科大讯飞·领域迁移机器翻译挑战赛

赛事任务

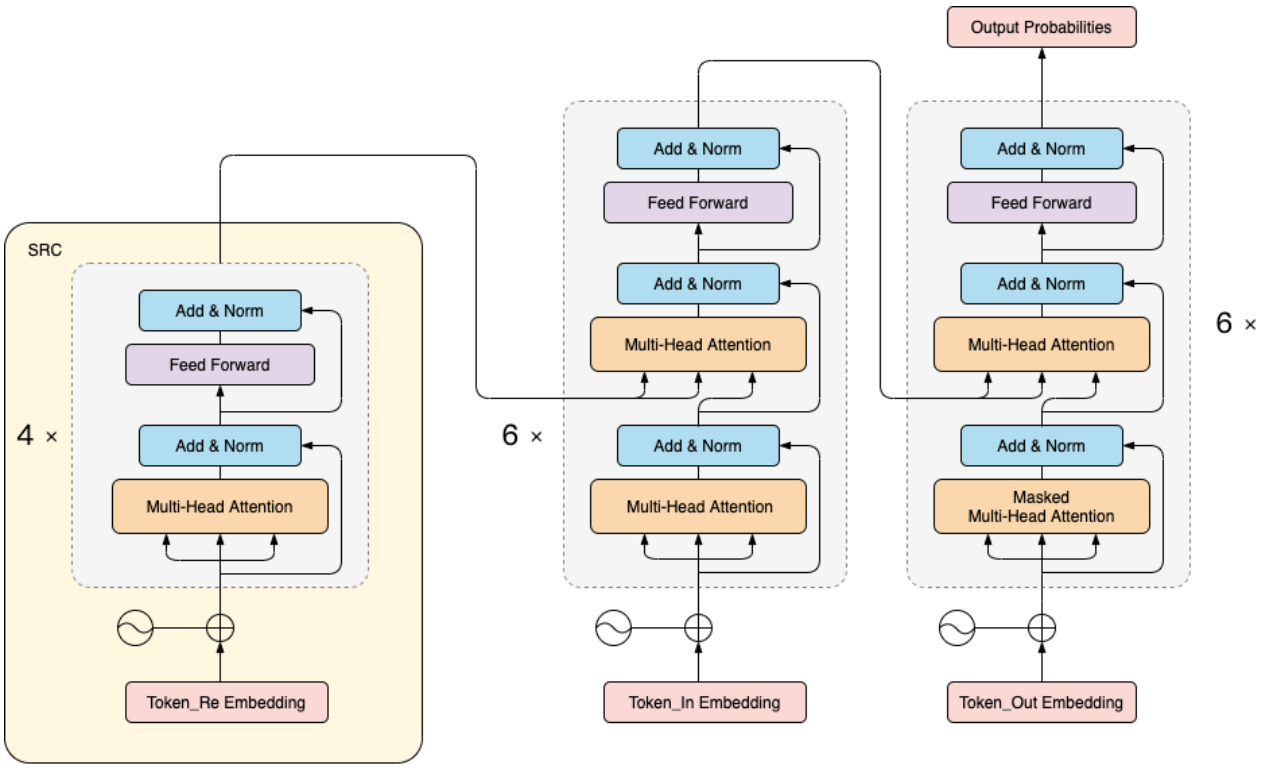
领域迁移机器翻译挑战赛旨在增强跨领域机器翻译技术，本次大赛提供少量新闻领域的中英平行句对和大量口语领域的中文数据作为训练样本，参赛队伍需要基于提供的训练样本进行中到英机器翻译模型的构建与训练，并基于测试集提供最终的翻译结果，数据包括：

- 训练集：
  - 双语数据：200万新闻领域中英双语句对
  - 单语数据：1000万口语领域中文数据
- 开发集：1000条口语领域双语句对
- 测试集：1000条口语领域双语句对

思路说明

基于少量新闻领域的中英平行句对和大量口语领域的中文数据，进行中到英机器翻译模型的构建，并实现目标端新闻到口语的领域迁移。训练并使用跨语言对齐的句子检索模型，在中文单语数据集检索得到英文句对应相似句，作为"源端翻译记忆"。通过独立的记忆编码器，以额外注意力层的形式，融入Transformer模型，实现机器翻译和领域迁移的目标。

模型结构



实验设置

环境设置

```
$ pip install -r requirements.txt
$ export MTPATH=<data_path>
```

数据预处理

分词：使用SentencePiece 从raw text训练分词模型，来作为text tokenizer和detokenizer，并使用subword (byte-pair-encoding (BPE) [Sennrich et al.]), 对所有实验数据进行分词处理，并生成词表。

```
$ spm_train --input=<input> --model_prefix=<model_name> --vocab_size=32000 --character_coverage=1.0 --model_type=bpe
# 或者使用
$ sh scripts/data_utils/tokenizer.sh
```

1. 对于字符集相对丰富的中文语料，我们将字符覆盖率`--character_coverage`设置为`0.9995`。
2. 中英双语，词表大小均设置为`32000`
3. 分词模型的元标记字符为 "__" (U+2581)

```
$ sh scripts/data_utils/data_prepare.sh
```

数据清理：将分词处理后的数据，(针对训练集数据)进一步修整，并重新统计词表。

1. 句子的`--max_len` 设置为 250
2. 源端与目标端的句子长度比`--ratio`，设置为2.0

# Sent	**1,970,438**
# zh.vocab	**36502**
# en.vocab	**31771**

Sample:

```
# train.txt
_比如 , 娱乐 的需求 一直 存在 , 但 娱乐 的方式 方法 却 随着 时间 而 改变 。 _The _need _to _be _entertained _exist _all _these _while _but _the _way _to _be _entertained _changes _over _time _.

# dev.txt
_我 的美 容 师 很 棒 , 你 下次 可以 跟 我 一起 去 。 _I _trust _my _beaut ician _a _lot _so _why _not _go _with _me _next _time .
```

Baseline-Transformer

基于新闻领域的中英平行句对的Transformer模型。

```
$ sh scripts/vanilla/train.sh
$ sh scripts/vanilla/work.sh
```

使用源端翻译记忆的Transformer模型

跨语言对齐的检索模型的训练

- 使用一个双塔结构（dual-encoder framework）的检索模型将源端（zh）句子和目标端（en）句子在向量空间对齐。（训练数据为新闻领域200w平行句子对）
- 在训练过程中，使用**英语端的新闻句子**作为query，在大规模的单语语料库中（1000w的中文口语数据集），检索出，（在向量空间）距离最近的k条句子，作为"**口语翻译记忆库**"，以期实现口语特征的抽取。
- **模型细节**

  训练可分为两个跨语言对齐任务，先用两个transformer encoder对句子对的句子分别编码得到$X = E_{\text{src}}(x)$和$Z = E_{\text{tgt}}(z)$, 采样B个句子对，第i个源句和第j个目标句组成句子对(X_i, Z_j)

  - sentence-level 跨语言对齐, ，计算内积矩阵$S = XZ^T$，作为得分矩阵，元素S_{ij}为对应得分，当$i = j$时，句子对(X_i, Z_j)是对齐的句子对，最大化对角线上的值，最小化其余的值，得到目标函数

$$\mathcal{L}_{\text{snt}}^{(i)} = \frac{-\exp(S_{ii})}{\exp(S_{ii}) + \sum_{j \neq i} \exp(S_{ij})}$$

  - token-level 跨语言对齐，给定源端的一个句子表示，用词袋模型，预测目标端的token

$$\mathcal{L}_{\text{tok}}^{(i)} = -\sum_{w_y \in \mathcal{Y}_i} \log p\left(w_y \mid X_i\right) + \sum_{w_x \in \mathcal{X}_i} \log p\left(w_x \mid Y_i\right)$$

  $\mathcal{X}_i$ $(\mathcal{Y}_i)$为源端（目标端）第i句的token集合，

  综上，联合损失函数为 $\frac{1}{B}\sum_{i=1}^{B} \mathcal{L}_{\text{snt}}^{(i)} + \mathcal{L}_{\text{tok}}^{(i)}$

```
$ sh scripts/pretrain.sh
```

> 检索模型的设计与训练，参考自ACL 2021 Best Performance Paper [Neural Machine Translation with Monolingual Translation Memory]

通过Faiss建立索引，检索得到“口语翻译记忆库”

```
$ sh scripts/zh-en/build_index.sh
$ sh scripts/zh-en/search_index.sh
```

> FAISS index code `IVF1024 HNSW32,SQ8`

翻译模型的训练

本模型，使用一个独立的4层Transformer的Encoder结构，来实现对“口语翻译记忆库”中句子的编码，翻译模型部分沿用Transformer模型的基本架构，在Encoder部分，特别地，添加额外的注意力层，用以融合“口语翻译记忆库”中的句子特征。

```
$ sh scripts/zh-en/train.sh
```

Postprocess以及评估

```
$ sh scripts/zh-en/work.zhen.sh
```

在测试阶段，我们使用Baseline模型所产出的测试集英文翻译作为query，来实现测试集的口语库检索。并通过来自验证集的口语句子对的“口语特征规则”，对翻译结果进行Postprocess，以强化口语特征。迭代执行5次上述操作。得到最终的翻译结果。