source side of a small corpus as queries to extract a domain specific training set. In this case, a sentence pair in the training data may occur in several sub training data, but this doesn't matter. The general model is used when the online input is not similar to any prepared submodels. We can use all available training data to train the general model since generally larger data can get better model even there are some noises.

3.2 Online model weighting

We also use TF-IDF information retrieval method for online model weighting. The procedure is as follows:

For each input sentence:

1. *Do IR on training data collection, using the input sentence as query.*
2. *Determine the weights of submodels according to the retrieved sentences.*
3. *Use the optimized model to translate the sentence.*

The information retrieval process is the same as the offline data selection except that each retrieved sentence is attached with the sub-corpus information, i.e. it belongs to which sub-models in the training process.

With the sub-corpus information, we can calculate the weights of submodels. We get the top N most similar sentences, and then calculate proportions of each submodel's sentences. The proportion can be calculated use the count of the sentences or the similarity score of the sentences. The weight of each submodel can be determined according to the proportions.

Our optimized model is the log linear interpolation of the sub-models as follows:

$$\hat{p}(e \mid c) = p_0(e \mid c)^{\delta_0} \times \prod_{i=1}^{M} p_i(e \mid c)^{\delta_i}$$

$$\hat{e} = \arg\max_{e}(\delta_0 \log(p_0(e \mid c)) + \sum_{i=1}^{M} \delta_i \log(p_i(e \mid c)))$$

where, p_0 is the probability of general model, p_i is the probability of submodel *i*. δ_0 is the weight of general model. δ_i is the weight of submodel *i*. Each model *i* is also implemented using log linear model in our SMT system. So after the log operation, the submodels are interpolated linearly.

In our experiments, the interpolation factor δ_i is determined using the following four simple weighting schemes:

Weighting scheme 1:

$$\delta_0 = 0; \quad \delta_{max\_model} = 1; \quad \delta_{i \neq max\_model} = 0;$$

Weighting scheme 2:

```
if Proportion(max_model) > 0.5
    Use weighting scheme1;
else
    $\delta_0 = 1$;  $\delta_i = 0$;
```

Weighting scheme 3:

$$\delta_0 = 0;$$

$$\delta_i = \text{Proportion } (model_i);$$

Weighting scheme 4:

```
if Proportion(max_model) > 0.5
    Use weighting scheme3;
else
    $\delta_0 = 0.5$;
    $\delta_i = 0.5 \times \text{Proportion } (model_i)$;
```

where, *model*$_i$ is the *i*-th submodel, $i = (1 \ldots M)$. Proportion (*model*$_i$) is the proportion of *model*$_i$ in the retrieved results. We use count for proportion calculation. *max_model* is the submodel with the max proportion score.

The training and translation procedure of online model optimization is illustrated in Figure 2.

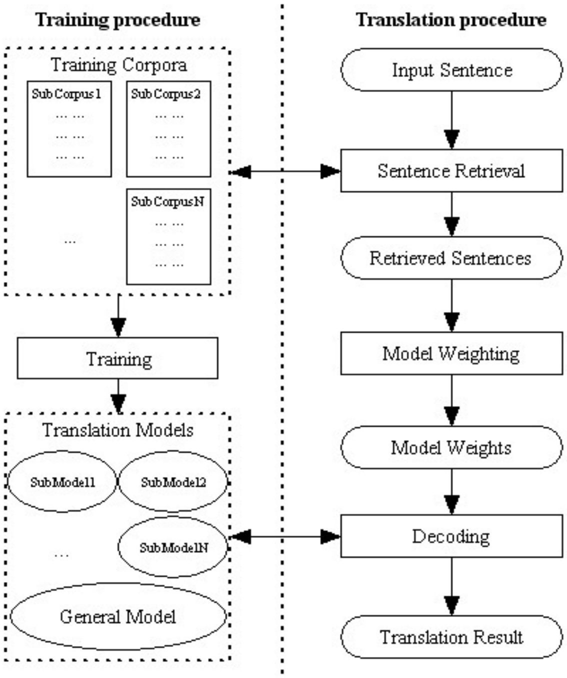

Figure 2. Online model optimization