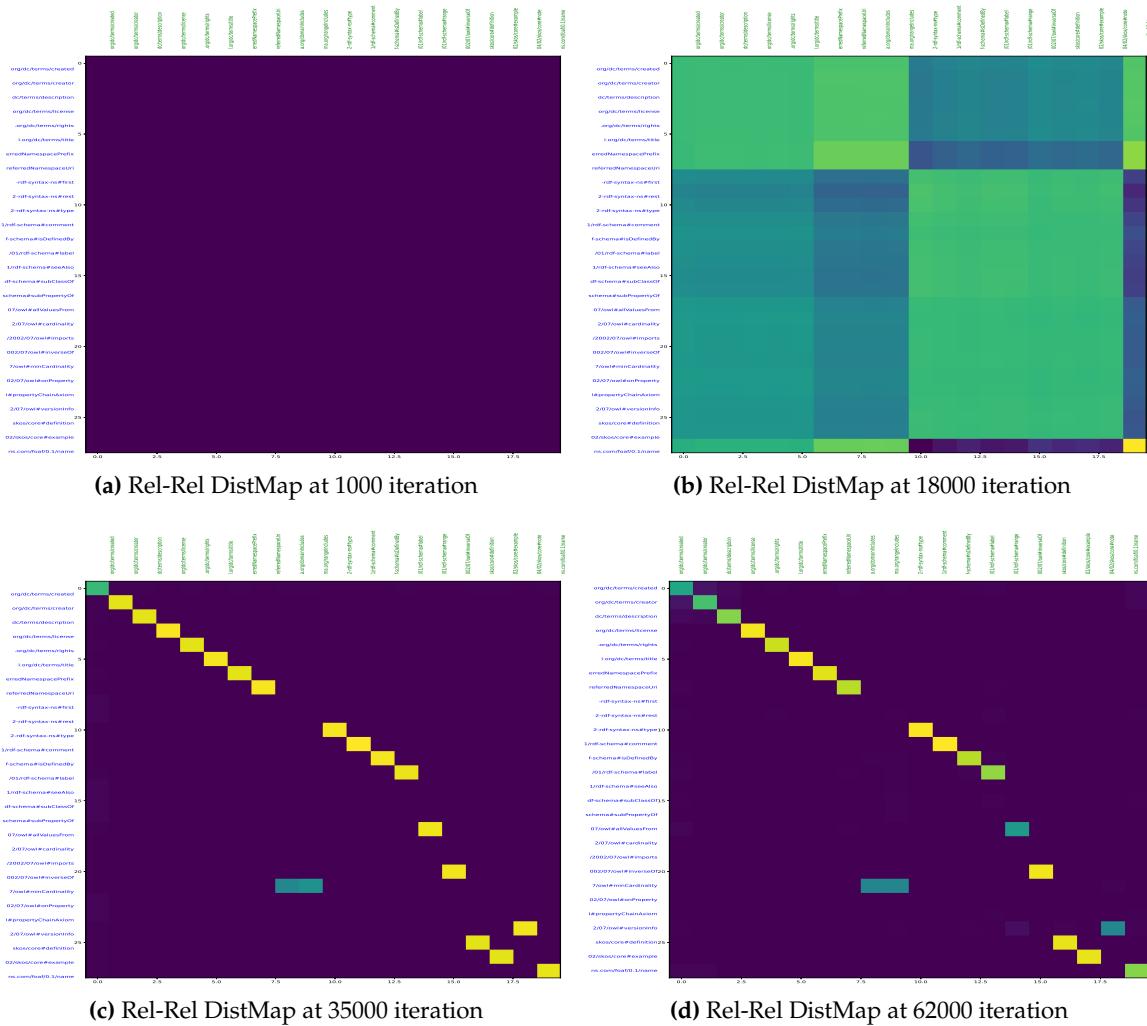


Figure 16. DistMap between Different Relations of SOSA and SSN KGs

9. Conclusion and Future Work

This paper is the first work to conceptualize ontology alignment for the Industrial Internet of Things (IIoT) domain based on a natural language processing (NLP) model for alignment among heterogeneous devices. The proposed model characterizes the ontology meta-data as side information and structure as the schema and learns vector embeddings for all entities and relations. Extensive experiments on both cross-lingual and cross-ontology tasks consistently outperform the baseline model BERT_INT model by 1.2-2.7% in HR and MRR scores. However, these results have few pertinent limitations. First, the ontology dataset had to be synthesized due to the lack of publicly available real-world smart sensor datasets. While language translation undoubtedly has a solid foundation, and large datasets are available for human language ontology, this is not true for the IIoT domain. Secondly, there is no bench-marking dataset available for establishing ground truth for IoT ontology alignment; therefore, the alignments between SSN and SOSA ontology were annotated by human experts. Although the results may be subjective to the alignment annotations, they are significantly important because of their novelty. Lastly, the ontology graphs of IoT ontology for sensor devices are very concise by design. The number of unique entities (nodes + relations) and triples in them are maximum in the hundreds as opposed to language ontology, which usually has thousands of nodes. For instance, SSN ontology has 125 unique entities, and SOSA has 75, so the accuracy results of correct alignments in Table 6 are as per the limited number of unique entities. Moreover, all the ontology for sensor devices is designed for similar types of devices functionally but with varying design principles.

Nevertheless, when the model learns language embeddings, it is easier to find nodes across ontologies that have labels with similar semantic meanings. To remove any such biases, the structure encoder is utilized to impose the context by correctly aligning only those nodes with matching labels and similar in-graphs (neighbors).

Table 6. Experiments C results of the ablation study using proposed ontology alignment model on smart building dataset with Unsupervised learning Approach).

Model Used		Result in Percentage		
Language Encoder (side information)	Structural Encoder (KG structure)	HR@1	HR@10	MRR
<i>Synthetic SOSA - KG SSN</i>				
✓	✓	87.6	94.3	89.5
✗	✓	81.9	88.6	82.8
✓	✗	83.8	92.4	84.8
<i>KG SOSA - KG SSN</i>				
✓	✓	80.9	91.4	83.8
✗	✓	75.2	88.6	77.1
✓	✗	77.1	90.5	79.0
<i>KG SOSA - Synthetic SSN</i>				
✓	✓	88.4	94.7	90.3
✗	✓	70.1	76.9	73.2
✓	✗	82.5	93.2	84.3

There are still several directions this work can potentially grow into. A generalized IoT ontology designed for any IoT device (beyond sensors) can be tested for ontology alignment to make an even stronger ablation study. One such ontology is SAREF [51], and it has approximately 1097 unique triples, the maximum among any IoT ontology. The next potential future work is that the paucity of benchmarking datasets can be resolved by conducting crowdsourcing ground truth to build validation data for IoT ontology alignment and annotations. There are public platforms such as BioPortal [52] for medical researchers that provide annotations for disparate biomedical ontologies. Inspired by this, IoT ontological resources must also be publicly provided for research to remove the bottlenecks of dataset limitations. Last but not least, as this work can be considered a step towards enabling translation between heterogeneous IoT sensor devices, the proposed model can be extended to a translation module in which, based on the ontology graphs of any device, the model can interpret the messages transmitted from that device. This idea is at an abstract level as of now and needs extensive efforts and empirical study to realize it fully.

Author Contributions: This work is a joint effort of contributors from two domains: 1) Machine Learning and 2) Cyber Physical Systems. Each author had a predefined task from the beginning of the research work. The CRediT Taxonomy is as follows:

- Saleha Javed: Conceptualization, Data Curation, Investigation, Methodology, Software, Validation, Visualization, Writing - original draft, review, and editing.
- Muhammad Usman: Validation, Writing - original draft, Writing - review and editing.
- Fredrik Sandin: Supervision, Resources, Validation.
- Marcus Liwicki: Supervision, Funding acquisition, Resources, Formal analysis, Validation.
- Hamam Mokayed: Project Administration, Supervision, Formal analysis, Writing - review, Validation.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mourtzis, D.; Angelopoulos, J.; Panopoulos, N. A Literature Review of the Challenges and Opportunities of the Transition from Industry 4.0 to Society 5.0. *Energies* **2022**, *15*, 6276.
2. Huang, S.; Wang, B.; Li, X.; Zheng, P.; Mourtzis, D.; Wang, L. Industry 5.0 and Society 5.0—Comparison, complementation and co-evolution. *Journal of manufacturing systems* **2022**, *64*, 424–428.

3. Usman, M.; Sarfraz, M.S.; Habib, U.; Aftab, M.U.; Javed, S. Automatic Hybrid Access Control in SCADA-Enabled IIoT Networks Using Machine Learning. *Sensors* **2023**, *23*. doi:10.3390/s23083931.
4. Javed, S.; Javed, S.; Deventer, J.v.; Mokayed, H.; Delsing, J. A Smart Manufacturing Ecosystem for Industry 5.0 using Cloud-based Collaborative Learning at the Edge. NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium, 2023, pp. 1–6. doi:10.1109/NOMS56928.2023.10154323.
5. Tumiwa, J.R.; Tuegeh, O.; Bittner, B.; Nagy, A. The challenges to developing smart agricultural village in the industrial revolution 4.0.: The case of indonesia. *Torun International Studies* **2022**, *1*, 25–45.
6. Javed, S.; Javed, S.; van Deventer, J.; Sandin, F.; Delsing, J.; Liwicki, M.; Martin-del Campo, S. Cloud-based Collaborative Learning (CCL) for the Automated Condition Monitoring of Wind Farms. 2022 IEEE 5th International Conference on Industrial Cyber-Physical Systems (ICPS), 2022, pp. 1–8. doi:10.1109/ICPS51978.2022.9816960.
7. Ryalat, M.; ElMoaqet, H.; AlFaouri, M. Design of a smart factory based on cyber-physical systems and Internet of Things towards Industry 4.0. *Applied Sciences* **2023**, *13*, 2156.
8. Ullah, F. Smart Tech 4.0 in the Built Environment: Applications of Disruptive Digital Technologies in Smart Cities, Construction, and Real Estate, 2022.
9. Grigoriadis, I.; Vrochidou, E.; Tsatsiou, I.; Papakostas, G.A. Machine Learning as a Service (MLaaS)—An Enterprise Perspective. Proceedings of International Conference on Data Science and Applications: ICDSA 2022, Volume 2. Springer, 2023, pp. 261–273.
10. Nilsson, J.; Sandin, F.; Delsing, J. Interoperability and machine-to-machine translation model with mappings to machine learning tasks. 2019 IEEE 17th International Conference on Industrial Informatics (INDIN). IEEE, 2019, Vol. 1, pp. 284–289.
11. van, O. Financesonline official website. <https://financesonline.com/number-of-internet-of-things-connected-devices>, 2022 (accessed January 7, 2022).
12. Nilsson, J. System of systems interoperability machine learning model. PhD thesis, Luleå University of Technology, 2019.
13. Paniagua, C.; Eliasson, J.; Delsing, J. Interoperability mismatch challenges in heterogeneous soa-based systems. 2019 IEEE International Conference on Industrial Technology (ICIT). IEEE, 2019, pp. 788–793.
14. Bengio, Y.; Courville, A.; Vincent, P. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* **2013**, *35*, 1798–1828.
15. Tim Berners-Lee, J.J. W3C Standards. <https://www.w3.org/standards/>, 2022.
16. Bizer, C.; Heath, T.; Berners-Lee, T. Linked data: The story so far. In *Semantic services, interoperability and web applications: emerging concepts*; IGI global, 2011; pp. 205–227.
17. Delsing, J. Smart City Solution Engineering. *Smart Cities* **2021**, *4*, 643–661.
18. Honti, G.M.; Abonyi, J. A review of semantic sensor technologies in internet of things architectures. *Complexity* **2019**, 2019.
19. Le-Phuoc, D.; Quoc, H.N.M.; Quoc, H.N.; Nhat, T.T.; Hauswirth, M. The graph of things: A step towards the live knowledge graph of connected things. *Journal of Web Semantics* **2016**, *37*, 25–35.
20. Janowicz, K.; Haller, A.; Cox, S.J.; Le Phuoc, D.; Lefrançois, M. SOSA: A lightweight ontology for sensors, observations, samples, and actuators. *Journal of Web Semantics* **2019**, *56*, 1–10. doi:10.1016/j.websem.2018.06.003.
21. Compton, M.; Barnaghi, P.; Bermudez, L.; Garcia-Castro, R.; Corcho, O.; Cox, S.; Graybeal, J.; Hauswirth, M.; Henson, C.; Herzog, A.; others. The SSN ontology of the W3C semantic sensor network incubator group. *Journal of Web Semantics* **2012**, *17*, 25–32.
22. Moutinho, F.; Paiva, L.; Köpke, J.; Maló, P. Extended semantic annotations for generating translators in the arrowhead framework. *IEEE Transactions on Industrial Informatics* **2017**, *14*, 2760–2769.
23. Campos-Rebelo, R.; Moutinho, F.; Paiva, L.; Maló, P. Annotation rules for xml schemas with grouped semantic annotations. IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society. IEEE, 2019, Vol. 1, pp. 5469–5474.
24. Halevy, A.; Norvig, P.; Pereira, F. The unreasonable effectiveness of data. *IEEE intelligent systems* **2009**, *24*, 8–12.
25. Nilsson, J.; Sandin, F.; Delsing, J. Interoperability and machine-to-machine translation model with mappings to machine learning tasks. 2019 IEEE 17th International Conference on Industrial Informatics (INDIN), 2019, Vol. 1, pp. 284–289. doi:10.1109/INDIN41052.2019.8972085.

26. Kolyvakis, P.; Kalousis, A.; Kiritsis, D. Deepalignment: Unsupervised ontology matching with refined word vectors. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), 2018, pp. 787–798.

27. Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; Taylor, J. Freebase: a collaboratively created graph database for structuring human knowledge. Proceedings of the 2008 ACM SIGMOD international conference on Management of data, 2008, pp. 1247–1250.

28. Carlson, A.; Betteridge, J.; Kisiel, B.; Settles, B.; Hruschka, E.R.; Mitchell, T.M. Toward an architecture for never-ending language learning. Twenty-Fourth AAAI conference on artificial intelligence, 2010.

29. Lin, X.; Yang, H.; Wu, J.; Zhou, C.; Wang, B. Guiding cross-lingual entity alignment via adversarial knowledge embedding. 2019 IEEE International Conference on Data Mining (ICDM). IEEE, 2019, pp. 429–438.

30. Cao, Y.; Liu, Z.; Li, C.; Li, J.; Chua, T.S. Multi-channel graph neural network for entity alignment. *arXiv preprint arXiv:1908.09898* 2019.

31. Wu, Y.; Liu, X.; Feng, Y.; Wang, Z.; Zhao, D. Neighborhood matching network for entity alignment. *arXiv preprint arXiv:2005.05607* 2020.

32. Liu, W.; Zhou, P.; Zhao, Z.; Wang, Z.; Ju, Q.; Deng, H.; Wang, P. K-bert: Enabling language representation with knowledge graph. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, Vol. 34, pp. 2901–2908.

33. Chen, X.; Jia, S.; Xiang, Y. A review: Knowledge reasoning over knowledge graph. *Expert Systems with Applications* 2020, 141, 112948.

34. Wang, H.; Zhao, M.; Xie, X.; Li, W.; Guo, M. Knowledge graph convolutional networks for recommender systems. The world wide web conference, 2019, pp. 3307–3313.

35. Galárraga, L.; Razniewski, S.; Amarilli, A.; Suchanek, F.M. Predicting completeness in knowledge bases. Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, 2017, pp. 375–383.

36. Chen, M.; Tian, Y.; Yang, M.; Zaniolo, C. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. *arXiv preprint arXiv:1611.03954* 2016.

37. Wang, Z.; Lv, Q.; Lan, X.; Zhang, Y. Cross-lingual knowledge graph alignment via graph convolutional networks. Proceedings of the 2018 conference on empirical methods in natural language processing, 2018, pp. 349–357.

38. Xu, K.; Wang, L.; Yu, M.; Feng, Y.; Song, Y.; Wang, Z.; Yu, D. Cross-lingual knowledge graph alignment via graph matching neural network. *arXiv preprint arXiv:1905.11605* 2019.

39. Trisedya, B.D.; Qi, J.; Zhang, R. Entity alignment between knowledge graphs using attribute embeddings. Proceedings of the AAAI Conference on Artificial Intelligence, 2019, Vol. 33, pp. 297–304.

40. Tang, X.; Zhang, J.; Chen, B.; Yang, Y.; Chen, H.; Li, C. BERT-INT: a BERT-based interaction model for knowledge graph alignment. Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, 2021, pp. 3174–3180.

41. Yang, J.; Wang, D.; Zhou, W.; Qian, W.; Wang, X.; Han, J.; Hu, S. Entity and Relation Matching Consensus for Entity Alignment. Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021, pp. 2331–2341.

42. Lieberman, M. Linguistic Data Consortium 2019.

43. Lhoest, Q.; del Moral, A.V.; Jernite, Y.; Thakur, A.; von Platen, P.; Patil, S.; Chaumond, J.; Drame, M.; Plu, J.; Tunstall, L.; others. Datasets: A community library for natural language processing. *arXiv preprint arXiv:2109.02846* 2021.

44. Lehmann, J.; Isele, R.; Jakob, M.; Jentzsch, A.; Kontokostas, D.; Mendes, P.N.; Hellmann, S.; Morsey, M.; Van Kleef, P.; Auer, S.; others. Dbpedia—a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web* 2015, 6, 167–195.

45. Rebele, T.; Suchanek, F.; Hoffart, J.; Biega, J.; Kuzey, E.; Weikum, G. YAGO: A multilingual knowledge base from wikipedia, wordnet, and geonames. International semantic web conference. Springer, 2016, pp. 177–185.

46. Sun, Z.; Hu, W.; Zhang, Q.; Qu, Y. Bootstrapping Entity Alignment with Knowledge Graph Embedding. IJCAI, 2018, Vol. 18, pp. 4396–4402.

47. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805* **2018**.
48. Haller, A. Semantic Sensor Network - W3C. <https://www.w3.org/TR/vocab-ssn/#apartment-134>, 2022 (accessed March 20, 2022).
49. Hong, D. Smart Buidling System. <https://www.kaggle.com/datasets/ranakrc/smart-building-system>, 2022 (accessed March 26, 2022).
50. Hong, D.; Gu, Q.; Whitehouse, K. High-dimensional time series clustering via cross-predictability. *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 642–651.
51. of Trento, K.U. Data Scientia. http://liveschema.eu/dataset/lov_saref/resource/f316a18f-c295-4cda-b6cf-d073f5a6b4be, 2022 (accessed April 6, 2022).
52. for Biomedical Ontology, N.C. BioPortal. <https://bioportal.bioontology.org/mappings>, 2022 (accessed April 6, 2022).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.