Supplementary Material

1 Hyperparameters tuning

Tthe hyperparameters of SCAR have been tuned to maximise the MAP for explanation regeneration on the WorldTree dev-set. Here, we report the best values for λ, which is used to assign the weights to relevance and explanatory power for computing the explanatory scoring function described in Equation 3:

- $\lambda = 0.89$

Figure 1 shows the MAP score obtained with different values of λ with 0 representing the extreme case in which only the explanatory power is active and 1 the case in which only the relevance is active. As shown in the graph, the explanatory power alone does not allow to achieve high performance on the task, demonstrating that explanations for unseen hypotheses cannot be simply regenerated considering similar hypotheses in the training set and that the relevance model is necessary for generalisation on unseen examples.

2 Dense Encoder

For the implementation of the dense encoder $d(\cdot)$ we adopt Sentence-BERT, whose package can be found at the following URL: https://pypi.org/project/sentence-transformers/. Specifically, we implement the bi-encoder with a *bert-base-uncased* model, adopting a mean-pooling operation to obtain fixed sized sentence embeddings and contrastive loss for training. We release the trained model adopted in the experiments at the following URL: https://drive.google.com/file/d/1iz38q8EIYZdO9U7mAMVz1qUprU8jmEwI/view.

2.1 Training Setup

We train the model using 1 16GB Nvidia Tesla P100 GPU for 3 epochs in total with contrastive loss, while 10% of the training data is used for warm-up. We adopted the following hyperparameters for training:

- batch size = 16
- margin (contrastive loss) = 0.25
- learning_rate = 2e-5
- weight_decay = 0.1
- adam_epsilon = 1e-8
- max_grad_norm = 1.0

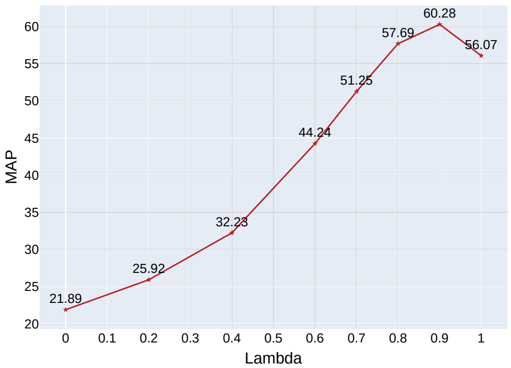

Figure 1: Tuning of λ for the explanatory scoring function $es(\cdot)$ (Equation 3) on the dev-set.

2.2 Faiss Index

For creating the index of dense vectors for the facts bank we use the Faiss package for Python available at the following URL: https://pypi.org/project/faiss-gpu/. Specifically, we adopt IndexIVFFlat.

3 Source Code and Data

The complete code adopted to run our experiments is available at the following URL: https://github.com/ai-systems/hybrid_autoregressive_inference. The WorldTree corpus can be downloaded at the following url: http://cognitiveai.org/dist/worldtree_corpus_textgraphs2019sharedtask_withgraphvis.zip.