UNIGE_SE @ PRELEARN: Utility for Automatic Prerequisite Learning from Italian Wikipedia

Alessio Moggio, Andrea Parizzi
DIBRIS, Università degli studi di Genova
{s4062312, s4048705}@studenti.unige.it

Abstract

The present paper describes the approach proposed by the UNIGE_SE team to tackle the EVALITA 2020 shared task on Prerequisite Relation Learning (PRELEARN). We developed a neural network classifier that exploits features extracted both from raw text and the structure of the Wikipedia pages provided by task organisers as training sets. We participated in all four sub–tasks proposed by task organizers: the neural network was trained on different sets of features for each of the two training settings (i.e., raw and structured features) and evaluated in all proposed scenarios (i.e. in– and cross– domain). When evaluated on the official test sets, the system was able to get improvements compared to the provided baselines, even though it ranked third (out of three participants). This contribution also describes the interface we developed to compare multiple runs of our models. [1]

1 Introduction

Prerequisite relations constitute an essential relation between educational items since they express the order in which concepts should be learned by a student in order to allow a full understanding of a topic. Therefore, automatic prerequisite learning is a relevant task for the development of many educational applications.

Prerequisite Relation Learning (PRELEARN) (Alzetta et al., 2020), a shared task organized within EVALITA 2020, the 7th evaluation campaign of Natural Language Processing and Speech tools for Italian (Basile et al., 2020), has, as a purpose, automatic prerequisite relation learning between pairs of concepts. For the purposes of the shared tasks, concepts are represented as learning materials written in Italian. In particular, each concept corresponds to a page of the Italian Wikipedia having the concept name as title. The goal of the shared task is to build a system able to automatically identify the presence or absence of a prerequisite relation between two given concepts. The task is divided in four sub-tasks: specifically in order to make a valid submission participants are asked to build at least one model for automatic prerequisite learning to be tested both in in- and cross-domain scenario since task organisers released four official training sets, one for each domain of the dataset. Concerning the model, it can exploit either 1) information extracted from the raw textual content of Wikipedia pages, 2) information acquired from any kind of structured knowledge resource (excluding the prerequisite labelled datasets). Eventually, we submitted our results on the official test sets for all four proposed subtasks. To tackle the problem proposed in the shared task, we propose an approach based on deep learning to classify on different sets of features in order to comply with the sub-tasks requirements. We also developed a user interface to support the comparison between the results obtained running the model trained using different sets of features. Other than selecting which features should be used to train the model, the user can exploit the interface to define the value of a set of parameters in order to customize the classifier structure. The interface reports, for each run, standard evaluation metrics (i.e., accuracy, precision, recall and F-score) and other statistics that allow to explore the model performances.

The remainder of the paper is organised as follows: we present our approach and system in Section 2, then we discuss the results and evaluation (Section 3). Section 4 describes the interface in

detail. We conclude the paper in Section 5.

2 System Description

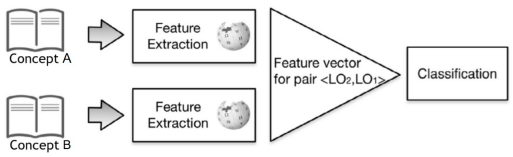

Figure 1: System architecture

In this Section we present our approach for automatic prerequisite learning between Wikipedia pages. We exploited a deep learning model that can be customised by the user on a dedicated GUI. The model was trained and tested on the official dataset of the PRELEARN task.

ITA-PREREQ (Miaschi et al., 2019) is a binary labelled dataset in which the labels stand for the presence or the absence (1 or 0) of the prerequisite relation between a pair of concepts. Each concept is an educational item associated to a Wikipedia page, therefore the concept name matches the title of the equivalent Wikipedia page. Hence, the dataset released for the shared task consists also of the content and the link of the Wikipedia pages referring to the concepts appearing in the dataset. It covers four domains, namely precalculus, geometry, physics and data mining.

2.1 Classifier

The classifier was built with the aim of testing the combination of different hand–crafted features on the automatic prerequisite learning task. More specifically our classifier, whose architecture is described in Figure 1, uses a two–dense–layers Neural Network built using Scikit-Learn and Keras libraries (wrapped for Tensorflow). The activation function for the hidden layer is ReLU while the Adam optimizer (Kingma and Ba, 2014) is used as training algorithm. The output layer consists of one neuron with sigmoid activation function.

Some structural properties of the classifier can be customised by the user from a dedicated GUI. In particular, for what concerns the structure of the neural network the user can define the size of the hidden layer and the number of epochs, while for the evaluation the user can set the number of cross validation folds. Moreover, training can be performed on a customizable set of features (see Section 2.2 for the complete list) since the input layer is set to dynamically match the size of the feature vector. For the specific purposes of this work, we used in every scenario a model exploiting a 20 neurons hidden layer trained on 15 epochs. A 4-fold cross validation was used for the in-domain scenario.

Training The official training set containing concept pairs and their binary labels was formatted as a pair of numpy arrays: one of them has variable length and contains the serialization of the features, which will be the model input, whilst the latter contains the binary labels of the pairs. For the in-domain scenario, the model was trained using stratified random folds of concept pairs that preserve the original proportion of domains' pairs. For the cross-domain evaluation scenario, a "leave one domain out" approach was used, training the model on all domains but the one used for test.

2.2 Features

We defined a set of features extracted from the Wikipedia page content and structure that are available in the GUI and can be selected by the user to train his model. While the pages content was provided in the official release of the training set, we exploited Wikipedia API [2] to extract the Wikipedia metadata and knowledge structure. Depending on the sub-task requirements, we trained our models with a different combination of features.

- Features used for the raw features model:
  - **titleInText**: given a pair (A, B), it checks if the title of page A/B is mentioned in the page of the other concept.
  - **Jaccard similarity**: a concept-based metric that measures the similarity between two pages by the number of words shared between them.
  - **LDA**: the Shannon Entropy of the LDA (Deerwester et al., 1990) of nouns and verbs in *A* and *B*. Nouns and verbs are identified thanks to a morpho-syntactic analysis of the page content performed by UDPipe pipeline (Straka and Straková, 2017).

[2] https://github.com/martin-majlis/Wikipedia-API

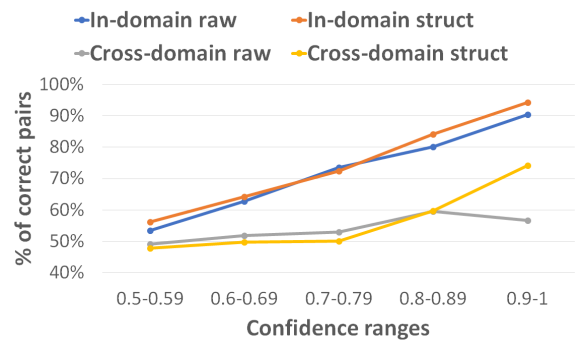

Figure 2: Variation of correctly labelled pairs wrt the classifier confidence for all submitted models.

  - **LDA Cross Entropy**: the cross entropy of the LDA vectors A\B.

- Features used for the raw and structural features model. We exploited all the above features combined with the followings:

  - **extractCategories**: the Wikipedia category(s) to which each page of the pair (A, B) belongs.
  - **extractLinkConnections**: for each pair of concepts (A, B) checks if the Wikipedia page of B contains a link to A.
  - **totalIncoming/OutgoingLinks**: it computes how much a concept is linked to/from other concepts.
  - **Reference distance**: a link-based metric that measures the relation between two pages by the links contained in each of them using the EQUAL weight (Liang et al., 2015).

3 Results and Error Analysis

Table 1 reports the results obtained by our models on the runs submitted for all four sub-tasks. On average, the performances of our systems in the different scenarios show that, as expected, training the model in a in–domain scenario allows to achieve better results. Among the different sets of features used, exploiting structural information extracted from Wikipedia pages' structure is in general more effective than relying only on raw textual data. Thus, our best performing model is the one exploiting both raw and structural features evaluated in-domain scenario which achieves an average accuracy computed across all four domains of 0.700. Interestingly Data Mining constitutes the only case where raw textual features are more effective than a combination of raw and structural features. In fact this domain shows lower accuracies in the structured settings, possibly due to the lower number of entries within the dataset of Data Mining with respect to the other three domains and to the lower coverage of Wikipedia.

If we compare our results obtained for each domain with those obtained by the official baseline, there are only two cases where our models do not outperform the baseline, i.e. Geometry in the raw feats in-domain subtask and physics in both cross-domain subtasks. During error analysis on geometry pairs, we observe that, while pages about geometric figures, e.g. "Rettangolo" and "Poligono", show a prerequisite relation in the gold dataset, our systems always fail to correctly classify them. Concerning Physics, we observe that in both cross-domain settings the classifier did not consider the page "Fisica" as prerequisite of other pages belonging to the Physics domain, causing the performances to be below baseline.

If we look at the variation of accuracy values for each model with respect to the classifier confidence (see Figure 2), we notice that although the four systems have a similar accuracy when the confidence is low those related to the two in-domain settings show a similar increase in accuracy confidence. Comparing cross-domain settings we notice that only the structured one is able to reach higher accuracy but only when it is highly confident.

4 System Interface

Together with our system we also developed a User Interface aimed at personalizing the network and comparing results obtained with different models. The interface is composed of the following three modules: i) setup module; ii) results module; iii) statistics module.

The setup module, loaded at the start of the program, allows to define:

- The input dataset;
- The parameters to setup the neural network architecture;
- The features for training the model.

Sub-Task	Data Mining	Geometry	Physics	Precalc	AVG
Raw Feats in-domain	0.595	0.620	0.530	0.675	0.605
Raw+Struct in-domain	0.565	0.755	0.725	0.755	**0.700**
Baseline in-domain	0.494	0.675	0.500	0.675	0.586
Raw Feats cross-domain	0.565	0.515	0.465	0.595	0.535
Raw+Struct cross-domain	0.545	0.665	0.560	0.710	**0.620**
Baseline cross-domain	0.494	0.500	0.605	0.500	0.525

Table 1: Results obtained for each sub-tasks by our models and the baseline on PRELEARN official test sets.

The module includes also a table where previously saved Configurations can be selected in order to run them again.

After running the model, the user can reach the results module in which are printed the performance statistics (accuracy, precision, recall, F-score) achieved by the performed configuration. Besides, the result module is composed of different buttons that allows to:

- Save the performed configuration.
- See the results of the classifier on concept pairs labelling.
- Save and download the results as csv file or txt file.

The statistics module plots in four bar charts the values of accuracy, precision, F-score and recall of all configurations saved in the interface. The repository containing the system and its GUI can be consulted on github [3].

5 Conclusion

In the paper we described the approach proposed by the UNIGE_SE team for the EVALITA 2020 PRELEARN shared task. The classifier relied on a set of features that was customised to address the specific requests of each sub-task. The results obtained by our models are all above baseline (if considered averaging the accuracies across all domains), although in some cases the results obtained by the baseline are still highly competitive. This suggests that automatic prerequisite learning is a difficult task requiring many different information to train the models. However, the obtained results suggest that, at least in a in-domain setting, features extracted from raw texts are sufficient to achieve competitive results. In the cross-domain setting exploiting only this type of features is not enough. Nevertheless, using information extracted from knowledge structures allows to achieve better results in all sub-tasks. Although our obtained results are promising, future work will be focused on analyzing the impact of each feature in training the model and exploring the inclusion of new features to improve the performance of the classifier.

[3] https://github.com/mnarizzano/se20-project-16

References

Chiara Alzetta, Alessio Miaschi, Felice Dell'Orletta, Frosina Koceva, and Ilaria Torre. 2020. Prelearn@evalita 2020: Overview of the prerequisite relation learning task for italian. In Valerio Basile, Danilo Croce, Maria Di Maro, and Lucia C. Passaro, editors, *Proceedings of Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)*, Online. CEUR.org.

Valerio Basile, Danilo Croce, Maria Di Maro, and Lucia C. Passaro. 2020. Evalita 2020: Overview of the 7th evaluation campaign of natural language processing and speech tools for italian. In Valerio Basile, Danilo Croce, Maria Di Maro, and Lucia C. Passaro, editors, *Proceedings of Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)*, Online. CEUR.org.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Chen Liang, Zhaohui Wu, Wenyi Huang, and C Lee Giles. 2015. Measuring prerequisite relations

among concepts. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1668–1674.

Alessio Miaschi, Chiara Alzetta, Franco Alberto Cardillo, and Felice Dell'Orletta. 2019. Linguistically-driven strategy for concept prerequisites learning on italian. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 285–295.

Milan Straka and Jana Straková. 2017. Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99.