Proceedings of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 185–193
DOI: 10.15439/2025F1850
ISSN 2300-5963 ACSIS, Vol. 43

Assigning scientific texts to existing ontologies

Lukáš Korel
0000-0002-4071-0360
Faculty of Information Technology
Czech Technical University in Prague
Thákurova 2700/9,
160 00 Praha 6, Czech Republic
Email: Lukas.Korel@fit.cvut.cz

Martin Holeňa
0000-0002-2536-9328
Institute of Computer Science,
Czech Academy of Sciences,
Pod Vodárenskou věží 271/2,
182 00 Praha 8, Czech Republic
Email: martin@cs.cas.cz

***Abstract*—Humans try to help computers understand the properties of the real world, and ontologies can be used for this task. Manual enhancement of ontologies is highly time-consuming for domain experts. This paper proposes a solution to match a scientific text to the most relevant ontology using artificial neural networks. Our approach selects a paragraph or a sentence, uses representation learning to embed it into a vector space by some embedder, and measures its relevance to embedded textual properties from the selected ontology by a modified version of a Siamese neural network. We have considered different embedders, their quality has been evaluated on a use case with available ontologies from several application domains.**

***Index Terms*—ontology, neural network, embedding, text matching, LLM**

I. Introduction

Knowledge graphs and ontologies are graph-based structures for knowledge representation. They are defined as "a formal specification of a shared conceptualization" [1], provide a standardized description formalism to express knowledge, facilitating its exchange. Knowledge is typically related to some specific domain. Knowledge graphs and ontologies are commonly used to increase understanding of the related domain.

A domain ontology serves as a framework defining fundamental concepts, such as classes, attributes, and relationships, within a specific knowledge domain. Their definitions encapsulate information about their meaning and constraints. Classes can be defined through annotations or by interconnecting them with properties. Each domain ontology typically employs domain-specific terms to denote its primitives.

The manual construction of classes and their relations is too demanding for the time of domain experts, and automation could bring large amounts of time savings. The automated construction of an ontology consists of several steps:

1) Parse the source texts and extract only relevant content
2) Extract content from the ontologies defining its domain
3) Represent the extracted contents in the same vector space
4) Find the ontology most relevant to the source text
5) Locate the new knowledge in the structure of the ontology
6) Insert the new knowledge, and align it with the ontology internal structure, e.g., via removing duplicities

This paper deals with the issue of automatic ontology design only by assigning articles to ontologies, which corresponds to points 3 and 4. It represents both, the content extracted from ontologies, and the given text in the same vector space and uses an artificial neural network (ANN) to find the ontology most relevant to the given research text.

The next section recalls related work. Section III describes the methodology of text preprocessing and text representation learning, and our architecture of the neural network for the task of text-to-ontology matching. In Section IV, the proposed methodology is validated in a case study by applying it to scientific data.

II. Related Works

In connection with learning and extending ontologies, ANNs have been primarily used for the identification of concepts, relations, and attributes [2], [3], [4]. With respect to relations, some ANN-based methods have been developed specifically for subsumption relations needed for the construction of taxonomies [5], [6], [7], [8]. In connection with the integration of ontologies, they have been used primarily for ontologies matching [9], [10], [11], [12], [13]. The variety of employed ANN types is rather large. It includes traditional multilayer perceptrons [14], adaptive resonance theory networks [15] and associative memories [16], as well as the modern deep convolutional networks [9], [17], deep belief networks [2], long short-term memory (LSTM) networks together with their bidirectional variant (BiLSTM) [18] and gated recurrent units networks [19], [20].

The dependence of ontologies on texts led to using networks developed for text and natural language representation learning, most importantly BERT [21], [22], the bidirectional encoder representations from transformers, and word2vec [23], the most traditional network for embedding text into a Euclidean space. The close relationship of ontologies to knowledge graphs led also to the use of RDF2Vec [7], [20], which was originally proposed for knowledge graphs [24]. Based on similar principles as word2vec and RDF2Vec, OWL2Vec for embedding ontologies [25] and recently OWL2Vec4OA (OWL2Vec for ontology alignment) [26] were proposed. The

This work has been supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS20/208/OHK3/3T/18.

most related paper to our assigning scientific texts to existing ontologies solution is [27]. It presents an attempt to solve a text-to-ontology mapping problem by utilizing NLP tools and neural networks, and assess the final result through visualizing the latent space and exploring the mappings between an input text and ontology classes. Finally, the graph-like structure of ontologies brought the application of graph neural networks [11], [12].

Closest to the proposed approach is the way ANNs have been used in connection with translating into OWL [28], [19], with predicate chaining and restriction [16], and with taxonomy extraction from knowledge graphs [7]. In [28], ontology learning is tailored as a transductive reasoning task that uses two recurrent neural networks to translate text in natural language into OWL specifications in description logic. That approach was further developed in [19], resulting in a system based on a single recurrent network of GRU type. It uses an encoder-decoder configuration and translates a subset of natural language into the description logic language ALLQ through syntactic transformation. Moreover, the system generalizes over different syntactic structures and has the ability to tolerate unknown words through copying input words as extralogical symbols to the output, as well as the ability to enrich the training set with new annotated examples.

In [16], a mapping is established between ontologies and a pair of interacting associative memories. One of them stores assertions, and the other stores entailment rules. The recent work [7] describes a method for the specific task of extracting a taxonomy from an embedding of a knowledge graph. Over that embedding, which can be obtained for example with RDF2Vec, hierarchical agglomerative clustering is performed, first without using type information, and then injecting types into the hierarchical clustering tree.

III. Methodology

This section describes details of the employed methods. In the first part, we address processing textual files, splitting them into paragraphs, and keeping only paragraphs fulfilling minimal length and content relevant to the document's topic. The second part describes the textual embedders considered in assigning texts to ontologies. The final part describes the learning of similarity between an ontology and a given text.

A. Text Preprocessing

The texts used in the case study in Section IV have been extracted from scientific PDFs (Portable Document Files) and the considered ontologies. To read and transform ontologies, we have used the Protégé tool [29], which can understand many commonly used formats for ontology representation. We have chosen the Ontology Web Language (OWL) as the target format and extracted it using the Python package owlready2 [30]. As a representation of each ontology, we use textual content in natural language (e.g., label, comment, description, definition, and note) extracted from their formal representations. Contents shorter than five words were removed because of low context information.

B. Text Representation Learning

For typical downstream data analysis tasks, such as classification or clustering, it is suitable to embed words or other parts of text in a Euclidean space. This representation is mostly the result of representation learning using ANN-based embedders. In our research, we consider the following embedders:

1) Global Vectors for Word Representation (GloVe): [31] is an unsupervised learning algorithm that captures word semantics by leveraging global statistical information. It constructs a word-context co-occurrence matrix and then factorizes it to obtain word embeddings. Each embedding represents a word's position in the semantic space based on its statistical relationships with other words. However, to create embeddings for entire texts, an aggregation process is necessary, which relies on averaging the individual word embeddings based on the TF-IDF (Term Frequency-Inverse Document Frequency) weights.

2) InferSent: [32] is designed for creating universal sentence representations through supervised learning on natural language inference data. It approaches the task by using a carefully crafted training set that includes pairs of sentences with labeled relationships (entailment, contradiction, or neutral). InferSent's embeddings are learned by predicting relationships between sentences. This approach generates sentence embeddings that encapsulate semantic relationships, making it suitable for various downstream natural language processing tasks.

3) Doc2vec-DM: , i.e., the distributed memory version of Doc2Vec [33], is an extension of Word2Vec [34] tailored for document-level embeddings. It operates by learning distributed representations of documents, considering both word and document context. The model can provide either a single embedding per document or embeddings for individual words within the document. This flexibility makes it versatile for various applications, such as document similarity analysis or content-based document retrieval.

4) Bidirectional Encoder Representation from Transformers (BERT): [35] captures contextual information bidirectionally, considering both the left and right context of each word in a sentence. BERT embeddings are contextualized and can be used at both word and sentence levels. Its variant SciBERT is specialized for scientific literature and the variant Sentence-BERT focuses on generating embeddings tailored for entire sentences. Each version of BERT addresses specific use cases, offering state-of-the-art performance across a range of natural language processing tasks.

5) Llama 3.1 + LLM2vec: uses the large language model (LLM) Llama 3.1 [36] published in July 2024, with various model sizes. We have chosen the 8B model, which fits into commonly available GPUs. Llama 3.1 is an auto-regressive language model that uses an optimized transformer architecture. We use it for representation learning using LLM2Vec [37], which enables bidirectional attention, trains the model with masked next-token prediction, and performs unsupervised contrastive learning. The model can be further fine-tuned to achieve state-of-the-art performance in specific cases.

C. Network Learning

Our proposed architecture is inspired by Siamese neural networks [38], which are widely used for tasks involving the estimation of the similarity between two inputs, such as in face verification or semantic textual similarity. The key motivation behind this inspiration lies in the core functionality of Siamese networks: they encode two inputs into a shared embedding space using twin subnetworks with shared weights, and then compute a distance or similarity measure between these embeddings.

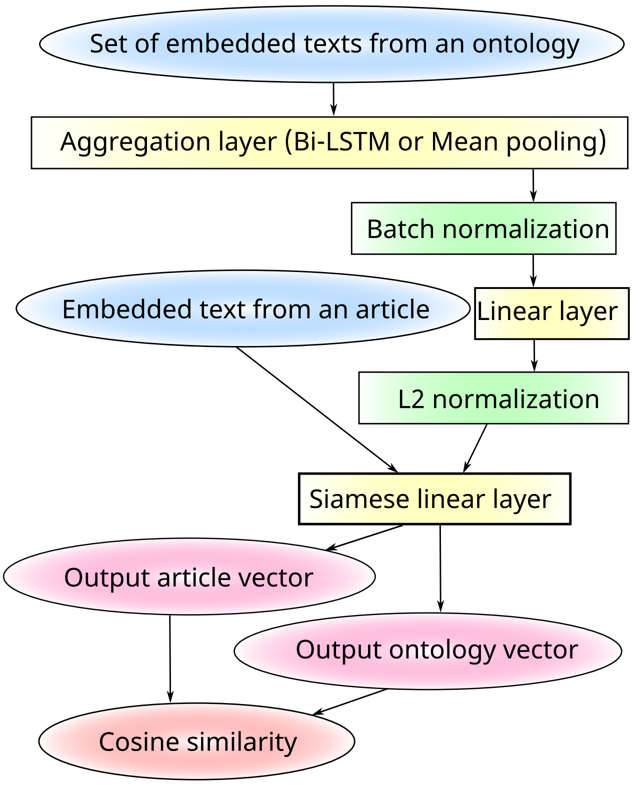

Fig. 1. Neural network schema illustrating used layers (represented with boxes) and flow of the embedded inputs from an ontology and an article to the final cosine distance between them.

In our case, the task requires assessing the semantic distance between a source text and an ontology concept. This parallels the role of Siamese networks, where the goal is to quantify how similar or different two representations are. By adopting this paradigm, our architecture learns to embed both the input text and ontology representation into a common space and returns a scalar distance that reflects their semantic alignment. Our neural network takes an embedded text as an anchor input and a list of embedded, randomly shuffled representative descriptions from an ontology as a representative input. As an output, it returns the distance of the embedded text and representative samples.

Figure 1 shows the internal architecture. The representative input dataset flows through the aggregator, which is represented by the Mean pooling layer or the bidirectional LSTM (long-short-term memory) with a training dropout equal to 0.2 layer in our experiments. The whole batch is then normalized, which is known to improve LSTM results [39]. The following linear layer reduces size and its output is L2 normalized. The following Siamese linear layer is the core of this architecture, which takes the anchor and the output from the second branch and returns for each of them a numeric vector, its size is a quarter compared to the anchor vector. Finally, the cosine similarity between those vectors is computed. More details are provided in Algorithm 1.

IV. Case Study

This section describes the data used in the case study, its experimental setting, and the evaluation of the obtained results.

A. Used Data

The ontologies used in our experiments come from chemistry for both training and testing, all of them being somehow related to catalysis, whereas for testing there are also ontologies from some more distant application domains. Table I shows the details of the considered ontologies.

TABLE I
This table shows the considered ontologies, their counts of textual descriptions of classes and relations, and scientific areas to which they are related.

Area	**Ontology name**	**Count of items with textual definitions**
Chemistry	Allotrope Foundation Ontology (AFO)	2894
Chemistry	Chemical Entities of Biological Interest (CHEBI)	176873
Chemistry	Chemical Methods Ontology (CHMO)	3084
Chemistry	Systems Biology Ontology (SBO)	694
Chemistry	National Cancer Institute Thesaurus (NCIT)	166212
Biology	Biological Collections Ontology (BCO)	671
Energy	Digital Construction Energy (DICES)	67
Environment	Environment Ontology (ENVO)	6605
Finance	Financial Industry Business Ontology (FIBO)	3362

B. Experimental Setting

Each anchor and representative input pair was extracted from the ontologies because we do not have other texts with known ground truth. Each extracted pair was required to fulfill the condition that the anchor text must not appear in the representative descriptions. For our experiment, we selected the considered ontologies and divided them into two parts. From the chemistry group, 80 % of the textual descriptions were randomly selected for training and the rest for testing on similar domains. The remaining ontologies, from more distant domains, were completely used for testing. From the training

Algorithm 1 Training the Neural Network (NN) with Texts from Ontologies and Anchor Texts

Require: Dataset $\mathcal{D} = \{(O_i, a_i, y_i)\}$, where O_i is a list of texts, a_i is the anchor text, and $y_i \in \{0, 1\}$ is the similarity label.
Require: Embedder EM, assigning to each input a vector of size Features.
Require: Aggregator A_ϕ (Bidirectional LSTM or Mean pooling) and Siamese model f_ψ, with trainable parameters ϕ and ψ, an optimizer minimizing a loss function $\mathcal{L}$.
Require: Hyperparameters: number of epochs E and batch size BS.

Initialize parameters ϕ and ψ of the A_ϕ and f_ψ.
Split the dataset $\mathcal{D}$: $\mathcal{D} = \mathcal{D}_{\text{tr}} \cup \mathcal{D}_{\text{val}}$ and $\mathcal{D}_{\text{tr}} \cap \mathcal{D}_{\text{val}} = \emptyset$
Set Siamese network output size: $\text{OutputSize} = \text{Features}/4$.
Initialize $\text{bestLoss} \leftarrow \infty$ and $\text{bestModel} \leftarrow (\phi, \psi)$.
for epoch $= 1$ to E **do**
  Shuffle the dataset $\mathcal{D}_{\text{tr}}$ and divide it into batches of size BS.
  for each batch $\mathcal{B} \subset \mathcal{D}_{\text{tr}}$ **do**
    for each $(O_i, a_i, y_i) \in \mathcal{B}$ **do**
      Aggregate branch:
      for each text $o \in O_i$ **do**
        Compute embedding: $h_o = EM(o)$.
      end for
      Aggregate embeddings using selected aggregator: $h_{O_i} = \mathrm{A}_\phi(\{h_o\})$.
      Apply 1D batch normalization to h_{O_i}, yielding $h_{O_i}^{\text{norm}}$.
      Pass $h_{O_i}^{\text{norm}}$ through a fully connected layer, yielding $h_{O_i}^{\text{fc}}$.
      Apply L2 normalization: $h_{O_i}^{\text{L2}} = \frac{h_{O_i}^{\text{fc}}}{\|h_{O_i}^{\text{fc}}\|_2}$.
      Anchor branch:
      Compute an embedding for the anchor text: $h_{a_i} = EM(a_i)$.
      Siamese network:
      Compute the outputs: $z_i^{\text{agg}} = f_\psi(h_{O_i}^{\text{L2}})$, $z_i^{\text{anchor}} = f_\psi(h_{a_i})$.
      Compute the cosine similarity o_i of $z_i^{\text{agg}}, z_i^{\text{anchor}}$.
      Compute the loss: $\ell = \mathcal{L}(o_i, y_i)$.
    end for
    Update the parameters ϕ and ψ using the optimizer to minimize ℓ for the batch.
  end for
  Compute the validation loss ℓ_{val} for the current model on $\mathcal{D}_{\text{val}}$.
  if $\ell_{\text{val}} < \text{bestLoss}$ **then**
    Update $\text{bestLoss} \leftarrow \ell_{\text{val}}$ and $\text{bestModel} \leftarrow (\phi, \psi)$.
  end if
end for
return bestModel with parts A_ϕ and f_ψ corresponding to the least validation loss.

data, 20 % of textual descriptions were randomly selected as a validation dataset, and the rest as a training dataset. Both datasets were resampled to keep the same proportion of data from each ontology, to mitigate overfitting with texts from one ontology.

For the validation of our model, we have considered the hyperparameters listed in Table II. Marked bold is the combination selected on the validation dataset by the contrastive loss.

$$L_{\text{contrastive}}(x_1, x_2, y) = \frac{1}{2} y \cdot d_{\cos}(x_1, x_2)^2 + \frac{1}{2}(1 - y) \cdot \max(0, 1 - d_{\cos}(x_1, x_2))^2 \quad (1)$$

In the above definition:

- x_1 and x_2 are the outputs of the network for which the contrastive loss is computed, namely the x_1 for the embedded anchor text and the x_2 for the embedded representative list of texts from the selected ontology
- y denotes whether the anchor on the input belongs to the selected ontology ($y = 1$) or does not belong to it ($y = 0$)
- $d_{\cos}(x_1, x_2)$ is the cosine distance between x_1 and x_2

C. Results

All results in this case study were obtained using an independent testing dataset. Writing T for True, F for False, P for positive, N for negative, $TPR = \frac{TP}{TP+FN}$ and $FPR = \frac{FP}{FP+TN}$, the definitions of the employed quality measures are as follows:

$Accuracy = \frac{TP+TN}{TP+FN+TN+FP}$,

TABLE II
CONSIDERED COMBINATIONS OF HYPERPARAMETERS BATCH SIZE (BS), OPTIMIZER (OPT), AND LEARNING RATE (LR), IN BOLD IS THE COMBINATION SELECTED ON THE VALIDATION DATASET BASED ON THE VALUE OF THE CONTRASTIVE LOSS (CL)

BS	OPT	LR	CL
8	Adam	0.1	0.126
8	Adam	0.01	0.126
8	Adam	0.001	0.126
8	SGD	0.1	0.125
8	SGD	0.01	0.126
8	SGD	0.001	0.125
16	Adam	0.1	0.129
16	Adam	0.01	0.081
16	Adam	0.001	0.089
16	SGD	0.1	0.066
16	SGD	0.01	0.063
16	SGD	0.001	0.065
32	Adam	0.1	0.129
32	Adam	0.01	0.124
32	Adam	0.001	0.125
32	SGD	0.1	0.047
32	**SGD**	**0.01**	**0.044**
32	SGD	0.001	0.046

$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$, where $Precision = \frac{TP}{TP+FP}$ and $Recall = TPR$

AUC $= \sum_{i=1}^{n} (\text{FPR}[i] - \text{FPR}[i-1]) \left(\frac{\text{TPR}[i] + \text{TPR}[i-1]}{2} \right)$, where i indexes all the values of the prediction score, which is a numerical output of the model that represents the confidence that a sample belongs to the positive class. The prediction score is sorted and thresholds are applied to generate pairs of TPR and FPR values, spanning the range from the minimum to the maximum prediction score.

1) Statistics from ontologies for near and far domains: The results in Table III for descriptions from ontologies with domains near the training data are best for embeddings obtained with the SciBERT in the model with LSTM aggregation. This may be due to SciBERT having been fine-tuned by scientific data. The worst embedder was Doc2Vec-DM. The results with Doc2Vec and GloVe have been expected because these solutions are based on simple textual embedders. The variant of the model with the mean pooling aggregation layer has even better results. The best results have been achieved with the Llama3.1+LLM2vec embedder without fine-tuning to a specific domain or task. Thanks to the fact that texts are the primary kind of data on which LLMs have been trained, our model had better performance in the considered quality measures.

These results were compared with our previous research [40], where we solved the text-to-ontology matching task. We used the *chemical-bert-uncased* [41] for embedding input texts. These embeddings were used as inputs to the following classifiers: Gaussian process, k-nearest neighbor, multilayer perception, random forest, and support vector machine. Among them, the multilayer perceptron classifier is the most similar to our ANN approach. The approach in [40] has not achieved as high values for the three considered quality measures as have been achieved in the experiments reported in this paper with the proposed ANN architecture, especially in combination with the SciBERT embedder. The best result in [40] were achieved by the Gaussian process classifier, compared to which the SciBERT-based solution reported here has accuracy better by 0.9, AUC by 5.2, and F1 by 8.9 percentile points in the LSTM variant, and in the mean pooling variant, accuracy is better by 1.8, AUC by 6.7, and F1 by 9.8 percentile points. The classification we performed in [40] was restricted to an a priori chosen set of chemical catalytic ontologies, whereas the approach we propose here is independent of the number of potential target ontologies.

TABLE III
RESULTS FOR ONTOLOGIES WITH DOMAINS NEAR TO THE TRAINING DATA COMPARED TO RESULTS FROM [40] BASED ON CLASSIFIERS

ANN LSTM solution	Accuracy	AUC	F1
Doc2Vec-DM	83.3 %	73.9 %	83.2 %
TF-IDF GloVe	84.8 %	76.3 %	84.7 %
InferSent	95.1 %	92.3 %	95.0 %
Llama3.1+LLM2vec	94.3 %	92.0 %	94.3 %
SciBERT	**98.2 %**	**97.1 %**	**98.1 %**
SentenceBERT	95.0 %	92.2 %	95.0 %
ANN Mean pooling solution	Accuracy	AUC	F1
Doc2Vec-DM	92.7 %	88.5 %	92.6 %
TF-IDF GloVe	95.9 %	93.6 %	95.8 %
InferSent	98.9 %	98.3 %	98.8 %
Llama3.1+LLM2vec	**99.3 %**	**98.9 %**	**99.3 %**
SciBERT	99.2 %	98.7 %	99.1 %
SentenceBERT	98.7 %	98.0 %	98.7 %
Classifiers-based solution	Accuracy	AUC	F1
Gaussian process	**97.5 %**	**92.2 %**	**89.5 %**
K-nearest neighbor	96.7 %	90.4 %	87.6 %
Multi-layer perception	97.0 %	91.0 %	87.8 %
Random forest	94.6 %	82.0 %	85.9 %
Support vector machine	97.2 %	91.6 %	88.7 %

TABLE IV
RESULTS FROM ONTOLOGIES WITH DOMAINS FAR FROM THE TRAINING DATA

ANN LSTM solution	Accuracy	AUC	F1
Doc2Vec-DM	67.0 %	55.9 %	66.8 %
TF-IDF GloVe	63.4 %	51.2 %	63.1 %
InferSent	**70.0 %**	**60.0 %**	69.6 %
Llama3.1+LLM2vec	60.2 %	50.9 %	59.3 %
SciBERT	63.5 %	51.3 %	63.2 %
SentenceBERT	69.9 %	59.9 %	**69.9 %**
ANN Mean pooling solution	Accuracy	AUC	F1
Doc2Vec-DM	67.5 %	56.6 %	67.3 %
TF-IDF GloVe	68.9 %	58.5 %	68.0 %
InferSent	69.7 %	59.6 %	**69.4 %**
Llama3.1+LLM2vec	62.7 %	50.3 %	62.2 %
SciBERT	**70.7 %**	**61.0 %**	67.8 %
SentenceBERT	63.7 %	51.6 %	62.9 %

The results in Table IV for texts from ontologies with domains far from the training data are best for embeddings obtained with InferSent and SentenceBERT for the case of LSTM aggregation. In the case of aggregation by mean pooling, the best results were achieved by the model with the SciBERT and InferSent embedders. We did not expect this behavior because LLMs can handle long inputs and understand their semantics pretty well. These results show that embedders that are not based on LLMs are still very useful. This behavior may be caused by the fact that the LLMs are not primarily optimized to create textual embeddings.

2) Statistical significance tests:

The differences between the considered embedders were tested for significance by the Friedman test. The basic null hypotheses that each of the AUC, accuracy, and F1 are for all 6 embedders the same, were rejected for the LSTM aggregation, with the achieved p-values 1×10^{-2}, 3×10^{-2}, and 2×10^{-2}, respectively. In case of mean pooling based aggregation, the achieved p-values were 1×10^{-1}, 3×10^{-2}, and 1×10^{-2}, so we are not able to reject the hypotheses for the AUC metric. As to the post-hoc analysis, because we agree with the opinion of the authors of [42], we followed their proposal to use the Wilcoxon signed rank test with the two-sided alternative for all pairs of compared embedders. For the correction due to multiple hypotheses testing, we used the Holm method. The results are given in Table V and Table VI, the best results by the total score have embeddings from InferSent and both BERTs in the case of LSTM aggregation, and from InferSent and Llama3.1 in the case of mean pooling.

V. Conclusion

In this paper, we proposed an approach to assigning scientific texts to ontologies. This solution contributes to the complex process of automated knowledge processing and highlights the possibilities for research on enhancing existing ontologies. We used neural networks for text embedding and for comparison of obtained embeddings to automatically determine the ontologies most relevant to texts. To this end, we adapted a Siamese neural network in combination with representation learning by various embedders, namely GloVe with TF-IDF weighted aggregation, SentenceBERT, SciBERT, InferSent, Doc2Vec-DM and Llama 3.1 with LLM2vec framework. The embedding of given text serves as the input to one branch of the Siamese neural network, and the list of embeddings of textual descriptions from the considered ontology serves as the input to the second branch. The solutions based on Doc2Vec-DM and GloVE were unsuccessful. In this approach, the most successful combination in ontologies from highly diverse domains was the InferSent embedder, which takes into account contextual relations and produces embeddings with the highest dimension. It achieved good results in both test cases. Our results show that the proposed method surpasses an earlier solution based on a multilayer perceptron classifier. However, challenges remain in the extraction of knowledge from ontologies and in merging new knowledge with existing ontology. In our future research, we want to address these challenges using graph neural networks. Our motivation for such an approach is the graph structure of ontologies.

Acknowledgment

The research reported in this paper has been supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS20/208/OHK3/3T/18.

Computational resources were provided by the e-INFRA CZ project (ID:90254), supported by the Ministry of Education, Youth and Sports of the Czech Republic and by the ELIXIR-CZ project (ID:90255), part of the international ELIXIR infrastructure.

References

[1] W. N. Borst, "Construction of engineering ontologies for knowledge sharing and reuse," Enschede Centre for Telematics and Information Technology (CTIT), Tech. Rep., 1997.

[2] F. Al-Aswadi, H. Chan, and K. Gan, "Extracting semantic concepts and relations from scientific publications by using deep learning," in *Proceedings of IRICT 2020*, 2021, pp. 374–383.

[3] N. Gupta, S. Podder, K. Annervaz, and S. Sengupta, "Domain ontology induction using word embeddings," in *ICMLA*, 2016, pp. 115–119.

[4] A. Katyshev, A. Anikin, M. Denisov, and T. Petrova, "Intelligent approaches for the automated domain ontology extraction," in *International Congress on Information and Communication Technology*, 2021, pp. 410–417.

[5] S. Althubaiti, S. Kafkas, M. Abdelhakim, and R. Hoehndorf, "Combining lexical and context features for automatic ontology extension," *Journal of Biomedical Semantics*, vol. 11, p. article no. 1, 2020.

[6] L. Espinoza-Anke, F. Ronzano, and H. Saggion, "Hypernym extraction: Combining machine-learning and dependency grammar," in *CICLing*, 2015, pp. 372–383.

[7] F. Martel and A. Zouaq, "Taxonomy extraction using knowledge graph embeddings and hierarchical clustering," in *SAC'21*, 2021, pp. 836–844.

[8] N.-A. R., J. R., and J. Castro, "Automated ontology extraction from unstructured texts using deep learning," in *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*. Springer, 2020, pp. 727–755.

[9] A. Bento, A. Zouaq, and M. Gagnon, "Ontology matching using convolutional neural networks," in *LREC*, 2020, pp. 5648–5653.

[10] J. Chakraborty, B. Yaman, L. Virgili, K. Konar, and S. Bansal, "Onto-Connect: Results for OAEI 2020," in *OM ISWC*, 2020, pp. 204–210.

[11] L. Hao, C. Lei, V. Efthymiou, A. Quamar, F. Özcan *et al.*, "MEDTO: Medical data to ontology matching using hybrid graph neural networks," in *KDD'21*, 2021, pp. 2946–2954.

[12] J. Wu, J. Lv, H. Guo, and S. Ma, "Daeom: A deep attentional embedding approach for biomedical ontology matching," *Applied Sciences*, vol. 10, p. article no. 7909, 2020.

[13] A. Benarab, G. Yin, and F. Rafique, "An embedding-based approach to the ontology matching problem," in *2023 IEEE 4th International Conference on Pattern Recognition and Machine Learning (PRML)*, 2023. doi: 10.1109/PRML59573.2023.10348247 pp. 642–649.

[14] S. Kolozali, G. Fazekas, M. Barthet, and M. Sandler, "A framework for automatic ontology generation based on semantic audio analysis," in *Audio Engineering Society International Conference*, 2014, pp. 87–96.

[15] M. Hourali and G. Montazer, "Using ART2 neural network and bayesian network for automating the ontology constructing process," *Procedia Engineering*, vol. 29, pp. 3914–3923, 2012.

[16] C. Mercier, H. Chateau-Laurent, F. Alexandre, and T. ViÃ©ville, "Ontology as neuronal-space manifold: Towards symbolic and numerical artificial embedding," in *Workshop on Knowledge Representation for Hybrid and Compositional AI*, 2021, pp. 1–11.

[17] G. Li, "CNN based ontology learning algorithm and applied in PE data," *IAENG International Journal of Computer Science*, vol. 48, pp. 1–8, 2021.

[18] R. Mueller and S. Abdullaev, "Deep cause: Hypothesis extraction from information systems papers with deep learning for theory ontology learning," in *Annual Hawaii International Conference on System Sciences*, 2019, pp. 6250–6259.

TABLE V

COMPARISON OF AUC, ACCURACY, AND F1 RESULTS ON ALL 9 CONSIDERED ONTOLOGY DATASETS FOR THE EMBEDDERS COMBINED WITH LSTM AGGREGATION. THE VALUES IN THE TABLE ARE COUNTS OF DATASETS, IN WHICH THE EMBEDDER IN THE ROW HAS A HIGHER VALUE OF THE CONSIDERED PERFORMANCE MEASURE COMPARED TO THE EMBEDDER IN THE COLUMN. IF THE DIFFERENCE IS SIGNIFICANT ON THE 5 % LEVEL THEN THE COUNT IS IN BOLD. THE COLUMN "TOTAL" CONTAINS THE SUM OF THE COUNTS IN THE ROW. THE LAST COLUMN CONTAINS THE MEANS ± STANDARD DEVIATIONS OVER ALL TESTING DATASETS OF THE RESULTS FOR THE CONSIDERED PERFORMANCE MEASURE.

AUC	GloVe	Sentence_BERT	Llama3.1	InferSent	Doc2Vec-DM	SciBERT	Total	$Mean \pm Std$
GloVe		1	3	**1**	4	**1**	10	0.605 ± 0.105
Sentence_BERT	8		7	6	**8**	6	35	0.723 ± 0.167
Llama3.1	6	2		2	5	3	18	0.684 ± 0.185
InferSent	**8**	3	7		**8**	4	30	0.716 ± 0.173
Doc2Vec-DM	5	**1**	4	**1**		4	15	0.614 ± 0.140
SciBERT	**8**	3	6	5	5		27	0.729 ± 0.204
Accuracy	GloVe	Sentence_BERT	Llama3.1	InferSent	Doc2Vec-DM	SciBERT	Total	$Mean \pm Std$
GloVe		2	4	**1**	6	**2**	15	0.844 ± 0.016
Sentence_BERT	7		6	5	7	2	27	0.891 ± 0.061
Llama3.1	5	3		4	6	2	20	0.875 ± 0.065
InferSent	**8**	4	5		**9**	5	31	0.888 ± 0.049
Doc2Vec-DM	3	2	3	**0**		**2**	10	0.848 ± 0.056
SciBERT	**7**	7	7	4	**7**		32	0.893 ± 0.056
F1	GloVe	Sentence_BERT	Llama3.1	InferSent	Doc2Vec-DM	SciBERT	Total	$Mean \pm Std$
GloVe		**2**	4	**1**	6	**1**	14	0.840 ± 0.023
Sentence_BERT	**7**		7	6	7	3	30	0.890 ± 0.062
Llama3.1	5	2		4	5	2	18	0.871 ± 0.066
InferSent	**8**	3	5		**9**	5	30	0.884 ± 0.054
Doc2Vec-DM	3	2	4	**0**		2	11	0.847 ± 0.053
SciBERT	**8**	6	7	4	7		32	0.886 ± 0.063

TABLE VI

Comparison of AUC, Accuracy, and F1 results on all 9 considered ontology datasets for the embedders combined with mean pooling. The values in the table are counts of datasets, in which the embedder in the row has a higher value of the considered performance measure compared to the embedder in the column. If the difference is significant on the 5 % level then the count is in bold. The column "Total" contains the sum of the counts in the row. The last column contains the means ± standard deviations over all testing datasets of the results for the considered performance measure.

AUC	GloVe	Sentence_BERT	Llama3.1	InferSent	Doc2Vec-DM	SciBERT	Total	$Mean \pm Std$
GloVe		3	2	1	6	2	14	0.756 ± 0.172
Sentence_BERT	6		2	3	5	5	21	0.757 ± 0.217
Llama3.1	7	7		7	5	8	34	0.783 ± 0.221
InferSent	8	6	2		**7**	5	28	0.789 ± 0.190
Doc2Vec-DM	3	4	4	**2**		3	16	0.699 ± 0.158
SciBERT	7	4	1	4	6		22	0.764 ± 0.217
Accuracy	GloVe	Sentence_BERT	Llama3.1	InferSent	Doc2Vec-DM	SciBERT	Total	$Mean \pm Std$
GloVe		4	3	2	**8**	2	19	0.903 ± 0.041
Sentence_BERT	5		4	3	**7**	3	22	0.904 ± 0.062
Llama3.1	6	5		6	**8**	6	31	0.914 ± 0.070
InferSent	7	6	3		**8**	7	31	0.917 ± 0.053
Doc2Vec-DM	**1**	**2**	**1**	**1**		2	7	0.881 ± 0.058
SciBERT	7	6	3	2	7		25	0.907 ± 0.062
F1	GloVe	Sentence_BERT	Llama3.1	InferSent	Doc2Vec-DM	SciBERT	Total	$Mean \pm Std$
GloVe		5	4	2	8	3	22	0.898 ± 0.051
Sentence_BERT	4		3	2	6	3	18	0.898 ± 0.069
Llama3.1	5	6		5	**8**	7	31	0.910 ± 0.074
InferSent	7	7	4		**9**	7	34	0.912 ± 0.059
Doc2Vec-DM	1	3	**1**	**0**		**2**	7	0.880 ± 0.059
SciBERT	6	6	2	2	**7**		23	0.899 ± 0.067

[19] G. Petrucci, M. Rospocher, and C. Ghindini, "Expressive ontology learning as neural machine translation," *Journal of Web Semantics*, vol. 52–53, pp. 66–82, 2018.

[20] J. Potoniec, "Learning OWL 2 property characteristics as an explanation for an RNN," *Bulletin of the Polish Academy of Sciences, Technical Sciences*, vol. 68, pp. 1481–1490, 2020.

[21] A. Memariani, M. Glauer, F. Neuhaus, T. Mossakowski, and J. Hatings, "Automated and explainable ontology extension based on deep learning: A case study in the chemical domain," in *3rd International Workshop on Data Meets Applied Ontologies*, 2021, pp. 1–16.

[22] A. Oba, I. Paik, and A. Kuwana, "Automatic classification for ontology generation by pretrained language model," in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, 2021, pp. 210–221.

[23] N. Teslya and S. Savosin, "Matching ontologies with Word2Vec-based neural network," in *ICCSA*, 2019, pp. 745–756.

[24] P. Ristoski and H. Paulheim, "Rdf2vec: Rdf graph embeddings for data mining," in *International Semantic Web Conference*, 2016, pp. 498–514.

[25] A. Ritchie, J. Chen, L. Castro, D. Rebholz-Schuhmann, and E. JimÃ©nez-Ruiz, "Ontology clustering with OWL2Vec," in *DeepOntoNLP*, 2021, pp. 54–61.

[26] S. Teymurova, E. Jiménez-Ruiz, T. Weyde, and J. Chen, "Owl2vec4oa: Tailoring knowledge graph embeddings for ontology alignment," in *Sixth International Knowledge Graph and Semantic Web Conference*. Springer International Publishing, October 2024. ISSN 0302-9743. [Online]. Available: https://openaccess.city.ac.uk/id/eprint/34052/

[27] U. Yorsh, A. S. Behr, N. Kockmann, and M. Holeňa, "Text-to-ontology mapping via natural language processing models," in *Conference on Theory and Practice of Information Technologies*, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:253161694

[28] G. Petrucci, C. Ghindini, and M. Rospocher, "Ontology learning in the deep," in *EKAW*, 2016, pp. 480–495.

[29] M. A. Musen, "The protégé project: a look back and a look forward," *AI Matters*, vol. 1, no. 4, pp. 4–12, 2015. doi: 10.1145/2757001.2757003. [Online]. Available: https://doi.org/10.1145/2757001.2757003

[30] J.-B. Lamy, "Owlready: Ontology-oriented programming in python with automatic classification and high level constructs for biomedical ontologies," *Artificial Intelligence In Medicine*, vol. 80, pp. 11–28, 2017.

[31] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, A. Moschitti, B. Pang, and W. Daelemans, Eds. Doha, Qatar: Association for Computational Linguistics, Oct. 2014. doi: 10.3115/v1/D14-1162 pp. 1532–1543. [Online]. Available: https://aclanthology.org/D14-1162

[32] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, M. Palmer, R. Hwa, and S. Riedel, Eds. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017. doi: 10.18653/v1/D17-1070 pp. 670–680. [Online]. Available: https://aclanthology.org/D17-1070

[33] A. M. Dai, C. Olah, and Q. V. Le, "Document embedding with paragraph vectors," *ArXiv*, vol. abs/1507.07998, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:7364264

[34] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds., vol. 26. Curran Associates, Inc., 2013. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2013/file\/9aa42b31882ec039965f3c4923ce901b-Paper.pdf

[35] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "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)*, J. Burstein, C. Doran, and T. Solorio, Eds. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019. doi: 10.18653/v1/N19-1423 pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[36] Meta, "Llama-models/models/llama3_1 at meta-llama/llama-models," Jul 2024. [Online]. Available: https://github.com/meta-llama/llama-models/tree/main/models/llama3_1

[37] P. BehnamGhader, V. Adlakha, M. Mosbach, D. Bahdanau, N. Chapados, and S. Reddy, "LLM2Vec: Large language models are secretly powerful text encoders," in *First Conference on Language Modeling*, 2024. [Online]. Available: https://openreview.net/forum?id=IW1PR7vEBf

[38] J. Bromley, J. Bentz, L. Bottou, I. Guyon, Y. Lecun, C. Moore, E. Sackinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 7, p. 25, 08 1993. doi: 10.1142/S0218001493000339

[39] J. Whang and A. Matsukawa, "Exploring batch normalization in recurrent neural networks," *Stanford Center for Professional Development*, 2019, accessed on 29 February 2024. [Online]. Available: https://jaywhang.com/assets/batchnorm_rnn.pdf

[40] L. Korel, A. S. Behr, M. Holeňa, and N. Kockmann, "Using artificial neural networks to determine ontologies most relevant to scientific texts," in *Proceedings of the 22nd Conference Information Technologies – Applications and Theory (ITAT 2022)*, ser. CEUR Workshop Proceedings, vol. 3226. CEUR-WS.org, 2022. ISSN 1613-0073 pp. 44–54. [Online]. Available: https://ceur-ws.org/Vol-3226/paper5.pdf

[41] R. A. Company, "Bert for chemical industry," https://huggingface.co/recobo/chemical-bert-uncased, 2022.

[42] A. Benavoli, G. Corani, and F. Mangili, "Should we really use post-hoc tests based on mean-ranks?" *Journal of Machine Learning Research*, vol. 17, pp. 1–10, 2016.