双语语料过滤任务数据文件格式说明

本文档对该评测任务中的相关数据文件及格式进行说明，文件包括评测组织方发放的数据文件以及参评单位需要提交的结果文件。所有文件均要求为 UTF-8 编码。

1. 评测组织方发放的数据格式说明

评测组织方发放的数据有三种：包含噪声的平行语料库、机器翻译开发集和测试集。

同时，评测组织方提供机器翻译模型训练脚本：包含基于 ID 排序的平行语料筛选脚本和神经网络机器翻译模型训练脚本(Marian)。

1.1 平行语料库

待过滤的平行语料库由两个文件组成，包括逐行对应的源语言文件 train.zh 和目标语言文件 train.en，两文件均为 UTF-8 编码格式。每行文本包含句子和对应的唯一 ID，ID 与句子之间由 TAB('\t')分隔。源语言文件和目标语言文件根据 ID 对齐。文本数据无任何分词、标点、大小写等预处理，为原始文件。

图 1、图 2 示例说明了“双语语料过滤”项目中源语言文件、目标语言文件的格式。

train.zh
<#A00000050#>　术后椎体高度恢复明显，随访发现患椎未出现塌陷。
<#A00000051#>　我很愿意同您再次会晤，讨论目前情形

图 1 平行语料的源语言文件

train.en
<#A00000050#>　Postoperative vertebral height restoration of obvious follow-up was found suffering from vertebral collapse phenomenon does not appear.
<#A00000051#>　I will be pleased to meet with you once more to discuss the situation

图 2 平行语料的目标语言文件

1.2 机器翻译开发集

同 WMT 文件格式，句子前无 ID 标识

1.3 机器翻译测试集

在评测阶段组织方将只发放 test.zh 和 test.en 两个文件，其格式与训练集相比，省去了标点符号。

1.4 平行语料筛选脚本

该任务要求参评单位提交一份平行语料句对 ID 的排序文件，其中将待筛选平行句对的 ID 按某种筛选方法由高到低排序，每行一个 ID。排序越高意味着质量越好。当给定一个 ID 排序文件，筛选脚本 subsample.sh 可分别将前 5 亿个词以及前 1 亿个词对应的句对筛选出来（词数统计均以英文端分词后为标准，分词工具已集成在脚本中）。请注意：如果根据参评单位提供的 ID 排序所筛选出的词数不满足上述量级，将以 ID 排序对应的全部平行句对为准，训练机器翻译模型。

脚本语法示例：`subsample.sh FILE_RANK FILE_ZH FILE_EN OUT`

（其中：FILE_RANK 为 ID 排序文件；FILE_ZH 为原始双语中文端文件；FILE_EN 为原始双语英文端文件；OUT 为输出筛选后双语数据文件的名称。）

通常用法类似：subsample.sh my-rank-file.txt train.zh train.en out

脚本会在当前路径下生成名称为 data_train 的路径，并在该路径下输出如下文件：out_500m.zh out_500m.en out_100m.zh out_100m.en，分别对应筛选出 5 亿词和 1 亿词（按英文端分词统计）的源语言、目标语言文件。

将生成的训练数据文件，配置在机器翻译模型训练脚本的实验配置文件中，见下文。

1.5 神经网络机器翻译模型训练脚本

一、依赖环境和工具：

1. linux 平台：推荐 ubuntu 16.04 或 18.04
2. python 3.5 或以上版本
3. python 包：jieba、sacrebleu。安装方式：pip install jieba sacrebleu
4. g++ 4.8 或以上
5. marian 工具集，marian 编译安装参考： https://marian-nmt.github.io/quickstart/

二、脚本说明：

1. s0-check_env.sh：检查运行环境是否符合要求
2. s1-preprocess.sh：数据预处理
3. s2-nmt_train.sh：模型训练
4. s3-nmt_test.sh：测试模型效果
5. settings.sh：整个实验的配置文件

三、原型实验：

0. 安装好依赖的工具和包，然后运行 sh s0-check_env.sh，如果显示‘success’表示安装成功（这一步必须做一次，并且只需要做一次）；

1. 配置 settings.sh

GPU='0 1' # 指定所用的 GPU id ，可以指定一个或多个。

work_dir=./exp_ccmt # 指定当前这个实验的目录，所有预处理完毕的数据、训练获得的模型、最终的 bleu 值都会保存在该路径下。不同的实验可以设置不同的目录。

src_lang=zh # 源语言

tgt_lang=en # 目标语言

train_set=/home/xxx/data_train/train # 训练集数据文件的名称，实际存储要求有 /home/xxx/data_train/train.zh 和 /home/xxx/data_train/train.en 这两个文件，且两个文件行数一致，对应行是互为翻译的句对，不需要分词。

dev_set=/home/xxx/data_test/wmt18 # 开发集数据，格式说明同上。

test_set=/home/xxx/data_test/wmt19 # 测试集数据，格式说明同上。

src_max_len=150 # bpe 后的最长原文句长，超过的训练集句对会被舍弃。

tgt_max_len=150 # bpe 后的最长译文文句长，超过的训练集句对会被舍弃。

src_bpe_size=50000 # 原文 bpe size，一般不用改

tgt_bpe_size=50000 # 译文 bpe size，一般不用改

marian=/home/xxx/marian/build/ # marian 工具集的路径，需要配置绝对路径。该路径下必须要有 marian-vocab、 marian 、 amun 这几个可执行文件。

2. 运行 sh s1-preprocess.sh 进行数据预处理

3. 运行 sh s2-nmt_train.sh 进行模型训练，开发集的 bleu 会记录在：$work_dir/model/bleu_scores

4. 模型收敛后，运行 sh s3-nmt_test.sh 模型测试，测试集的 bleu 保存在：$work_dir/test.bleu

2. 参评单位需要提交的数据格式说明

参评单位可以选择提交至多两个不同的 ID 排序文件。参评单位仅需要提供最终的排序文件及系统描述信息文件。其格式说明如下：

2.1 文件命名

所有需要提交的文件的命名方式请遵循下表要求：

（其中：项目代号以 corpus-filtering 为例子，参评单位代号以 ict 为例子）

文件	文件名模式	文件名举例
双语语料过滤 ID 排序结果 1 (默认为该单位主提交结果)	项目代号-参评单位代号-id-rank.1	corpus-filtering-ict-id-rank.1
双语语料过滤 ID 排序结果 2 (默认为该单位对比结果)	项目代号-参评单位代号-id-rank.2	corpus-filtering-ict-id-rank.2
系统描述信息	项目代号-参评单位代号-sys.txt	corpus-filtering-ict-sys.txt

2.2 最终提交结果文件

待过滤的双语语料句对 ID 排序文件以如下形式提交：句对 ID 按某种筛选方法由高到低排序，每行一个 ID。排序越高意味着质量越好。句对 ID 必须为原始语料句对 ID 的子集。

举例：

<#A00000000#>

<#A00000001#>

<#A00000002#>

<#A00000003#>

……

在参评系统的描述信息文件中，需要对一下内容给出说明：

- 软硬件环境：包括操作系统及其版本、所用开发环境和开发软件等等；
- 运行时间：参评系统从接受输入到产生全部输出所花费的时间；
- 技术概要：简要说明参评系统所采用的主要技术和重要参数；
- 训练数据说明：说明参评系统所使用的训练数据和开发数据，是否使用额外的数据资源，也需要进行说明。

外部技术说明： 说明除了参评单位自己的技术外，还采用了哪些外部技术，包括各种开源代码、自由软件、共享软件或商业软件。