Towards Knowledge Acquisition of Metadata on AI Progress

Zhiyu Chen*, Mohamed Trabelsi*, Brian D. Davison, Jeff Heflin

Lehigh University, Bethlehem, PA, USA
{zhc415,mot218,davison,heflin}@cse.lehigh.edu

Abstract. We propose an ontology to help AI researchers keep track of the scholarly progress of AI related tasks such as natural language processing and computer vision. We first define the core entities and relations in the proposed Machine Learning Progress Ontology (MLPO). Then we describe how to use the techniques in natural language processing to construct a Machine Learning Progress Knowledge Base (MPKB) that can support various downstream tasks.

Keywords: dataset search · information extraction · ontology · knowledge base

1 Introduction

In recent years, there has been a significant increase in the number of published papers for AI related tasks, and this leads to the introduction of new tasks, datasets, and methods. Despite the progress in scholarly search engines, it is challenging to connect previous technologies with new work. Researchers from the semantic web community have noticed the importance of organizing scholarly data from a large collection of papers with tools like Computer Science Ontology [10]. Natural language processing researchers have proposed methods to extract information from research articles for better literature review [8]. Different from previous work which focuses on the extraction of paper metadata and key insights, we propose to design an ontology and knowledge base for better evaluation of AI research. Papers With Code[1] is a website that shows the charts of progress of machine learning models on various tasks and benchmarks. Those charts can help researchers to identify the appropriate literature related to their work, and to select appropriate baselines to compare against. Although manually updating this leaderboard may keep it accurate, it will become more difficult and time consuming because of the large increase in published papers.

Knowledge extraction from research papers has been studied by the information extraction (IE) community for years. Hou et al. [4] extract ⟨*Task*, *Dataset*, *Metric*, *Score*⟩ tuples from a paper where the paper content is extracted from pdf files. In their two-stage extraction framework, they first extract ⟨*Task*, *Dataset*, *Metric*⟩ tuples, and then for each tuple, they separately extract ⟨*Dataset*, *Metric*, *Score*⟩ tuples. Kardas et al. [7]

* equal contribution

[1] https://paperswithcode.com/

specifically focus on extracting table results by taking the advantage of available latex source code of papers. Work developed in parallel to ours is proposed by Jain et al. [6], which uses the data from Papers With Code as a distant supervision signal and introduces a new document-level IE dataset for extracting scientific entities from papers. Our work is complementary to AI-KG [2] which takes the abstract of a paper as input. We also consider other sections and tables in a paper where the evaluation scores of different metrics always occur. Our ontology can be considered as the front end of a knowledge system that organizes all the extracted knowledge from different backend IE tasks.

In this paper, we first introduce the Machine Learning Progress Ontology (MLPO) which defines the core entities and relations useful for progress tracking of AI literature. Then, we propose to construct the Machine Learning Progress Knowledge Base (MPKB) from a paper corpus using information extraction techniques. The ontology definition and pipeline of knowledge construction are available online[2].

2 Machine Learning Progress Ontology

As shown in Figure 1, the MLPO focuses on the results of machine learning experiments, which differentiates it from prior work. This ontology defines five core classes: *Task*, *Dataset*, *Result*, *Model* and *Paper*. To support proper citation of results, it also includes general properties such as Venue, Author and Title which have already been defined in the BIBO ontology[3]. In total, MLPO has 22 classes, 18 object properties and 24 data properties.

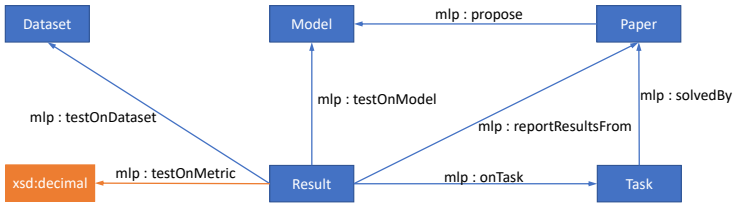

Fig. 1: The main classes and relations in MLPO. The blue arrow means object property and the orange arrow means data property.

It is important to notice that the *Result* class connects to all other core classes. From a single paper, we could extract multiple *Result* individuals and each *Result* individual records the used dataset, the used model, the target task and also the reported evaluation score. For *Task* class, we create different subclasses representing different AI tasks (e.g., natural language processing task). We create various data properties for evaluation metrics which have different range constraints. For example, the range of data property

[2] https://github.com/Zhiyu-Chen/Machine-Learning-Progress-Ontology
[3] https://www.dublincore.org/specifications/bibo/

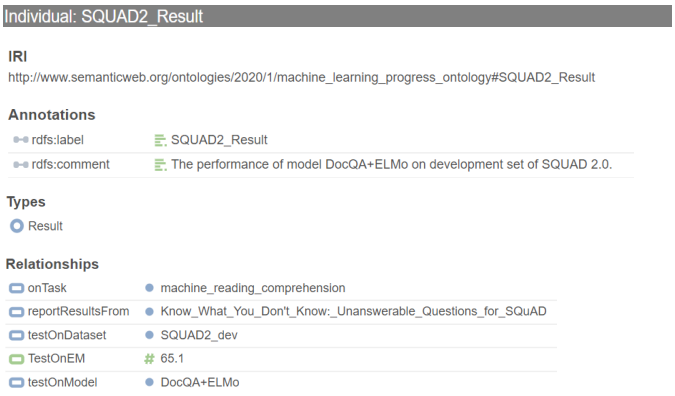

Fig. 2: An example of extracted Result individuals.

"TestOnEM" which represents the exact matching metric, is a decimal as shown in Figure 2. We use WebProtégé[4] to develop our ontology and an example of extracted individuals is shown in Figure 2.

3 Knowledge Base Construction

Constructing the Machine Learning Progress Knowledge Base (MPKB) involves two tasks: scientific entity recognition (SER) and relation classification (RC). For the SER task, we identify the core entities in a paper which are datasets, tasks and metrics. For the RC task, for simplicity, here we only show how to identify two relations in a paper: whether a dataset is used for a task and evaluated with a metric. For the example in Figure 2, we would like to know whether "SQUAD2_dev" is used for the task of "machine_reading_comprehension" and is evaluated with "TestOnEM". We believe the methods can also be applied to recognize other entities and relations. We leave extracting all the mentioned relations defined in MLPO to future work.

3.1 Scientific Entity Extraction

We treat entity extraction as a sequence tagging problem. One challenge is that we only have document-level instead of sequence-level annotations. As a solution, we use fuzzy matching to find the entity spans in a paper. Given the text of a paper, we first use spaCy[5] to find the noun phrases. Then we match the noun phrases with pre-curated entity names using the similarity measure based on Levenshtein Distance[6]. For tasks and metrics, we set the similarity threshold to 0.5. For datasets, we set the matching threshold to 1 (i.e.,

[4] https://webprotege.stanford.edu/
[5] https://spacy.io/
[6] https://chairnerd.seatgeek.com/fuzzywuzzy-fuzzy-string-\
matching-in-python/

exact match). If the fuzzy matching similarity between a noun phrase and an entity name is larger than the corresponding threshold, then we annotate the noun phrase as the target entity. We also designed a tagging schema similar to BILOU [9]. For section titles in the paper, we annotate every token either as at the first, middle or last position. For every sentence in each section, we tag the word as at the first, middle or last position of the sentence. For tokens belonging to an entity in a sentence, we tag them with the corresponding entity types. Based on the paper text and annotated tags, we train a Bi-LSTM-CRF model [5] to predict the tags of test data.

3.2 Relation Classification

We use an information retrieval method for relation classification. To construct the query q, we concatenate the text of a result tuple $\langle$*Task*, *Dataset*, *Metric*$\rangle$. We select the first 100 tokens from each section of a paper as its text representation T_p. Finally, we match the two inputs with a neural ranking model. In particular, we use Conv-KNRM [1] to predict the binary relevance score of a triple-paper pair:

$$label = ConvKNRM(q, T_p) \tag{1}$$

The label is equal to 1 if the triple is relevant to the paper, otherwise the label is equal to 0. We choose Conv-KNRM in this paper because it is efficient. A state-of-the-art model like BERT [3] can also be used as in Hou et al. [4].

4 Experiments and Evaluation

We randomly divided the paper collection of the **NLP-TDMS** dataset [4] into training (80%) and testing (20%) sets. For the Bi-LSTM-CRF model, we set the embedding dimension to 100. We use a bi-LSTM with 2 layers. When training relation classification, we create k positive result tuple-paper pairs (one for each tuple used to annotate the paper) and $n - k$ negative pairs, where n is the total number of result tuples in the ground truth. This results in many more negative samples than positive samples: 94% of result tuple-paper pairs are negative. To address this imbalance, we oversample the positive class by creating 20 copies of each positive sample.

From the result tables, we can see that among different entity types, *Task* is the easiest type to recognize. *Dataset* has higher precision but lower recall than *Metric*. Such variances may indicate that tasks have more observable patterns to appear in a paper than other entity types, so that the predicted sequence tagging is more accurate. Conv-KNRM achieves high results on all the evaluation metrics for predicting irrelevant paper-triple pairs. The most challenging part for the neural network is to capture the semantic similarities between paper content and $\langle$*Task*, *Dataset*, *Metric*$\rangle$ triple for positive pair.

Tag	Precision	Recall	F1
Task	0.99	1.00	0.99
Dataset	0.66	0.36	0.46
Metric	0.44	0.46	0.45

Paper-triple label	Precision	Recall	F1
Irrelevant (0)	0.93	0.99	0.96
Relevant (1)	0.98	0.51	0.67

5 Conclusion

We have proposed an ontology specifically designed for progress tracking of AI tasks. We also proposed methods to extract information from papers to construct a knowledge base for AI evaluation. The resulting knowledge graph can be used for various downstream tasks. For example, we can request the system to return the top-k text classification models ranked by accuracy on Yelp reviews dataset [11] by constructing the corresponding SPARQL query. Combined with methods of document summarization, we may be able to automatically generate a survey paper for a given task.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. IIS-1816325.

References

1. Dai, Z., Xiong, C., Callan, J., Liu, Z.: Convolutional neural networks for soft-matching n-grams in ad-hoc search. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining. p. 126–134 (2018)
2. Dessi', D., Osborne, F., Recupero, D.R., Buscaldi, D., Motta, E., Sack, H.: Ai-kg: an automatically generated knowledge graph of artificial intelligence. In: International Semantic Web Conference. Springer (In Press) (2020)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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. pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019)
4. Hou, Y., Jochim, C., Gleize, M., Bonin, F., Ganguly, D.: Identification of tasks, datasets, evaluation metrics, and numeric scores for scientific leaderboards construction. In: 57th ACL. pp. 5203–5213 (Jul 2019)
5. Huang, Z., Xu, W., Yu, K.: Bidirectional lstm-crf models for sequence tagging. arXiv preprint arXiv:1508.01991 (2015)
6. Jain, S., van Zuylen, M., Hajishirzi, H., Beltagy, I.: SciREX: A challenge dataset for document-level information extraction. In: Proc. 58th Annual Meeting of the Association for Computational Linguistics. pp. 7506–7516. Online (Jul 2020)
7. Kardas, M., Czapla, P., Stenetorp, P., Ruder, S., Riedel, S., Taylor, R., Stojnic, R.: Axcell: Automatic extraction of results from machine learning papers. arXiv preprint arXiv:2004.14356 (2020)
8. Nasar, Z., Jaffry, S.W., Malik, M.K.: Information extraction from scientific articles: a survey. Scientometrics **117**(3), 1931–1990 (2018)
9. Ratinov, L., Roth, D.: Design challenges and misconceptions in named entity recognition. In: Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009). pp. 147–155 (2009)
10. Salatino, A.A., Thanapalasingam, T., Mannocci, A., Osborne, F., Motta, E.: The computer science ontology: a large-scale taxonomy of research areas. In: International Semantic Web Conference. pp. 187–205. Springer (2018)
11. Tang, D., Qin, B., Liu, T.: Document modeling with gated recurrent neural network for sentiment classification. In: Proceedings of the 2015 conference on empirical methods in natural language processing. pp. 1422–1432 (2015)