Graph Sequence Learning for Premise Selection

Edvard K. Holden and Konstantin Korovin

The University of Manchester, U.K.

Abstract

Premise selection is crucial for large theory reasoning as the sheer size of the problems quickly leads to resource starvation. In this work we propose a premise selection approach inspired by the domain of image captioning, where language models automatically generate a suitable caption for a given image. Likewise, we attempt to generate the sequence of axioms required to construct the proof of a given problem. This is achieved by combining a pre-trained graph neural network with a language model. We evaluate different configurations of our method and experience a 17.7% improvement gain over the baseline.[1]

Introduction. State-of-the-art first-order theorem provers (ATPs) such as iProver [4, 10], Vampire [11], E [20] and SPASS [24] attempt to solve problems consisting of a conjecture and a set of axioms through saturation. In many applications, such as formalisation of mathematics or verification, theorem provers have to deal with large collections of axioms where only few of them are needed for the proof of the conjecture. It is often the case that if ATP is supplied only with relevant axioms the problem can be quickly solved and if all axioms are submitted the proof search is flooded with irrelevant conclusions and the proof is never found. Thus selecting relevant axioms, aka the premise selection problem, is crucial in solving problems with large theories. There are many works that studied premise selection problem at different angles: based on syntactic relevance [6, 14] and on different machine learning methods [1, 2, 8, 12, 17, 18, 19, 23].

In this work we explore the adaptation of image captioning language models to the premise selection task. Image captioning models aim to produce a sentence in natural language that describes a given image [26]. The captions are generated by embedding images using a pre-trained image model and combining the embedding with a language model. These models had shown to be very powerful and we believe that this success can be transferred to the premise selection problem.

Our approach can be split into two major tasks: first is problem embedding and second is language-based model for premise selection based on this embedding.

Problem embeddings using GNN. First-order problems consist of a set of tree-structured formulas which are not easily represented through a feature vector, as required for machine learning. Recent approaches for formula embeddings are based on graph neural networks (GNNs) [15, 16, 19]. We adapt approach from [19]. In this approach, a formula is represented as a directed acyclic graph. The vertices correspond to the types of elements occurring in the formula, such as (anonymised) function applications. The edges denote a relationship between the vertices, e.g., an argument supplied to a function. This representation extends to sets of formulas by computing a global graph over the node elements in the formulae. The graph representation captures many aspects of the formulae while invariant to symbol renaming and encoding problems with previously unseen symbols. This paper uses a graph encoding of 17 node types as described in [19].

A graph neural network is an optimisable transformation that operates on the attributes of a graph. It utilises a "graph-in, graph-out" methodology where it embeds the graph while preserving the structure and connectivity of the original graph. A randomly initialised vector represents each node type across

[1] Full version of this paper can be found at https://arxiv.org/abs/2303.15642

all graphs in an n-dimensional embedding space. Next, each node in a graph is assigned to its corresponding embedding vector, resulting in the node feature matrix. The GNN embeds the type features of each node into the node feature embedding through a node update function. This effectively transforms the graph features into a more favourable embedding which is done through message passing over neighbouring nodes [5]. Message passing is accomplished through graph convolutional layers, and we utilise the operations described in [9]. Finally, the problem graph is embedded into a feature vector using a pooling operation such averaging over all node embeddings.

One of the challenging tasks is the GNN training. In this work we investigate pre-trained graph neural networks (GNNs) to embed problems via transfer learning. We investigate two approaches: 1) supervised GNN pre-training using the binary classification premise selection task; and 2) unsupervised training based on graph similarities.

For the supervised task we add an extra prediction layer to the GNN, after the pooled embedding, which outputs binary prediction of the axiom relevance in the problem. This combined network is trained on the DeepMath dataset.

Although, supervised approach performs well, it requires labelled data which could be expensive to obtain and might be not feasible for large data-sets. Hence, we also investigated unsupervised approach which does not require external labelling. The unsupervised training approach consists of training a matching model which learns the distance between two problem graphs according to some metric. We adapt approach from [21]. The model takes two graphs, as input and passes them through the siamese GNN model, to predict the graph similarity. As a similarity measure we use Laplacian spectrum distance [25]. The Laplacian spectrum distance is a computationally cheap metric, even for graphs of the magnitude required to represent first-order problems.

Axiom Captioning After we considered problem embeddings using GNNs, we construct an *axiom captioning* model which takes this embedding as an input and predicts sequences of relevant axioms. This approach is inspired by image capturing models [26]. In the context of premise selection, the images are replaced by problems and the captions are replaced with the axioms that appear in the proof of the problems. We describe the task of premise selection in the context of sequence learning as maximising the probability of producing the sequence of axioms used in the proof of a given problem. We estimate conditional probabilities of the next relevant axiom with the recurrent neural network (RNN). The generative axiom prediction model is constructed using the par-inject architecture [22]. This architecture takes a token embedding **s** and a problem embedding **I** at each time step. The model is given the special start token s_{start} to initialise the axiom generation process. Likewise, a special end token, s_{end}, represents the end of a sequence. Consequently, start and end tokens are added to each axiom sequence such that the model is trained on the target sequence $\langle s_{start}, s_1, \ldots, s_m, s_{end}\rangle$. Axioms with few occurrences in the dataset are replaced by the Out-Of-Vocabulary token s_{unkown}. These three special tokens are included in the dictionary Ω. We experimented with different attention based RNN architectures which emphasise different components of the embedding vector in the time-dependent manner [3, 13, 26].

Results We conducted a range of experiments on DeepMath problems [2] to evaluate effects of different graph embeddings, attention mechanisms, input axiom orders, decoder sampling methods, and combination of syntactic premise selection SinE [6] with our axiom captioning method. Details of these experiments can be found in [7]. Here we show only online evaluation, Figure 1, which is evaluation of the effect of different premise selection methods on an ATP performance (iProver) in a realistic setting, where DeepMath problems (which are balanced for ML training) are extending corresponding axioms from the Mizar40 benchmarks. We can see that our axiom captioning method outperforms other ML-based premise selection methods that we tried, stand-alone ML-based premise selectors are still behind

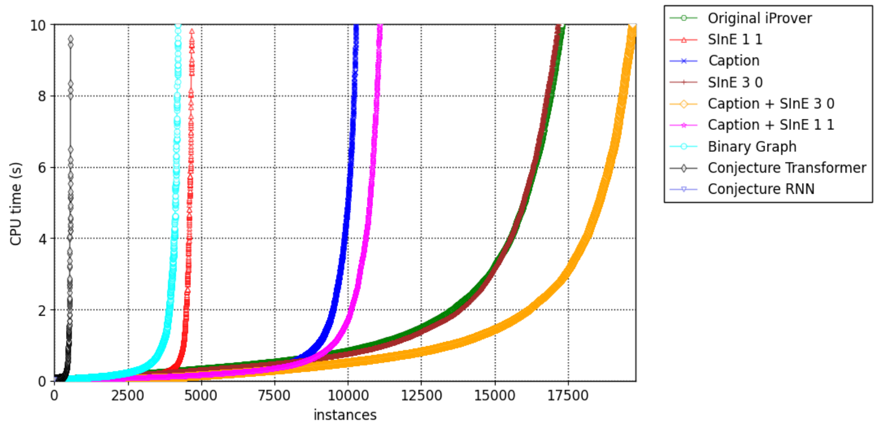

Figure 1: Online evaluation of SInE, Captioning, their combination, and related methods.

SinE, and most interestingly combination of SinE and axiom capturing combines best of both worlds resulting in 17.7% improvement gain over the baseline.

References

[1] Jesse Alama, Tom Heskes, Daniel Kühlwein, Evgeni Tsivtsivadze, and Josef Urban. Premise selection for mathematics by corpus analysis and kernel methods. *J. Autom. Reason.*, 52(2):191–213, 2014.

[2] A. A. Alemi, F. Chollet, G. Irving, C. Szegedy, and J. Urban. Deepmath - deep sequence models for premise selection. *CoRR*, abs/1606.04442, 2016.

[3] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate, 2014.

[4] André Duarte and Konstantin Korovin. Implementing superposition in iProver (system description). In Nicolas Peltier and Viorica Sofronie-Stokkermans, editors, *Automated Reasoning - 10th International Joint Conference, IJCAR, Proceedings, Part II*, volume 12167 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 2020.

[5] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. *CoRR*, abs/1704.01212, 2017.

[6] K. Hoder and A. Voronkov. Sine qua non for large theory reasoning. In N. Bjørner and V. Sofronie-Stokkermans, editors, *Automated Deduction - CADE-23*, volume 6803 of *LNCS*, pages 299–314. Springer, 2011.

[7] Edvard K. Holden and Konstantin Korovin. Graph sequence learning for premise selection. *CoRR*, abs/2303.15642, 2023.

[8] C. Kaliszyk and J. Urban. Mizar 40 for mizar 40. *J. Autom. Reason.*, 55(3):245–256, 2015.

[9] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2016.

[10] K. Korovin. iProver - an instantiation-based theorem prover for first-order logic (system description). In *IJCAR 2008. Proceedings*, pages 292–298, 2008.

[11] L. Kovács and A. Voronkov. First-order theorem proving and Vampire. In N. Sharygina and H. Veith, editors, *Computer Aided Verification CAV. Proceedings*, volume 8044 of *LNCS*, pages 1–35. Springer, 2013.

[12] A. S. Kucik and K. Korovin. Premise selection with neural networks and distributed representation of features. *CoRR*, abs/1807.10268, 2018.

[13] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. *CoRR*, abs/1508.04025, 2015.

[14] J. Meng and L. C. Paulson. Lightweight relevance filtering for machine-generated resolution problems. *J. Appl. Log.*, 7(1):41–57, 2009.

[15] Miroslav Olšák, Cezary Kaliszyk, and Josef Urban. Property invariant embedding for automated reasoning. In Giuseppe De Giacomo, Alejandro Catalá, Bistra Dilkina, Michela Milano, Senén Barro, Alberto Bugarín, and Jérôme Lang, editors, *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 1395–1402. IOS Press, 2020.

[16] Aditya Paliwal, Sarah M. Loos, Markus N. Rabe, Kshitij Bansal, and Christian Szegedy. Graph representations for higher-order logic and theorem proving. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 2967–2974. AAAI Press, 2020.

[17] B. Piotrowski and J. Urban. Stateful premise selection by recurrent neural networks. In E. Albert and L. Kovács, editors, *LPAR 2020*, volume 73 of *EPiC Series in Computing*, pages 409–422. EasyChair, 2020.

[18] K. Prorokovic, M. Wand, and J. Schmidhuber. Improving stateful premise selection with transformers. In F. Kamareddine and C. Sacerdoti Coen, editors, *Intelligent Computer Mathematics*, volume 12833 of *LNCS*, pages 84–89. Springer, 2021.

[19] M. Rawson and G. Reger. Directed graph networks for logical reasoning. In P. Fontaine, K. Korovin, I. S. Kotsireas, P. Rümmer, and S. Tourret, editors, *PAAR7*, volume 2752 of *CEUR Workshop Proceedings*, pages 109–119, 2020.

[20] S. Schulz. System description: E 1.8. In K. L. McMillan, A. Middeldorp, and A. Voronkov, editors, *LPAR-19*, volume 8312 of *LNCS*, pages 735–743. Springer, 2013.

[21] StellarGraph. Unsupervised graph classification/representation learning via distances. *StellarGraph 1.2.1 documentation*, 2022.

[22] M. Tanti, A. Gatt, and K. P. Camilleri. Where to put the image in an image caption generator. *CoRR*, abs/1703.09137, 2017.

[23] J. Urban. MPTP 0.2: Design, implementation, and initial experiments. *J. Autom. Reason.*, 37(1-2):21–43, 2006.

[24] C. Weidenbach, D. Dimova, A. Fietzke, R. Kumar, M. Suda, and P. Wischnewski. SPASS version 3.5. In Renate A. Schmidt, editor, *Automated Deduction - CADE-22. Proceedings*, volume 5663 of *LNCS*, pages 140–145. Springer, 2009.

[25] Peter Wills and Franç ois G. Meyer. Metrics for graph comparison: A practitioner's guide. *PLOS ONE*, 15(2):e0228728, feb 2020.

[26] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention, 2015.