Using Paraphrasers to Detect Duplicities in Ontologies

Lukáš Korel[1][a], Alexander S. Behr[2][b], Norbert Kockmann[2][c] and Martin Holeňa[1,3][d]

[1] *Faculty of Information Technology, Czech Technical University, Prague, Czech Republic*
[2] *Faculty of Biochemical and Chemical Engineering, TU Dortmund University, Dortmund, Germany*
[3] *Institute of Computer Science, Czech Academy of Sciences, Prague, Czech Republic*
fi

Keywords: Ontologies, Semantic Similarity, Duplicity Detection, Representation Learning, Paraphrasers, Classifiers.

Abstract: This paper contains a machine-learning-based approach to detect duplicities in ontologies. Ontologies are formal specifications of shared conceptualizations of application domains. Merging and enhancing ontologies may cause the introduction of duplicities into them. The approach to duplicities proposed in this work presents a solution that does not need manual corrections by domain experts. Source texts consist of short textual descriptions from considered ontologies, which have been extracted and automatically paraphrased to receive pairs of sentences with the same or a very close meaning. The sentences in the received dataset have been embedded into Euclidean vector space. The classification task was to determine whether a given pair of sentence embeddings is semantically equivalent or different. The results have been tested using test sets generated by paraphrases as well as on a small real-world ontology. We also compared solutions by the most similar existing approach, based on GloVe and WordNet, with solutions by our approach. According to all considered metrics, our approach yielded better results than the compared approach. From the results of both experiments, the most suitable for the detection of duplicities in ontologies is the combination of BERT with support vector machines. Finally, we performed an ablation study to validate whether all paraphrasers used to create the training set for the classification were essential.

1 INTRODUCTION

Increasing use of domain ontologies has led to attempts to construct, extend, or integrate them automatically or semi-automatically, to alleviate the need for the manual effort of domain experts. During the last decade, artificial neural networks (ANNs) have been often used to this end, though nearly always in simple data-driven methods based on empirical mappings. Only recently, several applications of ANNs to ontologies included knowledge modeling, making use, for example, of neural machine translation or embeddings obtained with representation learning methods.

This paper is devoted to a specific problem encountered during enhancing ontologies and sometimes during their merging: to decide whether a particular concept is already contained in the existing ontology. Our solution to the problem relies primarily on transformers, a kind of ANNs developed primarily for the transformation of natural language texts. The next section describes this problem in more detail. Section 3 outlines our proposed methodology. Related works are briefly described in Section 4. Section 5 deals with experimental validation and is divided into four parts. The first subsection describes our experimental setup. The second one compares all considered variants of our approach with respect to four quality measures. In the third subsection, they are compared on a dataset created from relevant real-world ontologies with a similar existing approach. Finally, the last one is an ablation study for the employed set of paraphrasers.

2 PROBLEM DESCRIPTION

Automated ontology construction and ontology mapping is a complex process, which consists of many steps. An important step is the merging of the semantic content expressed in an ontology by RDF triples. A triple consists of three components: a subject, a

[a] https://orcid.org/0000-0002-4071-0360
[b] https://orcid.org/0000-0003-4620-8248
[c] https://orcid.org/0000-0002-8852-3812
[d] https://orcid.org/0000-0002-2536-9328

Korel, L., Behr, A., Kockmann, N. and Holeňa, M.
Using Paraphrasers to Detect Duplicities in Ontologies.
DOI: 10.5220/0012164500003598
In *Proceedings of the 15th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2023) - Volume 2: KEOD*, pages 40-49
ISBN: 978-989-758-671-2; ISSN: 2184-3228

predicate, and an object. These triples may be automatically extracted from scientific texts and enhanced by descriptions of content. However, the set of extracted triples may contain many semantic duplicities, thus merging them into an ontology causes duplicities in the resulting ontology.

Detecting semantic duplicity manually may be prohibitive for domain experts because the ontology may contain too many nodes, relations and descriptions to detect all semantic duplicities manually. Our objective is, therefore, to detect duplicities in ontologies automatically.

3 PROPOSED METHODOLOGY

The methodology we suggest does not need a dataset in which semantic duplicities are marked by domain experts. It only needs an existing ontology that does not contain semantic duplicities, and has a sufficient number of nodes, typically more than a few thousand. Our methodology extracts content from such an existing ontology, in particular names and descriptions of the nodes and relations among them.

From the extracted names and descriptions of the nodes and relations in a given ontology, a dataset is prepared that contains for each description a few descriptions with the same meaning. To avoid the necessity to mark such pairs of strings manually by domain experts, our methodology makes use of paraphrasers, which are able to create a different sentence with the same or a very close meaning.

The results from the paraphrasers have been used to create a 3-column dataset containing: text A, text B, and a similarity mark. If text A and text B in one row are 2 different paraphrases of the same original text, their similarity is marked as true. Consequently, each such row contains semantically equivalent texts. The same number of pairs have been randomly selected from different node descriptions or paraphrases, so the similarity of those other pairs is marked as false. It means the texts in each such row are semantically different. The dataset is balanced because we have the same amount of pairs with the same and with different meanings.

BERT (Bidirectional Encoder Representations from Transformers), which is a kind of artificial neural network known as a transformer, is widely used for representation learning. Its ability, in comparison to more traditional text representation learning methods, is to embed whole parts of the text, thus to include also the context of each word. This makes it possible to achieve top results in text classification (Devlin et al., 2019). It has impressive transfer learning capabilities (Lu et al., 2021), which can be useful for fine-tuning the model for tasks that fall outside the data with which it was trained originally. Due to BERT's complexity, a pre-trained version is usually used, which can be fine-tuned using texts relevant to the topic of interest. For classification, we have decided to use embeddings of the whole class description. The paraphrases of descriptions from the ontology were embedded into a Euclidean space by the transformer as well.

Finally, the task of the classifiers is to decide, which textual pair has the same meaning and which has a different meaning. The embeddings of the above described pairs of texts serve as training data for training the classifiers, thus the trained classifiers are able to recognize if a given pair of embedded texts have the same or different meanings.

4 RELATED WORKS

The closest approach we are aware of is the model UTtoKB (Salim and Mustafa, 2021). Similarly to our approach, it relies on representation learning and searches for coreferences in connection with ontologies. However, representation learning is performed with WordNet and GLoVe, i.e. with simpler and more traditional methods than BERT, and the search is performed not directly in an ontology, but in texts interpreted by means of it. The ontology is combined with representation learning, semantic role labeling, and the resource description framework, to find semantic similarities in the texts.

In our opinion, UTtoKB is the only approach that is so close to ours that it makes sense to experimentally compare them. Still, we recall also several others that are somehow related. All of them are similar to UTtoKB in dealing with coreferences in texts, and not with coreferences in an ontology as our approach does. The only one that also uses representation learning, actually also BERT, is (Trieu et al., 2019). Different to our approach and to UTtoKB, however, it focuses on syntactic aspects of mentions. What is in their approach learned, is a syntactic parsing model.

The other approaches are not representation-learning-based. The system presented in (Chen et al., 2011) performs coreference resolution in two steps. At first, all mentions in the text are detected by means of classification using a maximum entropy classifier, or alternatively a classification tree or a support vector machine. Then, those mentions are clustered into coreference chains. In (Lee et al., 2018), span ranking is combined with searching maximum-likelihood span pairs. Their approach is based on coarse-to-fine inference: in each iteration, it uses the antecedent distribution to infer later coreference decisions using earlier coreference decisions. Similarly to UTtoKB, ontologies are in search for coreferences used to interpret texts also in (Garanina et al., 2018). That is a multiagent approach, in which for each ontology class, there is a specific agent performing a rule-based check whether a given information object is consistent with that class. Finally, the Tree Coreference Resolver (Novák, 2017) operates on the tectogrammatical layers, which allows a deeper syntax representation of the text than all previously mentioned approaches. However, this representation is advantageous primarily for pronoun and zero coreferences, whereas duplicities in ontologies rely on nominal groups.

5 EXPERIMENTAL VALIDATION

The ontologies used in our experiments come from a chemical domain, namely from catalysis. The considered ontologies for paraphrasing their textual content are listed in Table 1. The Allotrope Foundation Ontology (AFO) has rich textual descriptions of the classes and relations. The BioAssay Ontology (BAO) is focused on biological screening assays and their results. Certain concepts in the BAO concern the chemical roles of substances (e.g. catalysts). The Chemical Entities of Biological Interest (CHEBI), and Chemical Methods Ontology (CHMO) are closely related to the chemical domain and contain concepts related to chemical experiments in laboratories. In contrast, the Systems Biology Ontology (SBO) concerns system biology and computational modeling. We have taken it into consideration as it includes also relations regarding substances and general laboratory contexts, which are contained in texts from catalysis. The IUPAC Compendium of Chemical Terminology (IUPAC) and the National Cancer Institute Thesaurus (NCIT) cover vast amounts of chemical species and domain-specific chemical knowledge. Contrary to the other ontologies investigated, the NCIT does not contain relationships between classes as it is constructed to serve as a thesaurus rather than as an ontology. In order to be processed properly, all ontologies were used in the OWL file format. Based on the above-outlined content of the considered ontologies, we decided to use the AFO and the SBO for experimental validation.

Table 1: The initial pool of ontologies from which the considered ontology SBO and AFO were selected. This table also shows the count of textual descriptions of nodes and relations in each ontology.

Ontology name	**Count of items** **with textual definitions**
AFO	2894
BAO	7514
CHEBI	176873
CHMO	3084
SBO	694
IUPAC	7038
NCIT	166212

5.1 Experimental Setup

We have extracted content from two ontologies, from the AFO for training and from the SBO for independent testing, using the Owlready2 Python package. We have chosen those two ontologies, due to their rich text descriptions and size. For each description of node and relation, taken from them, we have prepared different texts with the same or very close meaning.

We divide the employed paraphrasers into four groups by the original transformer and tuning data source, by which final paraphrasers have been created: Bart, Pegasus, Paws, and T5 paraphraser. Altogether, we employed the following paraphrasers:

- Eugenesiow/Bart-paraphrase **(Bart)** (available from (Huggingface, 2019a), based on (Lewis et al., 2019))
- Tuner007/Pegasus-paraphrase **(Pegasus)** (available from (Huggingface, 2019b), based on (Zhang et al., 2019a))
- Vamsi/T5-paraphrase-paws **(Paws)** (available from (Vamsi, 2019), based on (Yang et al., 2019; Zhang et al., 2019b))
- PrithivirajDamodaran/Parrot-paraphraser **(Paws)** (Damodaran, 2021)
- Humarin/Chatgpt-paraphraser-on-T5-base **(T5 paraphraser)** (Vorobev and Kuznetsov, 2023)
- Ramsrigouthamg/T5-large-paraphraser-diverse-high-quality **(T5 paraphraser)** (Ramsrigouthamg, 2022a)
- Ramsrigouthamg/T5-paraphraser **(T5 paraphraser)** (Ramsrigouthamg, 2022b)
- Valurank/T5-paraphraser **(T5 paraphraser)** (Valurank, 2022)

The following example illustrates the possibility to paraphrase a description from chemical domain by paraphrasers:

Source Text: If sodium metal and chlorine gas mix under the right conditions, they will form salt. The sodium loses an electron, and the chlorine gains that electron. This reaction is highly favorable because of the electrostatic attraction between the particles. In the process, a great amount of light and heat is released.

Paraphrased Text: If sodium metal and chlorine gas are mixed in appropriate conditions, they will create salt. Sodium surrenders an electron, and chlorine gains this particular electron. This reaction occurs favorably due to the electrostatic pull between the particles. Throughout this process, a substantial amount of light and heat is emitted.

All textual outputs from the paraphrasers have been embedded using the state-of-the-art sentence transformer named all-MiniLM-L6-v2 (Reimers and Gurevych, 2019). It is able to make embedding of the whole description or its paraphrase. Behind this transformer is SBERT (Thakur et al., 2021), which is the modification of the BERT using siamese and triplet networks that is able to derive semantically meaningful sentence embeddings.

In our approach, we have used the following classifiers from the Scikit-Learn library for the classification of pairs of embeddings: random forest, gradient boosting, Gaussian process, multi-layer perceptron, support vector machine, their team with hard voting, and their team with soft voting. Hard voting sums predictions for each class, and the team decides based on the highest count of votes. Soft voting takes probability distribution over the classes from each classifier in the team, then sums them per class and makes the decision based on the highest value of the sum of prediction probabilities.

For tuning the hyperparameters of those classifiers, we have considered hyperparameter values shown in Table 2. The optimal values, marked bold in Table 2, have been selected by grid-search using 3-fold cross-validation on 15 % descriptions randomly selected of the paraphrased sentences obtained from the AFO.

For testing, we have selected the descriptions of nodes and relations of the SBO ontology. These texts we have paraphrased by the same paraphrasers described above. The received texts have been embedded by the same sentence transformer all-MiniLM-L6-v2 (Reimers and Gurevych, 2019). A balanced testing dataset has been created from these embeddings by random sampling. The whole dataset from the AFO was randomly divided into 5 datasets as input to the 5-fold cross-validation. For each of the above-listed classifiers with the most suitable combination of parameters, a 5-fold cross-validation was performed and the model with the best precision score on validation data was selected, in order to mitigate overfitting on training data.

5.2 Statistical Comparison of Employed Classifiers

To compare results obtained with different classifiers on validation data, we have randomly split the dataset of the generated paraphrases from the SBO's descriptions into 23 datasets in such a way that their content could be considered approximately independent. This was the lowest number of datasets with such an approximate independence property. We have compared the employed classifiers with respect to four quality measures, namely accuracy, precision, recall, and F1-measure. The resulting distributions of those quality measures are depicted as box plots in Figure 1. The worst result has been achieved by the multi-layer perceptron classifier. Very low standard deviations have been achieved by the gradient-boosting classifier and by both used teams.

Firstly, we have performed a Friedman's test to check, for the measures, the hypothesis that the assessment of all considered classifiers by the respective quality measure is the same. This hypothesis has been rejected for all considered quality measures, the achieved significance levels, a.k.a *p*-values, were the following: for accuracy 3.43×10^{-12}, for precision 2.75×10^{-11}, for recall 3.43×10^{-12}, and for F1-measure 4.40×10^{-12}.

After the hypotheses of the same assessment of all classifiers were for all quality measures rejected, we performed Wilcoxon signed rank test to compare them with the classifier that achieved the best result with respect to the considered quality measure. Table 3 shows the results of all classifiers for all quality measures based on the 23 considered datasets. According to these results, the team with hard voting achieved the best results among all the considered classifiers and teams combining them. The multi-layer perceptron has achieved the worst results. This may be caused by its sensitivity to a domain different from the domain corresponding to its training data because the domain of the ontology SBO differs a lot from the domain of the ontology AFO.

Table 2: Considered hyperparameters of the considered classifiers (hyperparameters selected by grid search cross-validation are marked bold).

Classifier	**Hyperparameter**	**Considered values**
Random forest	max depth	3, 5, 7, 9, **11**
	min samples split	**5**
	criterion	entropy, **gini**
	n estimators	50, 100, 150, **200**, 250
	max features	**sqrt**, log2
	bootstrap	**False**, True
Gradient boosting	max depth	3, 5, 7, 9, **11**
	learning rate	0.05, **0.1**, 0.2
	criterion	friedman mse, **squared error**
	n_estimators	50, 150, **250**
	max_features	**sqrt**, log2
Support vector machine	C	0.001, **0.01**, 0.1, 1, 10, 100
	gamma	0.001, 0.01, 0.1, 1, 10, **100**
	kernel	**rbf**, sigmoid, poly
Gaussian process	kernel	1*RBF(0.1), 1*RBF(1.0), **1*RBF(10)**, 1*DotProduct(0.1), 1*DotProduct(10), 1*Matern(), 1*RationalQuadratic(), 1*WhiteKernel()
Multi-layer perceptron	random state	0, **1**
	max iter	**500**, 1000
	activation function	identity, logistic, tanh, **relu**
	solver	**lbfgs**, sgd, adam
	hidden layer sizes	16, 32, 64, 128, **256**, 512
	alpha	0.00001, 0.001, **0.1**, 10.0, 1000.0
	learning rate	**constant**, invscaling, adaptive

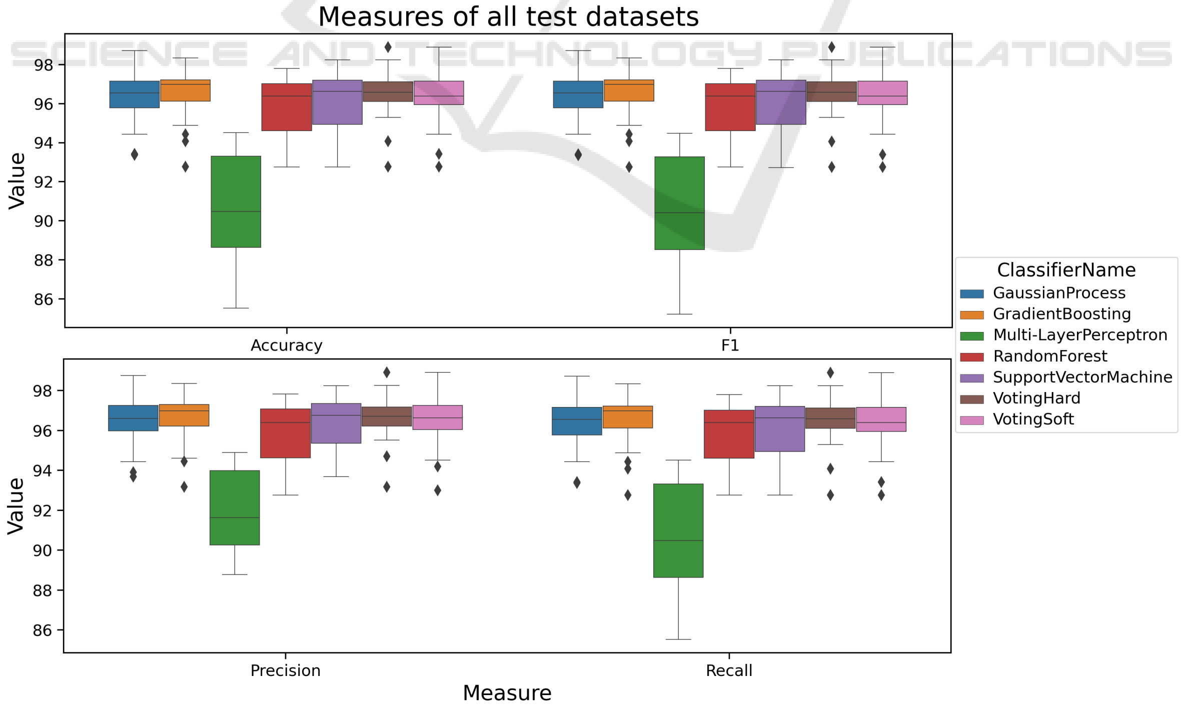

Figure 1: The quality measures with a standard deviation of the considered classifiers on independent datasets extracted from SBO. All considered classifiers were compared with respect to the accuracy, precision, recall, and F1 score.

Table 3: Comparison of accuracy, precision, recall, and F1 score results on the 23 independent sets extracted from SBO. The values in the table are mean values ± standard deviation. The green cells (values in bold) mark the highest value in a particular metric. The white cells (values in italics) mark classifiers for which the difference to the classifier with the highest score is not significant according to the Wilcoxon signed rank test after Holm correction for multiple hypotheses testing and the red cells mark classifiers for which that difference is significant.

Values are in %	Gaussian process	Gradient boosting	Multi-layer perceptron	Random forest	Support vector machine	Team with hard voting	Team with soft voting
Accuracy	*96.4* *±1.42*	*96.5* *±1.36*	90.6 ±2.64	95.6 ±1.56	*96.0* *±1.63*	**96.5** **±1.29**	*96.4* *±1.49*
Precision	*96.6* *±1.34*	*96.6* *±1.28*	92.0 ±1.96	95.7 ±1.53	*96.2* *±1.39*	**96.7** **±1.18**	*96.5* *±1.39*
Recall	*96.4* *±1.42*	*96.5* *±1.36*	90.6 ±2.64	95.6 ±1.56	*96.0* *±1.63*	**96.5** **±1.29**	*96.4* *±1.49*
F1 measure	*96.4* *±1.43*	*96.5* *±1.36*	90.5 ±2.71	95.6 ±1.56	*96.0* *±1.64*	**96.5** **±1.29**	*96.4* *±1.49*

5.3 Comparison with the Most Similar Existing Approach

To compare our method with an existing approach, we have manually prepared a dataset based on annotations from BAO, CHEBI, CHMO, NCIT, SBO, and IUPAC ontologies. The dataset considers descriptions of equivalent classes that appear at least in two considered ontologies. This dataset contains approximately 400 unique descriptions of classes and for each of them on average 4 equivalent descriptions with the same meaning from all considered ontologies. From all those sentences and phrases, we have randomly combined 3200 pairs of descriptions, thus for each pair, it is known whether both sentences or phrases in the pair are equivalent or not.

Results from our experiment are presented in Table 4. We have compared our approach in variants with all considered classifiers, including classifier teams, to the only approach indicated in Section 4 as sufficiently similar, i.e. (Salim and Mustafa, 2021), which is based on GloVe and WordNet. The best results have been achieved by the multi-layer perceptron (MLP) and by the support vector machine. The MLP achieves substantially better results in this experiment than in the experiment in Subsection 5.2. In our opinion, this is due to the fact the domains of most of the ontologies employed in this experiment are much closer to the domain corresponding to training data than the domain of the ontology SBO employed in Subsection 5.2.

5.4 Ablation Study of the Employed Paraphrasers

To assess the importance of using particular paraphrasers and groups of paraphrasers to generate the training data for the classifiers sets of paraphrases, we have used the same testing dataset as in the experiment described in Subsection 5.2. We have performed the ablation study of the employed paraphrasers separately for each of the considered quality measures accuracy, precision, recall, and F1-measure. Each experiment uses paraphrases generated by all paraphrasers from the list in Subsection 5.1, except a particular one or a particular group.

Results from our experiment are presented in Tables 5 and 6. The results with all paraphrasers were better than with one paraphraser or a group of paraphrasers missing. The support vector machine achieved better results when some paraphraser was missing in comparison to other classifiers. As expected, leaving out any of the two considered groups of paraphrasers Paws or T5 decreased the values of the measures more than leaving out only one paraphraser from that group. A significant impact had the missing Eugenesiow/Bart-paraphraser. These results confirm our expectations that combinations of more papaphrasers have a potential to reach better results.

Our last two experiments present results artificially paraphrased descriptions in a real environment. These data came from a real ontology. Hence, the obtained results confirm that it is possible to use artificial paraphrasers to generate paraphrases for training models to detect duplicities and use them in a real environment. So the results in the table 4 show achievable values in considered metrics in the real ontology

Table 4: Comparison of all variants of our approach with the most similar existing approach (Salim and Mustafa, 2021). The dataset for this experiment is based on descriptions of classes encountered in at least three from the ontologies BAO, CHEBI, CHMO, NCIT, SBO, and IUPAC. The results obtained with that approach are in the bottom part of the table.

	ACCURACY	Precision	Recall	F1 score
Gradient boosting	75 %	83 %	75 %	74 %
Gaussian process	78 %	84 %	78 %	77 %
Multi-layer perceptron	**84** %	**85** %	**84** %	**84** %
Random forest	74 %	82 %	74 %	72 %
Support vector machine	79 %	**85** %	79 %	78 %
Team with hard voting	77 %	84 %	77 %	76 %
Team with soft voting	77 %	84 %	77 %	76 %
GloVe with cosine distance	66 %	66 %	66 %	66 %
GloVe with Euclidean distance	65 %	65 %	65 %	65 %
WordNet	71 %	81 %	71 %	68 %

environments.

6 CONCLUSION AND FUTURE WORK

In the automated construction of ontologies, it is often necessary to merge knowledge extracted from scientific articles with the knowledge already contained in the ontology. Merging parts of text from such articles with the text from that ontology can easily introduce duplicities into the ontology. The removal of duplicities in an ontology is often a manual process and automated solutions save the time of domain experts. This process means that two or more terms occurring in different ontologies are associated to unify ontologies. The automated mappings encountered so far focused on the detection of similar class labels or the same URIs of the classes, for example in the bio-ontology bio portal mapping[1]. However, the detection of similar classes based on their description is rather new. In this research, we have focused on the meaning of nodes, relations, and descriptions occurring in ontologies. Our main objective was to mitigate manual effort in dataset preparation to train a model that classifies text in ontologies with respect to their semantic equivalence.

To achieve that objective, we have taken the textual content of an ontology existing for the considered domain. To preprocess the data, we have used paraphrasers, which automatically generate paraphrases with the same or very close meaning. These paraphrases have been embedded using BERT and the embeddings were used to train classifiers to detect duplicates in the ontology.

We have compared our approach with the most similar existing approach (Salim and Mustafa, 2021) based on WordNet and GloVe. The best results have been achieved using the combination of BERT with the multi-layer perceptron or the support vector machine. Both these combinations yielded better results than the existing WorNet-based and GloVe-based approaches. Due to the better consistency between the results from both experiments, we consider support vector machines to be the most suitable kind of classifiers for the detection of duplicates in ontologies.

To assess the importance of using particular paraphrasers and groups of them to generate the training data for the classifiers sets, we have performed the ablation study. The results show the highest impact brought by missing the whole group of paraphrasers or a paraphraser that was alone in its group. In comparison to other classifiers, the support vector machine has been able to keep very good results of all metrics in case one or more paraphrasers were missing. Using all paraphrasers was for almost all combinations of quality measures and classifiers better than with one paraphraser or a group of paraphrasers missing.

In the future, our approach can be improved by further kinds of paraphrasers. The paraphrasers are its core part. Another improvement of our approach may be the usage of some corpus providing a wider range of synonyms. However, this may bring some issues. It is, however, not possible to replace words by synonyms from different domains. For example, the words ”array” and ”field” may be viewed as duplicities in the IT domain, but not in the physics domain. The problem when different text parts of an ontology can be viewed as duplicities, and therefore are replaceable without deteriorating their meaning, definitely requires further research.

[1] https://www.bioontology.org/wiki/BioPortal_Mapping s

Table 5: Comparison of precision and recall results on the 23 independent sets extracted from SBO. The columns with italic names mark where only one of several members of a paraphraser group was removed. The values in the table are mean values ± standard deviations. Green cells (values in bold italic) mark results obtained by the combination of all paraphrasers. White cells (values in italics) mark where the difference to the complete set of paraphrasers is not significant according to the Wilcoxon signed-rank test with correction by the Holm method, and red cells (values in bold) mark where the difference is significant. Pink cells (normal font) mark where mean values are higher than values obtained by the combination of all paraphrasers and the differences are not significant.

ACCURACY	All paraphrasers	Without all PAWS	Without all T5	Without Pegasus	Without BART	*Without Ramsrigouhamg*	*Without Humarin*	*Without Large*	*Without Valurank*	*Without Parrot*	*Without Vamsi*
Gaussian process	***95.8 ±1.41***	*95.4 ±2.02*	*94.6 ±1.97*	*94.5 ±2.28*	*94.9 ±1.92*	*95.1 ±2.51*	*95.3 ±1.66*	*95.5 ±1.60*	*95.4 ±1.62*	*95.6 ±1.64*	*95.6 ±1.70*
Gradient boosting	***87.0 ±2.10***	88.5 ±2.71	*84.9 ±2.78*	*84.7 ±2.75*	*87.0 ±2.96*	*86.6 ±3.08*	*86.7 ±3.05*	*86.3 ±2.89*	87.3 ±2.92	88.7 ±2.48	87.9 ±3.23
Multi-layer perceptron	***89.8 ±3.47***	*88.5 ±3.43*	*86.9 ±3.71*	*89.0 ±2.83*	**86.2 ±3.34**	*89.8 ±2.09*	89.9 ±3.05	*87.6 ±3.21*	*89.8 ±3.01*	*89.6 ±3.24*	90.0 ±2.13
Random forest	***86.8 ±2.30***	88.1 ±2.48	*84.6 ±2.91*	*84.5 ±2.83*	*86.5 ±2.77*	*86.2 ±3.42*	*86.3 ±3.33*	*86.0 ±2.55*	87.0 ±2.89	88.1 ±2.54	87.8 ±3.00
Support vector machine	***96.3 ±1.78***	*95.8 ±1.94*	*95.6 ±1.85*	*95.8 ±1.97*	*95.5 ±1.49*	*95.2 ±2.30*	*95.9 ±1.77*	*96.1 ±1.99*	*95.6 ±1.60*	*96.0 ±1.80*	*96.3 ±1.61*
Team with hard voting	***95.3 ±1.52***	*95.1 ±1.91*	*94.4 ±1.87*	*94.0 ±2.37*	*94.7 ±1.91*	*94.4 ±2.56*	*94.9 ±1.73*	*95.3 ±1.54*	*95.0 ±1.71*	*95.2 ±1.67*	*95.2 ±1.69*
Team with soft voting	***94.8 ±1.25***	*94.8 ±1.98*	*93.8 ±1.92*	*93.9 ±2.77*	*94.4 ±1.91*	*93.9 ±2.68*	*94.6 ±2.33*	*94.5 ±1.88*	*94.8 ±1.70*	95.1 ±1.98	95.2 ±1.84

F1	All paraphrasers	Without all PAWS	Without all T5	Without Pegasus	Without BART	*Without Ramsrigouhamg*	*Without Humarin*	*Without Large*	*Without Valurank*	*Without Parrot*	*Without Vamsi*
Gaussian process	***95.8 ±1.41***	*95.4 ±2.02*	*94.6 ±1.97*	*94.5 ±2.29*	*94.9 ±1.92*	*95.1 ±2.51*	*95.3 ±1.67*	*95.5 ±1.60*	*95.4 ±1.63*	*95.6 ±1.65*	*95.5 ±1.71*
Gradient boosting	***86.8 ±2.19***	88.4 ±2.76	*84.6 ±2.89*	*84.4 ±2.90*	*86.8 ±3.10*	*86.4 ±3.17*	*86.4 ±3.21*	*86.0 ±3.02*	87.1 ±3.06	88.6 ±2.54	87.8 ±3.42
Multi-layer perceptron	***89.7 ±3.56***	*88.4 ±3.56*	*86.7 ±3.89*	*88.9 ±2.88*	**86.0 ±3.54**	89.8 ±2.12	89.8 ±3.16	*87.4 ±3.34*	*89.7 ±3.07*	*89.5 ±3.32*	90.0 ±2.18
Random forest	***86.6 ±2.40***	87.9 ±2.53	*84.3 ±3.02*	*84.2 ±3.02*	*86.3 ±2.91*	*86.0 ±3.56*	*86.0 ±3.54*	*85.8 ±2.66*	86.8 ±3.06	88.0 ±2.61	87.6 ±3.18
Support vector machine	***96.3 ±1.78***	*95.8 ±1.95*	*95.6 ±1.86*	*95.8 ±1.98*	*95.5 ±1.49*	*95.2 ±2.31*	*95.9 ±1.77*	*96.0 ±2.00*	*95.6 ±1.60*	*95.9 ±1.81*	*96.3 ±1.61*
Team with hard voting	***95.3 ±1.52***	*95.1 ±1.91*	*94.4 ±1.88*	*94.0 ±2.38*	*94.7 ±1.91*	*94.4 ±2.57*	*94.9 ±1.73*	*95.3 ±1.54*	*95.0 ±1.72*	*95.2 ±1.68*	*95.2 ±1.70*
Team with soft voting	***94.8 ±1.26***	*94.8 ±1.98*	*93.8 ±1.92*	*93.9 ±2.78*	*94.4 ±1.92*	*93.9 ±2.69*	*94.6 ±2.36*	*94.5 ±1.88*	*94.8 ±1.71*	95.1 ±1.99	95.2 ±1.84

Table 6: Comparison of precision and recall results on the 23 independent sets extracted from SBO. The columns with italic names mark where only one of several members of a paraphraser group was removed. The values in the table are mean values ± standard deviations. Green cells (values in bold italic) mark results obtained by the combination of all paraphrasers. White cells (values in italics) mark where the difference to the complete set of paraphrasers is not significant according to the Wilcoxon signed-rank test with correction by the Holm method, and red cells (values in bold) mark where the difference is significant. Pink cells (normal font) mark where mean values are higher than values obtained by the combination of all paraphrasers and the differences are not significant.

PRECISION	All paraphrasers	Without all PAWS	Without all T5	Without Pegasus	Without BART	*Without Ramsri-gouthamg*	*Without Humarin*	*Without Large*	*Without Valurank*	*Without Parrot*	*Without Vamsi*
Gaussian process	***95.8 ±1.39***	*95.5 ±1.96*	*94.7 ±1.90*	*94.7 ±2.17*	*95.0 ±1.84*	*95.2 ±2.43*	*95.4 ±1.59*	*95.6 ±1.59*	*95.5 ±1.52*	*95.7 ±1.56*	*95.6 ±1.70*
Gradient boosting	***89.1 ±1.75***	89.7 ±2.41	*87.4 ±2.43*	*87.3 ±2.08*	*88.9 ±2.19*	*88.4 ±2.65*	*88.9 ±2.08*	*88.4 ±2.32*	89.2 ±1.99	90.0 ±2.04	89.6 ±2.19
Multi-layer perceptron	***91.1 ±2.73***	*90.1 ±2.46*	*88.7 ±2.91*	*90.0 ±2.56*	*88.3 ±2.27*	*90.8 ±1.88*	*90.9 ±2.40*	*89.3 ±2.41*	*91.0 ±2.50*	*90.9 ±2.63*	*91.1 ±1.71*
Random forest	***89.0 ±1.93***	89.4 ±2.28	*87.2 ±2.54*	*87.2 ±2.02*	*88.6 ±2.11*	*88.2 ±2.83*	*88.6 ±2.09*	*88.2 ±2.14*	*89.0 ±1.90*	89.7 ±1.99	89.4 ±2.07
Support vector machine	***96.4 ±1.68***	*95.9 ±1.78*	*95.7 ±1.78*	*95.9 ±1.92*	*95.6 ±1.48*	*95.3 ±2.27*	*96.0 ±1.67*	*96.1 ±1.92*	*95.8 ±1.48*	*96.1 ±1.72*	*96.4 ±1.57*
Team with hard woting	***95.4 ±1.49***	*95.2 ±1.86*	*94.5 ±1.81*	*94.2 ±2.24*	*94.8 ±1.84*	*94.6 ±2.44*	*95.1 ±1.63*	*95.4 ±1.54*	*95.1 ±1.61*	*95.4 ±1.58*	*95.3 ±1.66*
Team with soft voting	***94.9 ±1.17***	*94.9 ±1.92*	*94.0 ±1.82*	*94.1 ±2.59*	*94.6 ±1.82*	*94.1 ±2.51*	*94.9 ±2.00*	*94.6 ±1.87*	95.0 ±1.56	95.2 ±1.94	95.3 ±1.80

RECALL	All paraphrasers	Without all PAWS	Without all T5	Without Pegasus	Without BART	*Without Ramsri-gouthamg*	*Without Humarin*	*Without Large*	*Without Valurank*	*Without Parrot*	*Without Vamsi*
Gaussian process	***95.8 ±1.41***	*95.4 ±2.02*	*94.6 ±1.97*	*94.5 ±2.28*	*94.9 ±1.92*	*95.1 ±2.51*	*95.3 ±1.66*	*95.5 ±1.60*	*95.4 ±1.62*	*95.6 ±1.64*	*95.6 ±1.70*
Gradient boosting	***87.0 ±2.10***	88.5 ±2.71	*84.9 ±2.78*	*84.7 ±2.75*	*87.0 ±2.96*	*86.6 ±3.08*	*86.7 ±3.05*	*86.3 ±2.89*	87.3 ±2.92	88.7 ±2.48	87.9 ±3.23
Multi-layer perceptron	***89.8 ±3.47***	*88.5 ±3.43*	*86.9 ±3.71*	*89.0 ±2.83*	**86.2 ±3.34**	*89.8 ±2.09*	89.9 ±3.05	*87.6 ±3.21*	*89.8 ±3.01*	*89.6 ±3.24*	90.0 ±2.13
Random forest	***86.8 ±2.30***	88.1 ±2.48	*84.6 ±2.91*	*84.5 ±2.83*	*86.5 ±2.77*	*86.2 ±3.42*	*86.3 ±3.33*	*86.0 ±2.55*	87.0 ±2.89	88.1 ±2.54	87.8 ±3.00
Support vector machine	***96.3 ±1.78***	*95.8 ±1.94*	*95.6 ±1.85*	*95.8 ±1.97*	*95.5 ±1.49*	*95.2 ±2.30*	*95.9 ±1.77*	*96.1 ±1.99*	*95.6 ±1.60*	*96.0 ±1.80*	*96.3 ±1.61*
Team with hard woting	***95.3 ±1.52***	*95.1 ±1.91*	*94.4 ±1.87*	*94.0 ±2.37*	*94.7 ±1.91*	*94.4 ±2.56*	*94.9 ±1.73*	*95.3 ±1.54*	*95.0 ±1.71*	*95.2 ±1.67*	*95.2 ±1.69*
Team with soft voting	***94.8 ±1.25***	*94.8 ±1.98*	*93.8 ±1.92*	*93.9 ±2.77*	*94.4 ±1.91*	*93.9 ±2.68*	*94.6 ±2.33*	*94.5 ±1.88*	*94.8 ±1.70*	95.1 ±1.98	95.2 ±1.84

ACKNOWLEDGEMENTS

The research reported in this paper has been supported by the German Research Foundation (DFG) funded projects NFDI2/12020 for AB and NK, and 467401796 for MH, and by the Grant Agency of the Czech Technical University in Prague, grant No. SGS20/208/OHK3/3T/18 for LK.

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

Chen, W., Zhang, M., and Qin, B. (2011). Coreference resolution system using maximum entropy classifier. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, CONLL Shared Task '11, page 127–130, USA. Association for Computational Linguistics.

Damodaran, P. (2021). Parrot: Paraphrase generation for nlu. 2023-02-11.

Devlin, J., Chang, M.-W., Lee, K., and 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Garanina, N. O., Sidorova, E. A., and Seryi, A. S. (2018). Multiagent approach to coreference resolution based on the multifactor similarity in ontology population. *Programming and Computer Software*, 44(1):23–34.

Huggingface (2019a). Bart paraphrase model (large). 2023-02-11.

Huggingface (2019b). tuner007/pegasus_paraphrase. 2023-02-11.

Lee, K., He, L., and Zettlemoyer, L. (2018). Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 687–692, New Orleans, Louisiana. Association for Computational Linguistics.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

Lu, K., Grover, A., Abbeel, P., and Mordatch, I. (2021). Pretrained transformers as universal computation engines. *CoRR*, abs/2103.05247.

Novák, M. (2017). Coreference resolution system not only for czech. In *Proceedings of the 17th conference ITAT 2017: Slovenskočeský NLP workshop (SloNLP 2017)*, pages 193–200, Praha, Czechia. CreateSpace Independent Publishing Platform.

Ramsrigouthamg, H. (2022a). Ramsrigouthamg/t5-large-paraphraser-diverse-high-quality. 2023-02-11.

Ramsrigouthamg, H. (2022b). Ramsrigouthamg/t5_paraphraser. 2023-02-11.

Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Salim, M. N. and Mustafa, B. S. (2021). Uttokb: a model for semantic relation extraction from unstructured text. In *2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pages 591–595.

Thakur, N., Reimers, N., Daxenberger, J., and Gurevych, I. (2021). Augmented SBERT: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 296–310, Online. Association for Computational Linguistics.

Trieu, H.-L., Duong Nguyen, A.-K., Nguyen, N., Miwa, M., Takamura, H., and Ananiadou, S. (2019). Coreference resolution in full text articles with BERT and syntax-based mention filtering. In *Proceedings of the 5th Workshop on BioNLP Open Shared Tasks*, pages 196–205, Hong Kong, China. Association for Computational Linguistics.

Valurank, H. (2022). Valurank/t5-paraphraser. 2023-02-11.

Vamsi, H. (2019). Paraphrase-generation. 2023-02-11.

Vorobev, V. and Kuznetsov, M. (2023). A paraphrasing model based on chatgpt paraphrases. In *A paraphrasing model based on ChatGPT paraphrases*.

Yang, Y., Zhang, Y., Tar, C., and Baldridge, J. (2019). PAWS-X: A Cross-lingual Adversarial Dataset for Paraphrase Identification. In *Proc. of EMNLP*.

Zhang, J., Zhao, Y., Saleh, M., and Liu, P. J. (2019a). Pegasus: Pre-training with extracted gap-sentences for abstractive summarization.

Zhang, Y., Baldridge, J., and He, L. (2019b). PAWS: Paraphrase Adversaries from Word Scrambling. In *Proc. of NAACL*.