Mappathon GMDS 2018
Marvelous Mapper

Stefan Hegselmann[1*], Philipp Neuhaus[1*], Michael Storck[1*]

[1]Institute of Medical Informatics, University of Münster
[*]all authors contributed equally

1 Introduction

To enable secondary use of medical data, it is inevitable to understand the meaning of the data items that should be reused. Both syntactical and semantical information of the data items have to be considered to gain this understanding. These information are included in the data items' metadata. Based on the metadata, relations between different items can be found and it is possible to decide if stored data for items in different information systems is related and how it can be combined. Objective of this work is to establish an automated process to find related data items in two different sets of forms including multiple data items.

2 Methods

We generate features for item pairs to encode relationships between two item definitions. The ground truth item relationships serve as labels for these features. Hence, task 2 is reduced to a multiclass classification in which we predict a label from features of an unseen item pair. For task 1, we simply use the item pairs of task 2 that were labeled as *equivalent*. We apply two different methods. First, we treat the problem as a *classification task* and use classification models to learn the unknown mapping to target labels. We perform experiments with linear support vector classification and k-nearest neighbors. Second, we reformulate the problem as a *regression task* where we treat the labels as continuous values and discretize the prediction results back to classes. This is motivated by the dependency of the different relation classes and we aim to incorporate the features of all labels for a single model. For instance, this is not the case for classification with linear support vector classification, where separate models for each label are trained and compete against each other. We evaluate linear regressions and support vector regression as regression models. Experiments are implemented with the Python libraries Pandas for data processing and scikit-learn for machine learning.

To test the feature sets and the trained machine learning procedure, a cross validation test was implemented. Within every run (total:10) the training data was shuffled and 90% of the data was used to train and the other 10% was used to predict mappings. The predicted mappings were verified using the ground truth data and the Mappathon score was calculated. Afterwards, the false negative and false positive mappings were analysed to improve the existing or generate new features.

2.1 Feature Selection

At first, some naive features were defined like equality of the names and data types of the compared items. In the second step more sophisticated comparison between item names, descriptions and questions where implemented using stemming and returning the proportional overlap of the stemmed words. Using the same algorithm also a combination of item names and item group names were compared.

Since the output of the trained machine learning procedure was not satisfactory, we had to incorporate external sources that include additional metadata. One of these source is the Metadata Repository (MDR) of the Portal of Medical Data Models (MDM Portal). A publicly accessible web service was developed to suggest semantic codes from the Unified Medical Language System (UMLS) for given phrases. The MDR contains a sophisticated and fast search mechanism based on more than 330,000 semantic code suggestions derived from semantically annotated data elements contained in the MDM Portal manually curated by medical professionals.[1] The MDR was queried using the names and English and German questions of the data items. The top 1, 5 and 10 retrieved UMLS codes were used to calculate the overlap comparing the retrieved codes for both items.

Another source is an internally used tool called MultiMapper.[2] For this challenge we configured it to use the UMLS Metathesaurus to find suitable UMLS-Codes for items based on their English or German question. Similar to the features using the MDR this feature calculates the similarity between the top answer of the MultiMapper, the best 5 and 10.

3 Results

3.1 Cross validation testing

There are 887 items in the training data sets resulting in a total of 77,385 combinations. In Table 1 the results for the cross validation testing are shown. As mentioned in the Methods four different machine learning models were tested. The resulting score is the sum of the Mappathon scores of all 10 runs. Since, the 2-nearest neighbor vote produces the best score, it will be chosen for the prediction of the evaluation datasets.

3.2 Analysis of false positive and false negative mappings

Feature generation is the hardest part while implementing machine learning procedures. Thus, it is crucial to review the false negative and false positive

[1] Hegselmann S. et al. A Web Service to Suggest Semantic Codes Based on the MDM-Portal. Stud. Health Technol. Inform. 2018;253:35-39. Available from: http://ebooks.iospress.nl/publication/50019

[2] Neuhaus P. et al. Standardized Mappings A Framework to Combine Different Semantic Mappers into a Standardized Web-API. Stud. Health Technol. Inform. 2015;212:23-6.

Table 1: Results of cross validation testing

Method	**Model**	**10-fold CV Mappathon Score**
Classification	Linear Support Vector Classification	1.0
	1-nearest neighbors vote	-9.9
	2-nearest neighbors vote	18.2
	3-nearest neighbors vote	13.0
	4-nearest neighbors vote	8.0
	5-nearest neighbors vote	9.6
	10-nearest neighbors vote	10.2
Regression	Linear Regression	-56.2
	Support Vector Regression	3.8

mappings to learn which feature differentiates well between correct and incorrect mappings and which characteristics of the metadata can be used for new features.

While examinig the mappings, some potential errors appeared in the ground truth data as shown in Table 2. Some possible related data items showed up in the false positive mappings. So the algorithm detected relations even though they were not listed in the ground truth. The false negative mappings showed possible false mappings in the ground truth data. The first two lines indicate that the mappings are shifted for one item (S.0011_IG.4_I.158 should be mapped to S.0021_IG.2_I.5, S.0011_IG.4_I.159 to S.0021_IG.2_I.6 and etc.).

Table 2: Potential errors in ground truth data

Validation result	Reference item	Mapped item	ground truth	prediction
False positives	S.0011_IG.4_I.28 (Temperatur)	S.0023_IG.2_I.12 (Temperatur)	undefined	equivalent
	S.0011_IG.9_I.85 (ROTEM Zeit)	S.0023_IG.3_I.30 (Zeitpunkt ROTEM)	undefined	equal
	S.0011_IG.4_I.18 (SpO2)	S.0023_IG.2_I.13 (Sauerstoffsättigung (SpO2))	undefined	equivalent
False negatives	S.0011_IG.4_I.158 (Augenöffnung)	S.0021_IG.2_I.6 (Verbale Antwort)	equivalent	undefined
	S.0011_IG.4_I.159 (Verbale Antwort)	S.0021_IG.2_I.7 (Motorische Antwort)	equal	undefined
	S.0011_IG.1_I.2 (Patienten-ID)	S.0021_IG.3_I.11 (Alarmierung)	equivalent	undefined

4 Outlook and future work

We showed that our method is feasible to identify possible mappings between routine and research data models. Nevertheless, our method depends on correct ground truth data and selective features. To improve the selectiveness of the features, we have to develop further debug-tools to identify weak and strong methods. Furthermore, a skript that presents false positives and negatives with their names, questions and itemgroups would be helpful to get a quick overview of the quality of the ground truth or reference data.

At the moment, our classifier can not predict ”narrower” or ”wider” between two items - it treats them as ”relatedTo” instead. An option to face this would be a second run with other training data and other features only with the items identified as ”relatedTo” in a first run.