TEAM LITTLEANT'S SOLUTION OF TASK2

A PREPRINT

June 16, 2021

1 Background

Task2 is based on dataset WikiKG90M-LSC, which is a Knowledge Graph (KG) extracted from the entire Wikidata knowledge base (Hu et al. [2021]). The dataset consists of 87,143,637 entities and 1,315 relations, represented in form $(head, relation, tail)$. For example, $Hinton \stackrel{\text{citizenof}}{\longrightarrow} Canada$. The goal of Task2 is to automatically impute missing triplets that are not yet present in the current KG. The task can be abstracted as a link prediction problem on KGs, where we define a score function $f(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t})$ on the triplets and optimize the function (we call it Decoder in later sections) to maximize the scores on triplets that exist in the knowledge graph and minimize the scores on triplets that do not exist. To define it formally, our goal is to train a model on a set of positive and negative samples from the knowledge graph, which suffices the following goal:

$$minimize \sum_{\boldsymbol{h},\boldsymbol{r},\boldsymbol{t} \in \mathcal{D}^+ \cup \mathcal{D}^-} -y * \log(\frac{1}{1+\exp\left(-f(\boldsymbol{h},\boldsymbol{r},\boldsymbol{t})\right)}) \tag{1}$$

where $\mathcal{D}^+$ and $\mathcal{D}^-$ are the positive and negative sets of triplets, respectively and y is the label of a triplet, +1 for positive and -1 for negative.

2 Algorithm Architecture

Our architecture is shown in Fig. 1. We used the concatenation of RoBerta's embedding, shallow embedding and some heuristic features as the entities' and relations' embeddings to train our model, optimizing shallow embeddings and model parameters. In the validation phase, we used two Decoders (will introduce them in later section) and learned the weights of them. There are also some other tricks. We trained two models, the first one called the Ensemble Model, which is trained on the whole training set. Another called the Focal Model, which is based on a pre-trained model, and then trained on the samples the Ensemble Model performs badly on. We tune the combining strategy of the two models on validation set. In the end, we apply the same strategy we use for validation phase on the test dataset. Finally, our method achieves MRR=0.951 on test dataset.

3 Training Phase Optimization

In this section, we introduce the two optimization strategies in training phase. We mainly include two parts: 1) feature engineering towards the triplets and 2) decoder selection.

3.1 Feature Engineering

There are two types of heuristic features: node side and relation side. We extract those features mainly based on the statistics of the entities and relations. Those features are concatenated with other embeddings (RoBerta, Shallow), acting as part of the node&relation feature vector.

Main node features:

- Nodes' in-degree.

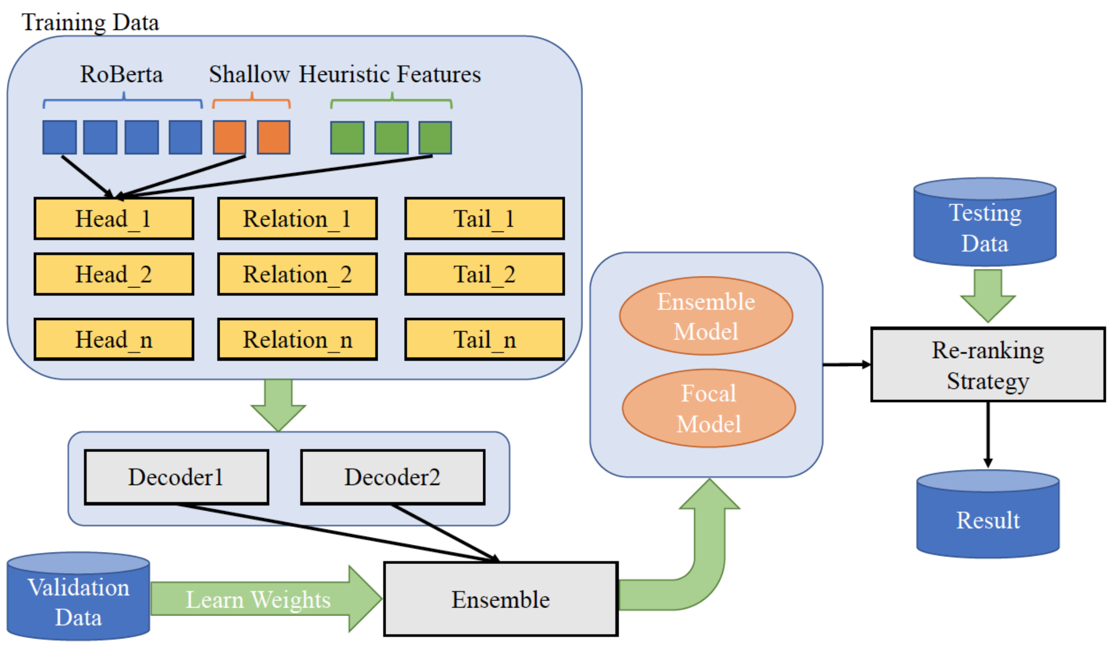

Figure 1: Overview of Architecture.

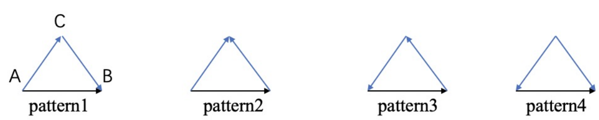

Figure 2: Four different types of patterns. In our work, we count the occurrence of a nodes' as node C in picture.

- Nodes' out-degree.
- Nodes' occurrences on four different patterns (described in Fig. 2).

Main relation features:

- Counts of different co-occur heads.
- Average of heads' occurrence the relation connects.
- Average of tails' occurrence the relation connects.
- Counts of co-occur tails.

3.2 Decoder Selection

We chose PairRE (Chao et al. [2020]) and RotateE (Sun et al. [2019]), which have the best performance on validation set. The decoders' scoring functions are shown in Tab. 1.

Table 1: Decoder Selection

Name	Description
PairRE	$-\|\boldsymbol{h} \circ \boldsymbol{r}^H - \boldsymbol{t} \circ \boldsymbol{r}^T\|_1$
RotateE	$-\|\boldsymbol{h} \circ \boldsymbol{r} - \boldsymbol{t}\|_2$

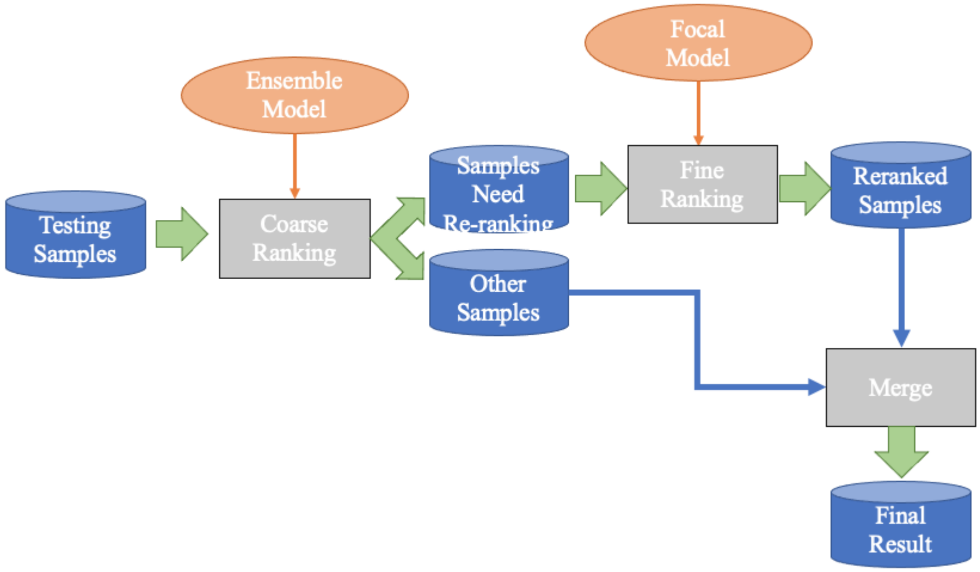

Figure 3: Re-ranking strategy.

4 Validation Phase Optimization

In order to fully utilize the data in validation, we designed an ensemble module, in which we learn the weight of the two aforementioned decoders. To describe it formally, let $f_{PairRE}(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t})$ be the the prediction of PairRE given the triplet, and let $f_{RotatE}(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t})$ be the the prediction of RotatE. The final score is calculated as shown in Eq. 2, where w is the parameter to be learned.

$$score = g_{w_1}(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}) * f_{PairRE}(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}) + g_{w_2}(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}) * f_{RotatE}(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}) \tag{2}$$

Where g_{w_1} and g_{w_2} are two weight learning functions, outputing two weights $w_1, w_2 \in \mathcal{R}$. Note that we do not limit the scale of w_1 and w_2, considering that the scores given by the two Decoders are at different scales.

The ensemble procedure is listed as follows:

1. Train MLP on validation set and learn the parameters g_{w_1} and g_{w_2}.
2. Fix parameter functions g_{w_1} and g_{w_2}.
3. Merge train set and validation set, and retrain the model.

The resulting model is called the Ensemble Model.

5 Fine-tuning

In this section, we describe how we further improve the performance of the Ensemble Model.

Data Sampling for Focal Model We analyze the validation samples where the Ensemble Model performs badly on. We found that among the Ensemble Model's ranking on validation samples, when the top 1s' confidence score is lower than a certain threshold, the MRR is often low in the same time. Henceforth, we up-sampled the cases with low confidence scores on the training set. We then train another model based on a pre-trained model using PairRE as decoder, focusing on the cases the Ensemble Model performs badly on. This model is called the Focal Model.

6 Re-ranking Using the Focal Model

Although the Focal Model performs better than the Ensemble Model on the aforementioned groups, we found that the Ensemble Model can recall more correct answers in the top 10 candidates. We then design the re-ranking strategy described as follows:

1. Use the Ensemble Model to make prediction on test dataset.
2. Find the samples where the confidence scores are below the threshold.
3. Re-rank the chosen samples using the Focal Model.

The re-ranking strategy is shown in Fig. 3.

References

Weihua Hu, Matthias Fey, Hongyu Ren, Maho Nakata, Yuxiao Dong, and Jure Leskovec. Ogb-lsc: A large-scale challenge for machine learning on graphs. *arXiv preprint arXiv:2103.09430*, 2021.

Linlin Chao, Jianshan He, Taifeng Wang, and Wei Chu. Pairre: Knowledge graph embeddings via paired relation vectors. *arXiv preprint arXiv:2011.03798*, 2020.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*, 2019.