Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 218 (2023) 367–374

Procedia
Computer Science

www.elsevier.com/locate/procedia

International Conference on Machine Learning and Data Engineering

WSD based Ontology Learning from Unstructured Text using Transformer

Akshay Hari[1], Priyanka Kumar[1,*]

Department of Computer Science and Engineering, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India

Abstract

Representation of knowledge and making it machine comprehensible has become a necessity in modern times but with the large amount of data being generated nowadays, this process has to be automated as much as possible. In this work, we propose a deep-learning based model to build an RDF based Ontology from Unstructured Text. We aim to evaluate the proposed model by creating a general knowledge ontology from newspaper article corpora. The proposed model is based on transformer, Natural Language Processing and contains a Relation Extraction model and novel implementation of RDF mapping algorithm. The main highlight of our model is its ability to handle the Word Sense Disambiguation problem. The model was able to perform well and achieved very high accuracy scores.

Keywords: Deep Learning; Transformers; Ontology; Word Sense Disambiguation; RDF

1. Introduction

Easy accessibility of information is one of the key requirements in the modern era. We often access information through an electronic device such as computer or mobile phones from the internet. One of the main advantages of this approach is that information can be manipulated as per our requirements, say in the graphical form etc. But for this to happen, the data should be in a machine readable and understandable format. Here is where the Semantic Web comes into the picture, which can be attributed as a formal or standardized way to represent data or knowledge.

However, with the vast volume of data being generated everyday it would be a tedious task to manually convert all these unstructured textual data into standardized format for Ontology Engineering. Another challenge is the ambiguity of the words or the homonyms as this is often related to the Word Sense Disambiguation problem and some of the

[*] Corresponding author.
E-mail address: cb.en.p2aid20009@cb.students.amrita.edu (Akshay Hari), k_priyanka@cb.amrita.edu (Priyanka Kumar)

10.1016/j.procs.2023.01.019

Ontology learning models does not address this issue. Also, most of the current Ontology models are built based on older Deep Learning architectures and word embedding models.

In this work, we propose an approach to build an Ontology from Unstructured Text. Our work consists of two parts: first is the extraction of relational triples from unstructured text using a deep-learning based transformer model called RELATION EXTRACTOR [6] and second is the mapping of the extracted triples into RDF format using novel RDF mapping module called RDF MAPPER.

The main research contributions of this work are as follows:

- We propose a novel RDF mapping for generating the URI which is based on contextual embeddings from transformer. Most of the current ontology models use older deep learning models and static word embeddings within their architecture.
- Our model takes account of Word Sense Disambiguation problem ensuring that even if two words are spelled the same, they will have correct URI based on their context.

The content of this paper is organized as follows. Sub-section 1.1 gives a general idea of the relation extraction component used in our work. Sub-section 1.2 explains about the Ontology and standards used in our work. The Word Sense Disambiguation problem and how contextual embeddings could be useful to solve this problem is explained in sub-section 1.3. In section 2, we covers literature survey of related works. Section 3 explain the architecture of our model and section 4 covers about the dataset used for evaluation. In section 5, we analyze our results and section 6 concludes our work with future scope and areas for improvements.

1.1. Relation Extraction

Most of the sentences contain entities (often nouns) within them and these entities have a relationship between them. The process of extracting the relationship between the entities from a sentence is called Relation Extraction and it is one of the problems in Natural Language Processing. The extracted relation will be in the form of subject – predicate – object (s-p-o) which is called relational triples. The relationship between the subject and object is defined by the predicate. For the relation extraction task in our work, we use the relation extraction model from our previous work [6]. This is a joint entity-relation extraction model and based on DistilBERT [13] based transformer language model and can handle scenarios such as Single Entity Overlapping (SEO) and Entity pair Overlapping (EPO) which are the cases where one or more entities being part of multiple relational triples. The conversion of Unstructured text to Relational Triples are done at this stage.

1.2. Ontology

To represent the extracted relational triples in Ontology framework, we would need a standardized ontology format. Such a framework is Resource Description Framework or RDF, which is a standardized model for data transfer in the semantic web [10]. The RDF format also has data represented in the triple form, but there are some key differences between the triples from relation extraction and the RDF triples. The first one is the presence of Uniform Resource Identifiers (URI) which uniquely identifies the entity as opposed to the triplets from the relation extraction. In relation extraction, the homonym entities are treated as one and will have the same name. In this work, the RDF mapper takes care of the URI assigning and Word Sense Disambiguation (WSD).

1.3. Contextual Embedding and WSD

As mentioned in the previous section, the homonym entities are treated as a single entity in relation extraction task. Here is where Word Sense Disambiguation comes into picture as we want to distinguish between the homonym words to assign the appropriate URI for the Resource Description framework. In this proposed work, the RDF mapper is based on transformer-based language model and we could use this to disambiguate words. In the older word embedding models, the embeddings are static in nature i.e., words with same spelling but different meaning will have same embedding vector. For transformer-based language models, embeddings are contextual in nature i.e., embedding vector of a word is determined by the context of the sentence and words around it rather than by the spelling of the

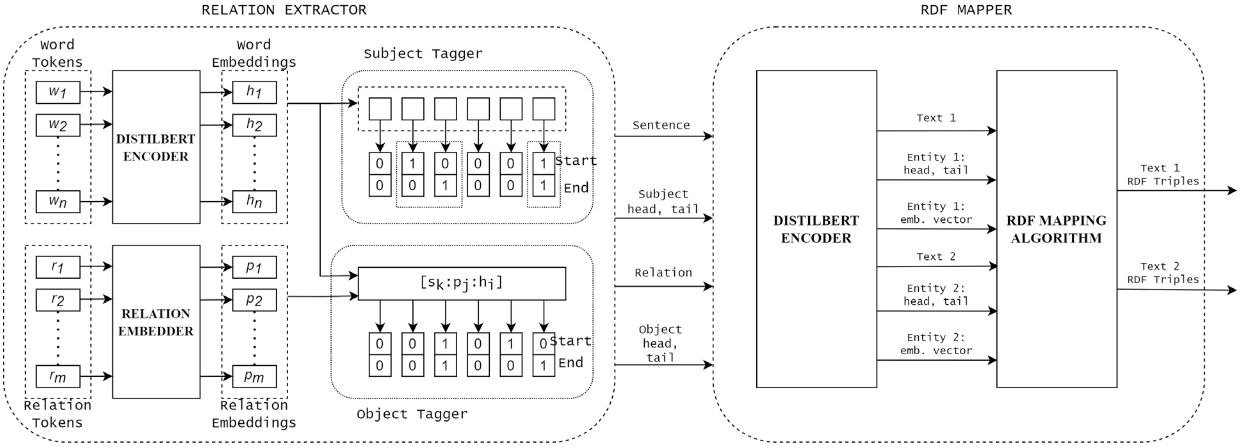


Fig. 1. Architecture of the proposed model. RELATION EXTRACTOR from [6]

word. This feature is very useful in dealing with Word Sense Disambiguation. In the proposed work, we are using the contextual embeddings from the transformers and similarity measures to compare the meaning of the entities extracted by RELATION EXTRACTOR.

2. Related Works

Ontology Learning is a vast area and there are multiple approaches and classifications on the type of learning. These includes scope of the knowledge it covers i.e., whether it is general ontology or domain ontology, type of specification language used such as web-based languages like RDF, OWL or traditional representations like First-Order Logic etc. Another classification could be based on the approach used for Ontology Learning such as Linguistic or Statistical approach. Detailed information about these classifications, approaches and comparative studies are covered in [8].

One of the earlier works related to Ontology learning from text is TEXT-TO-ONTO [9], a semi-automated approach to build domain ontology. Later [2] proposed a probabilistic ontology model and NLP to build ontology. In the work of [7], proposed a domain ontology learning system based on linguistic and statistical based approach with NLP techniques such as POS tagging, Berkley Parser. In [1], they proposed a model with NLP pre-processing techniques, Named Entity Recognition, WSD etc to generate RDF linked to DBPedia and WordNET. Another RDF based work for Ontology Learning is [5]. An application of Ontology Learning from heterogeneous data resources in the field of medicine is proposed in the work of [15] .In paper[14], the authors proposes an ontology learning approach with POS tagging and clustering. In the work of [11], the author proposed an ontology model using word2vec embedding, semantic similarities approach and k-means clustering.

3. Model Architecture

The architecture of the proposed model is given in Figure 1. The architecture contains combined diagram of Relation Extractor and the RDF mapper. The relation extractor is a supervised transformer based deep learning model which gives relational triples as output for a given sentence. The relation extractor is based on our previous work and its detailed working can be found here [6].

A brief overview of the relation extractor is as follows: for the input sentence given, the distilbert encoder extracts each word's token and for pre-defined relations, word embeddings and relation embeddings are created. Now for the main relation extraction task, subject and object taggers are used. The subject tagger will identify all the possible subjects among the all the words in the sentence. Similarly, the object tagger will use words token apart from the ones used by the subject tagger. The mathematical representation of the likelihood function for the relation extractor is given in equation 1.

$$\begin{aligned} & \prod_{(s,r,o)\in T} p((s,r,o) \mid x) \\ &= \prod_{s\in T} p(s \mid x) \prod_{(r,o)\in T|s} p((r,o) \mid x, s) \\ &= \prod_{s\in T} p(s \mid x) \prod_{r\in T|s} p(o \mid x, s, r) \prod_{r\in R\backslash T|s} p\left(o_{\varnothing} \mid x, s, r\right) \end{aligned} \tag{1}$$

where $T \mid s$ is the triplet set with s as subject in training data T. Similarly, $(r, o) \in T \mid s$ is the set of all relation-object pair in T. R is the set of all relations and $R\backslash T \mid s$ means all the relations except subject s in T. $o_{\varnothing}$ represents all relations except those in triplet $T \mid s$ will have no corresponding objects [6].

From the relation extractor, we extract relational triples as well as beginning and ending of the token positions of subject and object in the sentence as output. Now we need to convert the relational triples into RDF triples. For this purpose, we have designed an RDF mapper. The working of the RDF mapper is explained as follows.

First, we need to combine all the subjects and the objects into a single entity because this approach will be easier for assigning URI and in our RDF mapper, the subject and object triplet have same RDF namespace. We also need to make sure that these combined entities should be able to converted back into corresponding subject, predicate, and object in final RDF triples output stage. This is the reason we are using the beginning and ending token positions of the subject and objects from the previous stage as these values will act as a primary key for the values in the combined entity table. These values are then passed into a DistilBERT encoder for obtaining the word embedding vectors of the entities.

Now the URI for the entities are generated by the RDF mapping algorithm explained in algorithm 1. The working of the algorithm can be summarized as follows. We are taking two entities – Entity 1 and Entity 2 and checking whether they have same name or not. If they have different name, we are assigning different URI. But if two entities have same name, we need to check whether they have the same meaning and context or not. This can be done by checking the cosine similarity of the word level embedding vectors of the entity names.

Algorithm 1 RDF mapping algorithm

```
 1: $N \leftarrow size(entity)$
 2: for $i = 1, 2, \ldots$ do
 3:     for $j = 1, 2, \ldots, N - i$ do
 4:         if $i = N - j$ then
 5:             go to 3
 6:         end if
 7:         if $entity.name(i) \neq entity.name(N - j)$ then
 8:             ASSIGN_DIFFERENT_URI($entity[i], entity[N - j]$)
 9:         else if  then
10:             $distance$ = CHECK_DISTANCE($entity.name[i], entity.name[N - j]$
11:             if $distance < threshold$ then
12:                 ASSIGN_DIFFERENT_URI($entity[i], entity[N - j]$)
13:             else if  then
14:                 ASSIGN_SAME_URI($entity[i], entity[N - j]$)
15:             end if
16:         end if
17:     end for
18: end for
```

$$cosineSimilarity(x, y) = \frac{x \cdot y}{||x|| \cdot ||y||} \tag{2}$$

Cosine similarity as defined in equation 2 is a useful measure for checking the similarities of the vector embeddings in NLP related tasks. In the equation 2 x and y corresponds to the embedding vectors of Entity 1 and Entity 2 from the final layer of the DistilBERT encoder. This is implemented in the CHECK_DISTANCE() function in algorithm 1. The *threshold* is a cut-off value which determines whether the entity should have same URI or not. This value is usually set high and arbitrarily. The functions ASSIGN_SAME_URI and ASSIGN_DIFFERENT_URI() assigns URI to the entities based on various conditions like whether they already have URI or not, adding and modifying serial numbers of already assigned URI etc. After the above steps, we would have all the entities with RDF URI assigned.

4. Dataset

For the evaluation of the model, we are using New York Times dataset [12] and WebNLG dataset [4]. The datasets are originally created for the Relation Extraction task but can be used for our work as well to an extent. Both datasets have samples for Single Entity Overlapping and Entity Pair Overlapping scenarios. They also have samples with up to 5 triples in a single sentence. Although the dataset has entities having same meaning in different sentences, they do not have any homonym entities. Therefore, for the testing purpose, an extra sentence with an entity which is in different context with other entities was added.

5. Result Analysis

The result from the relation extractor is mentioned in Table 1. The results are mostly same from our previous work and for the newly added sample, model was able to predict triple correctly. This output from the relation extractor is then passed to the RDF mapper which assigns the URI. A word cloud of the extracted entities is shown in Figure 2. This gives us overall idea of extracted Entities.

Table 1. Result of the Relation Extractor

Dataset	F1-score	Precision	Recall
NYT	89.66	90.22	89.10
WebNLG	88.95	89.86	88.05

An analysis of cosine similarities of two entities with same name from NYT and WebNLG datasets is given in Figure 3. The x – axis in the histogram refers to the cosine similarity of Entity 1 and Entity 2 and y-axis refers to the count. Both of the histograms are left-skewed which is working as expected because in our dataset, except for the newly added sample, all of the Entity-1 and Entity-2 names have same meaning hence have high cosine similarity. Since the threshold value set for NYT dataset and WebNLG dataset are 0.75 and 0.8 respectively, all the values below this threshold are actually errors except for the newly added sample which should be present below the threshold value.

A more detailed analysis of cosine similarity score comparison can be drawn from sample dataframe Figure 4 which is taken from the intermediate output of the RDF mapper. In the dataframe table, for the *entity*1, first three rows refer to ‘Baked Alaska’ which is a type of food whereas *entity*2 refers to Alaskan state in the United State of America. Therefore, the cosine similarity score of these two entities will be low. For the other rows, *entity*1 and *entity*2 refers to the Alaskan state and hence we have high cosine similarity score. The RDF mapper has taken account of all these values and have successfully created two URI for Alaska. This sample output dataframe is shown in Figure 5. Here we can observe that our model has assigned URI for state of Alaska as ’Alaska000’ and food Alaska as ’Alaska001’

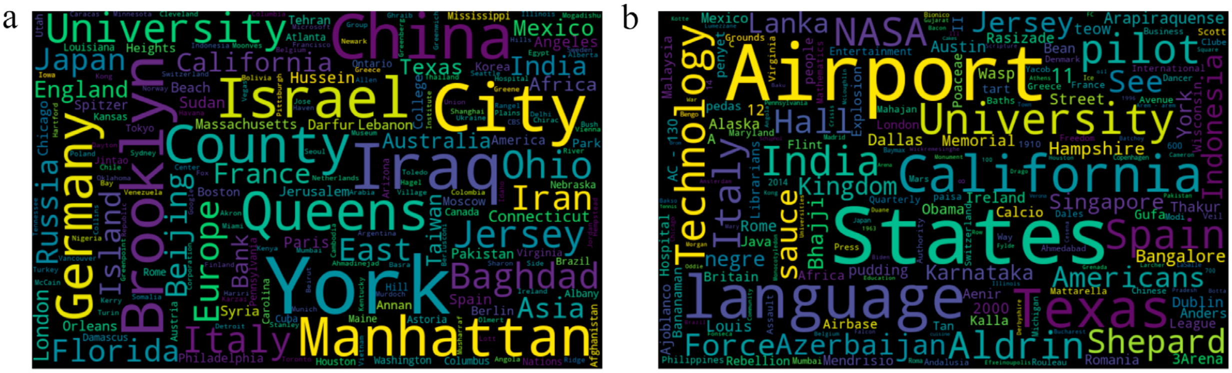


Fig. 2. Wordcloud of Entities from results of (a) NYT dataset; (b) WebNLG dataset.

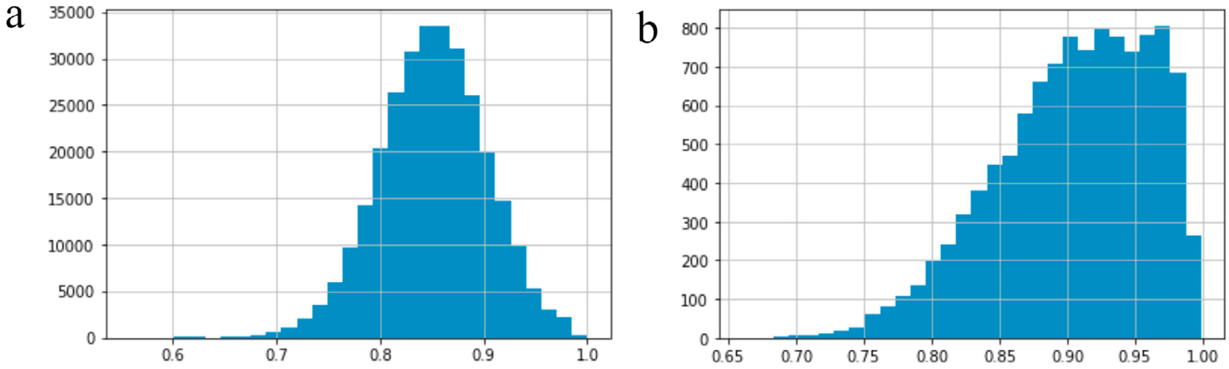


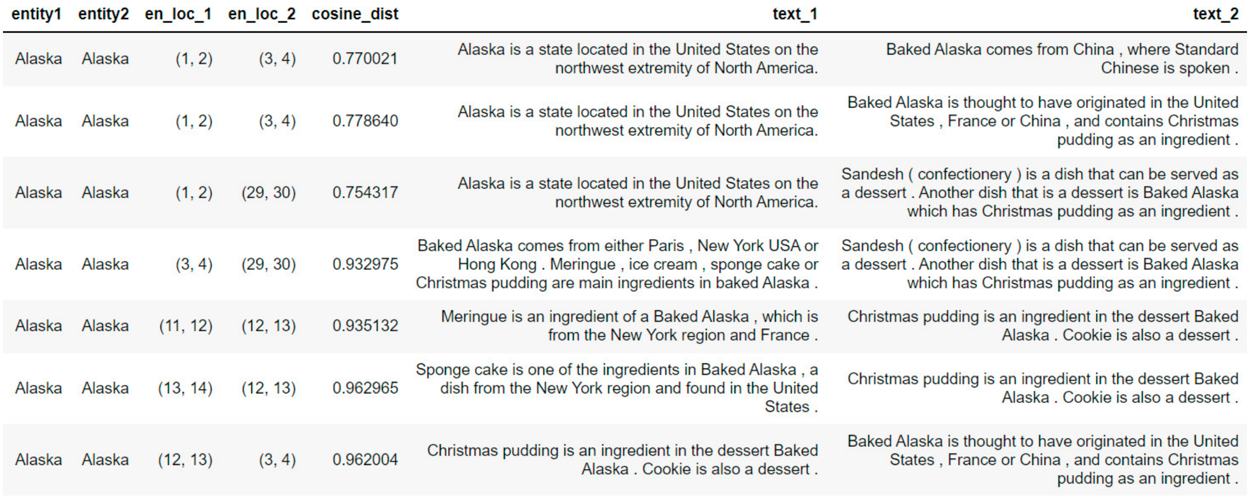
Fig. 3. Cosine Similarity comparison of entities with same name and context from (a) NYT dataset; (b) WebNLG dataset.

Note that in both figure 4 and 5, *en_loc* is the position of the tokenized word from transformer not the position of the word in sentence.

Table 2. Result of the RDF mapper

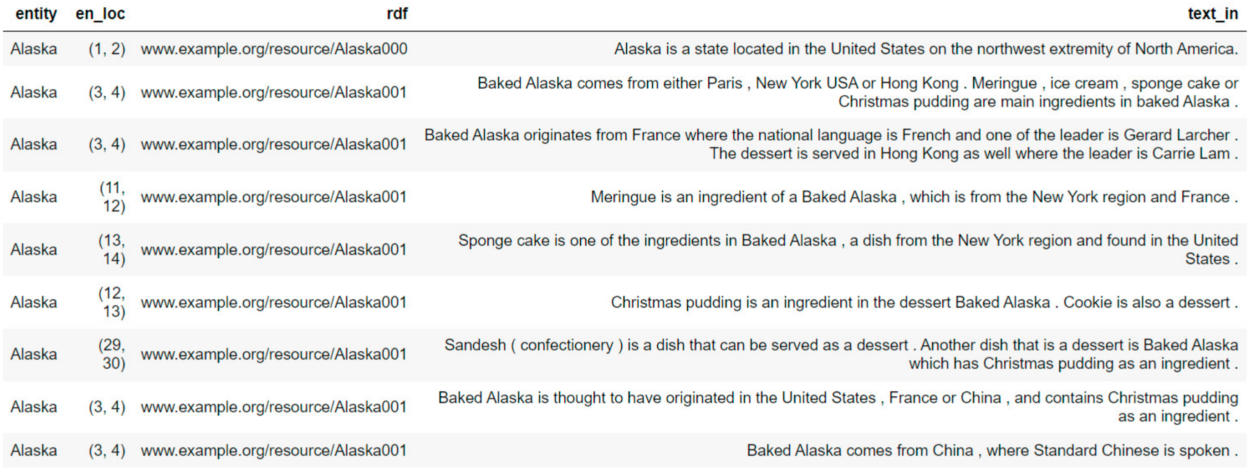
Summary	NYT dataset	WebNLG dataset
Total number of entities	10852	2127
Entities with correct RDF assigned	10382	2089
Unique RDF created	2385	587
Error RDF Created	20	38
Accuracy of Entities with correct RDF	99.81%	98.21%
Accuracy of Unique RDF created	99.16%	93.52%

The final results from the RDF mapper is tabulated in the Table 2. From the result, we can see that our RDF mapper has been able to correctly assign URI's to majority of the entities.



entity1	entity2	en_loc_1	en_loc_2	cosine_dist	text_1	text_2
Alaska	Alaska	(1, 2)	(3, 4)	0.770021	Alaska is a state located in the United States on the northwest extremity of North America.	Baked Alaska comes from China , where Standard Chinese is spoken .
Alaska	Alaska	(1, 2)	(3, 4)	0.778640	Alaska is a state located in the United States on the northwest extremity of North America.	Baked Alaska is thought to have originated in the United States , France or China , and contains Christmas pudding as an ingredient .
Alaska	Alaska	(1, 2)	(29, 30)	0.754317	Alaska is a state located in the United States on the northwest extremity of North America.	Sandesh (confectionery) is a dish that can be served as a dessert . Another dish that is a dessert is Baked Alaska which has Christmas pudding as an ingredient .
Alaska	Alaska	(3, 4)	(29, 30)	0.932975	Baked Alaska comes from either Paris , New York USA or Hong Kong . Meringue , ice cream , sponge cake or Christmas pudding are main ingredients in baked Alaska .	Sandesh (confectionery) is a dish that can be served as a dessert . Another dish that is a dessert is Baked Alaska which has Christmas pudding as an ingredient .
Alaska	Alaska	(11, 12)	(12, 13)	0.935132	Meringue is an ingredient of a Baked Alaska , which is from the New York region and France .	Christmas pudding is an ingredient in the dessert Baked Alaska . Cookie is also a dessert .
Alaska	Alaska	(13, 14)	(12, 13)	0.962965	Sponge cake is one of the ingredients in Baked Alaska , a dish from the New York region and found in the United States .	Christmas pudding is an ingredient in the dessert Baked Alaska . Cookie is also a dessert .
Alaska	Alaska	(12, 13)	(3, 4)	0.962004	Christmas pudding is an ingredient in the dessert Baked Alaska . Cookie is also a dessert .	Baked Alaska is thought to have originated in the United States , France or China , and contains Christmas pudding as an ingredient .

Fig. 4. Sample Cosine Similarity comparison of entity with same name.



entity	en_loc	rdf	text_in
Alaska	(1, 2)	www.example.org/resource/Alaska000	Alaska is a state located in the United States on the northwest extremity of North America.
Alaska	(3, 4)	www.example.org/resource/Alaska001	Baked Alaska comes from either Paris , New York USA or Hong Kong . Meringue , ice cream , sponge cake or Christmas pudding are main ingredients in baked Alaska .
Alaska	(3, 4)	www.example.org/resource/Alaska001	Baked Alaska originates from France where the national language is French and one of the leader is Gerard Larcher . The dessert is served in Hong Kong as well where the leader is Carrie Lam .
Alaska	(11, 12)	www.example.org/resource/Alaska001	Meringue is an ingredient of a Baked Alaska , which is from the New York region and France .
Alaska	(13, 14)	www.example.org/resource/Alaska001	Sponge cake is one of the ingredients in Baked Alaska , a dish from the New York region and found in the United States .
Alaska	(12, 13)	www.example.org/resource/Alaska001	Christmas pudding is an ingredient in the dessert Baked Alaska . Cookie is also a dessert .
Alaska	(29, 30)	www.example.org/resource/Alaska001	Sandesh (confectionery) is a dish that can be served as a dessert . Another dish that is a dessert is Baked Alaska which has Christmas pudding as an ingredient .
Alaska	(3, 4)	www.example.org/resource/Alaska001	Baked Alaska is thought to have originated in the United States , France or China , and contains Christmas pudding as an ingredient .
Alaska	(3, 4)	www.example.org/resource/Alaska001	Baked Alaska comes from China , where Standard Chinese is spoken .

Fig. 5. Sample output dataframe of Alaska entity.

6. Conclusion and Future Scope

This work proposed a method to extract RDF based Ontology from Unstructured Text using deep learning based Relation Extractor and RDF mapper model both using contextual word embeddings from DistilBERT transformer. We were able to successfully extract the relational triples and assign correct URI even with triple scenarios such as Single Entity Overlapping, Entity Pair Overlapping, same entities in different sentences, homonym entities and was able to achieve high accuracy scores. Our model is equipped to handle Word Sense Disambiguation problem and we evaluated our model on two dataset and was able to build a general knowledge Ontology. Therefore, this approach can be attributed as a recent work which is using contextual embedding to build Ontology when compared with traditional deep learning models and models with static word embeddings.

The Ontology framework used here is RDF which is generic and can be extended to build knowledge graph and for other complex applications. For the future work, we aim to use an Ontology Framework with more expressive power such as OWL etc. Also, the current datasets we used does not have entity names in different context beforehand and we had to add homonym sample for the testing. So the testing dataset is heavily class imbalanced with non-homonym entities. In the future, we aim to address this issue by selecting a more ideal dataset. Another shortcoming with the current approach is that the Relation Extractor and the RDF mapper are two separate units and error from one stage could propagate to another stage. In the future, we aim to resolve by building a joint model. Finally, the transformer variant we used is DistilBERT, which is a BERT [3] based transformer. Even though, we got good results for our work with using word embeddings from the DistilBERT, they are not originally designed for this purpose and further experimentations would be useful. We could also experiment with newer state of the art transformer-based language models for this task.

References

[1] Augenstein, I., Padó, S., Rudolph, S., 2012. Lodifier: Generating linked data from unstructured text, in: Extended Semantic Web Conference, Springer. pp. 210–224.

[2] Cimiano, P., Völker, J., 2005. text2onto, in: International conference on application of natural language to information systems, Springer. pp. 227–238.

[3] Devlin, J., Chang, M.W., Lee, K., Toutanova, K., 2019. BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota. pp. 4171–4186. doi:10.18653/v1/N19-1423.

[4] Gardent, C., Shimorina, A., Narayan, S., Perez-Beltrachini, L., 2017. Creating training corpora for nlg micro-planning, in: 55th annual meeting of the Association for Computational Linguistics (ACL).

[5] Gerber, D., Hellmann, S., Bühmann, L., Soru, T., Usbeck, R., Ngonga Ngomo, A.C., 2013. Real-time rdf extraction from unstructured data streams, in: International semantic web conference, Springer. pp. 135–150.

[6] Hari, A., Kumar, P., 2022. Automated relational triple extraction from unstructured text using transformer, in: International Conference on Electrical and Electronics Engineering, Springer. pp. 472–480.

[7] Jiang, X., Tan, A.H., 2010. Crctol: A semantic-based domain ontology learning system. Journal of the American society for information science and technology 61, 150–168.

[8] Khadir, A.C., Aliane, H., Guessoum, A., 2021. Ontology learning: Grand tour and challenges. Computer Science Review 39, 100339.

[9] Maedche, A., Volz, R., 2001. The ontology extraction & maintenance framework text-to-onto, in: Proc. Workshop on Integrating Data Mining and Knowledge Management, USA, Citeseer. pp. 1–12.

[10] Manola, F., Miller, E., McBride, B., et al., 2004. Rdf primer. W3C recommendation 10, 6.

[11] Muppavarapu, V., Ramesh, G., Gyrard, A., Noura, M., 2021. Knowledge extraction using semantic similarity of concepts from web of things knowledge bases. Data & Knowledge Engineering 135, 101923.

[12] Riedel, S., Yao, L., McCallum, A., 2010. Modeling relations and their mentions without labeled text, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer. pp. 148–163.

[13] Sanh, V., Debut, L., Chaumond, J., Wolf, T., 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108 .

[14] Veena, G., Peter, A.S., Rajkumari, K.A., Ramanan, N., 2016. A concept-based model for query management in service desks, in: Innovations in Computer Science and Engineering. Springer, pp. 255–265.

[15] Yadav, M., Singh, V., et al., 2021. Ontology based data integration and mapping for adverse drug reaction, in: 2021 6th International Conference on Signal Processing, Computing and Control (ISPCC), IEEE. pp. 719–727.