Report - KG Mini Project By ARM

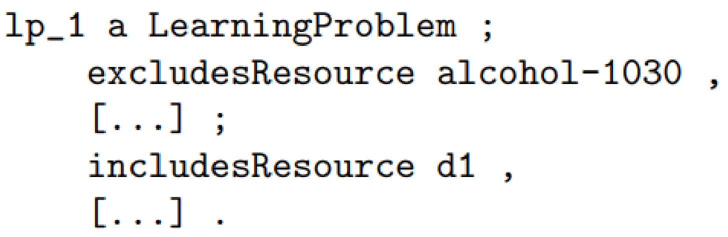
Problem Description:
The project deals with the classification of individuals as carcinogenic or non-carcinogenic based on the carcinogenesis dataset using Machine Learning. A training dataset has been provided which has Learning Problems(LP). Each LP contains classified individuals(carcinogenic in includeResource and non-carcinogenic in excludeResource).

1 Description of the approach

The project deals with the classification of individuals as carcinogenic or non-carcinogenic based on the carcinogenesis dataset using Machine Learning. A training dataset has been provided which has Learning Problems(LP). Each LP contains classified individuals (carcinogenic in includeResource and non-carcinogenic in excludeResource). The working principle of the algorithm is illustrated using a simple example. In this example, all RDF namespaces are omitted for readability.

1.1 Input

An example of an input file can be the following



```
lp_1 a LearningProblem ;
    excludesResource alcohol-1030 ,
    [...] ;
    includesResource d1 ,
    [...] .
```

The code/algorithm makes use of the knowledge graph defined in the carcinogenesis. owl. This file has a resulting ontology containing 142 classes, 19 properties, 22373 instances, and 74567 triples related to chemical compounds.

Our training and test datasets have the learning problem (LP) as lp_1. The LP has positive (+) and negative examples (-)

1.2 Compute Embeddings

The carcinogenesis data has to be converted in a way such that the learner can learn from the data and construct an efficient model. The algorithm labels subject, predicate and object individually from the ontology, assigns ids and forms a proper data frame. Thereafter, The embeddings are generated using TransE.

1.3 Compute Clusters

After computing all the embeddings, these embeddings can be trained with Nearest Neighbor Classifier using the library sci-kit learn where the clusters are learned using all embeddings of the positive and negative individuals. After tuning the hyperparameters, we realized that the hyperparameter nearest neighbour value set to k= 5 gives us the optimal result.

1.4 Classify The Remaining Individuals

After computing the nearest neighbour, they can be utilized to classify the remaining individuals by looking at which of the two classes they belong to. The model then uses the test dataset to determine which of the individuals are positive and negative by going through the test dataset. The entries that are not present in the test dataset but present in the DL-Learner carcinogenesis dataset are given to our KNN model for the prediction of labels.

Subsequently, the predictions are written into the output file.

2 Instruction and Execution of the program

In order to execute the algorithm, a python 3 interpreter is needed. We have used Python version 3.9
The README is included for convenience. We have to use docker to execute the result. First, create the docker image by using the command “docker build -t <custom_name> .” and then use the command to execute the code “docker run <custom_name>” to run the python code. The Dockerfile installs all the packages and creates a build that further can be executed using the docker run command.
The main function will execute the python script and subsequently run the appropriate functions.

The following steps are executed in our python script:

Append data: takes the ontology and creates ids for the labels for subject, predicate and object.

Create data frame embeddings: from the labels that are created, the code then creates embeddings for the labels using torch LongTensor.

Removal of unwanted data: The Ontologies, classes, datatype properties and object properties are removed after the calculation of embeddings for simplification.

Train dataset: the dataset is then trained and tested for predicting the validation scores from the learning problems sampled from the input space. The model's accuracy is evaluated.

3 Results

To evaluate the performance of the implementation, the dataset which contains all individuals for LP was split into a training set and a test set. For each LP, individuals are put into the test set. All remaining individuals are put into the training set.
This process then puts roughly 20% of all individuals in the test set and roughly 80% of all individuals in the training set. Additionally, it is ensured that enough positive examples remain in the training set as the number of positive examples is significantly lower than the number of negative examples. The resulting training and test sets can then be fed into the program. The result of this evaluation is a mean F1-score of 0.6337631686304457.

4 Discussion

The fundamental premise of the classification technique was that the vector space is subject to geometric limitations imposed by the embeddings with TransE. Based on these limitations, we sought to compute a clustering that may benefit from the predicted spatial division. However, this did not turn out as planned, as the results section already makes clear. Our classification is just as accurate as a simplistic baseline that labels every occurrence as a bad example. We identified two causes for this subpar performance. First, our model might not be able to learn anything useful because the training data is so biased. Second, we believe the geometric constraints may not be as beneficial and useful as we had thought. We believe that the large amount of data paired with the few classes and relations leads to a suboptimal embedding that is unable to satisfy the equation $s + p \approx o$ only for most facts (s, p, o).