英语学习者作文质量自动评测系统 AES 使用手册

刘磊

2019.08

系统配置

操作系统：Ubuntu 18.04.1

系统内存：16 G

处理器：Intel(R) Core(TM) i5-3210M CPU @ 2.50GHz

编程语言：Perl; R; JAVA

一、数据预处理

本研究使用公开数据集 FCE 英语学习者语料库训练和测试系统性能 (Yannakoudakis *et al*. 2011)。该语料库由剑桥 FCE 考试作文构成，包含作文 1244 篇，共 95 万词。可从网站 *https://ilexir.co.uk/datasets/index.html* 免费下载本研究使用的训练和测试数据。

FCE 原始数据存放于 *data/xml* 文件夹，包含训练作文 1141 篇 (*data/xml/train*)和测试作文 97 篇 (*data/xml/test*)。为了便于对比系统性能，本研究使用的训练和测试数据与 Yannakoudakis *et al*.(2011) 、 Yannakoudakis & Briscoe(2012) 、 Zhang *et al*.(2018) 和 Vajjala(2018)的研究一致。FCE 采用 XML 格式存储数据，人工标注了英语学习者作文中的 75 类语法错误。为了便于自动提取作文中的词汇、语法和语篇特征，本研究需要将 XML 文本转换为纯文本文件，具体方法如下：

1. 提取元信息

元信息指学习者国籍、性别、年龄和作文主题、分数等与学习者作文相关的变量。本研究通过下列程序提取 FCE 语料库的元信息文件：

```
# 进入程序所在文件夹，如 /home/jason/aes
cd /your/path/to/app
# 提取 FCE 语料库元信息
perl code/utils/get_fce_meta.pl --data train
perl code/utils/get_fce_meta.pl --data test
```

通过上述程序获取的 FCE 语料库元信息保存于 *data/meta/train* 和 *data/meta/test*。初步分析元信息文件后，得到图 1 所示 FCE 训练集和测试集的作文分数分布。

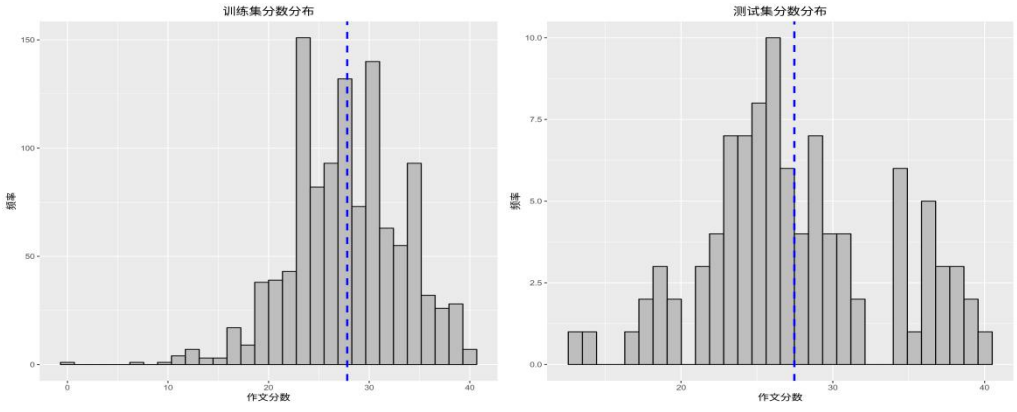

图 1. FCE 训练集和测试集分数分布

FCE 语料的作文主题保存于 *data/prompts*，如图 2 所示，训练和测试集中的数据选自不同年份的 FCE 考试作文，写作主题并不重合。

为了便于后期提取语言特征，本研究在训练语料的元信息中添加了 3 列数据：*score_mean*, *score_median* 和 *fold*。如图 3 所示，元信息共包含 9 列：前 6 列分别是作文编号、学习者国籍、年龄和作文数量、字数、分数；第 7 和第 8 列对应作文的平均分和中

训练集作文主题 0102_2000_6_4:

Your class has had a discussion about how science and technology affects our lives. Your teacher has now asked you to write a composition answering the following question: How has modern technology changed your daily life? Write your composition.

测试集作文主题 0100_2001_6_2:

A group of American students has just arrived in your town and the group leader has asked for information on an interesting building to visit. Write a report for the group leader describing one building and giving reasons for your recommendation. Write your report.

图 2. FCE 训练集和测试集作文主题示例

位分，如小于均值则标记为 0，大于均值标记为 1；第 9 列是用于 10 折交叉验证的数据组别。

id	***lang***	***age***	***texts***	***words***	***score***	***score_mean***	***score_median***	***fold***
doc1787	*Russian*	*16-20*	*2*	*658*	*27*	*0*	*0*	*1*
doc2093	*Korean*	*16-20*	*2*	*1146*	*22*	*0*	*0*	*5*
doc741	*Greek*	*26-30*	*2*	*816*	*26*	*0*	*0*	*2*
doc330	*Polish*	*21-25*	*2*	*1064*	*27*	*0*	*0*	*9*
doc279	*French*	*16-20*	*2*	*752*	*32*	*1*	*1*	*8*
.....								

图 3. FCE 训练语料元信息示例

获取上述元信息文件的方法如下：

```
# 进入程序所在文件夹，如 /home/jason/aes
cd /your/path/to/app
# 运行 R 软件
sudo R
# 运行 R 脚本，转换 FCE 语料库元信息
source("code/utils/transform_fce_meta.r")
```

2. 提取作文

FCE 语料库采用 XML 文件存储作文，如图 4 所示。其中，XML 元素<*p*>表示作文段落，<*NS*>表示语法错误类型，<*i*>是包含语法错误的原文本，<*c*>是修改语法错误后的正确表述。如第 2 段中的<*NS type="RJ"*><*i*>*exciting*</*i*><*c*>*excited*</*c*></*NS*>表示该句存在形容词类错误 *RJ*，应该将 *exciting* 替换为 *excited*。

```
<p>Dear Mr Ryan<NS type="RP"><i>.</i><c>,</c></NS></p>
<p>Thanks for <NS type="DD"><i>you</i><c>your</c></NS> letter. I am so <NS type="RJ">
<i>exciting</i><c>excited</c></NS> that I have won the first prize.
```

图 4. FCE XML 格式示例

FCE 语料库中的语法错误均为人工标注，有助于研究英语学习者的二语写作能力与语法错误之间的关系。但是，本研究的目的是自动评估学习者作文质量，不借助人工标注的数据训练模型，因此需要将图 4 所示的 XML 文本转换为纯文本格式。转换方法如下：

```
# 进入程序所在文件夹，如 /home/jason/aes
cd /your/path/to/app
# xml 转换为纯文本
perl code/utils/get_fce_txt.pl --data train
perl code/utils/get_fce_txt.pl --data test
```

转换后的作文以 TXT 格式存储于 *data/txt/train* 和 *data/txt/test*。

3. 标注数据

本研究使用斯坦福自然语言处理工具 Stanford CoreNLP(Manning *et al*. 2014)自动标注构建评分模型所需的语言学信息。标注方法如下：

```
# 进入程序所在文件夹，如 /home/jason/aes
cd /your/path/to/app
# xml 转换为纯文本
```

```
perl code/utils/annotate_fce.pl --data train
perl code/utils/annotate_fce.pl --data test
```

标注后的文本格式如图 5 所示。其中，*text*、*token* 表示原始文本和分词后的文本；*lemma*、*tag* 和 *parse* 分别对应作文各句的词元、词性和句法结构。标注后的文本保存于 *data/anno/train* 和 *data/anno/test*。

```
{
"text":
"I am so exciting that I have won the first prize.",
"token":
["I","am","so","exciting","that","I","have","won","the","first","prize","."],
"lemma":
["I","be","so","exciting","that","I","have","win","the","first","prize","."],
"tag":
["PRP","VBP","RB","JJ","IN","PRP","VBP","VBN","DT","JJ","NN","."]
"parse":
"(ROOT (S (NP (PRP I)) (VP (VBP am) (ADJP (RB so) (JJ exciting) (SBAR (IN that) (S (NP (PRP I)) (VP (VBP have) (VP (VBN won) (NP (DT the) (JJ first) (NN prize)))))))) (. .)))",
},
```

图 5. 数据标注示例

二、特征提取和筛选

Farag *et al*. (2017)的研究表明，采用卷积神经网络(CNN)训练的评分模型性能低于支持向量机等传统的机器学习算法。因此，本研究采用人工提取的语言学特征训练作文自动评分模型，未考虑近期兴起的深度学习算法。

1. 词袋特征

词袋特征是构建学习者英语作文评分模型的常用特征(Yannakoudakis *et al*. 2011; Islam & Hoque 2012; Mayfield & Rosé 2013)。本研究的词袋模型构建过程如下：

(1) 特征提取

选取包含词形和词性的 1-3 元 *N* 元序列训练词袋模型。提取 *N* 元序列的方法如下：

```
# 进入程序所在文件夹，如 /home/jason/aes
cd /your/path/to/app
# 提取 N 元序列
perl code/utils/txt2bow.pl --data train --nopunct --lc --t 3 --p 3
perl code/utils/txt2bow.pl --data test --nopunct --lc --t 3 --p 3
```

转换后的特征文件保存于 *data//bow/train* 和 *data/bow/test*。然后运行下列程序构建词袋特征列表：

```
# 构建特征列表
perl code/bin/extract_bow_feats.pl --fold 0
```

特征列表文件位于 *feats/bow_table*，格式如下：

Type	***Term***	***Length***	***Freq***	***Cover***	***IDF***	***MI***
t	*going to*	*2*	*477*	*333*	*0.53*	*14.47*
t	*that means*	*2*	*10*	*9*	*2.10*	*9.20*
p	*nn in nnp*	*3*	*875*	*567*	*0.30*	*10.51*
p	*rb rb vbd*	*3*	*23*	*23*	*1.70*	*12.62*
.....						

图 6. 词袋特征列表

其中 *type* 为特征类型，*p* 代表词性，*t* 代表词形；*term* 为具体特征，如 *nn* 表示名词，*in* 表示介词等；*length*、*freq*、*cover* 和 *IDF* 分别代表特征长度、频率、覆盖率和逆文档频率；*MI* 是互信息值，计算方法详见论文 3.1 节。

(2) 特征筛选

根据 N 元序列长度、互信息值，采用 LibLinear 工具包中的 SVR 算法筛选词袋特征，具体方法如下：

```
# 进入程序所在文件夹，如 /home/jason/aes
cd /your/path/to/app
# 筛选词袋特征
perl code/bin/select_bow_feat.pl
```

筛选后的词袋特征位于 *feats/selection/bow*。

2. 语言学特征

语言学特征包括文本表层特征、词汇多样性、文本可读性、句法复杂度、语法正确性和语篇连贯度等 6 个维度，详见论文 3.2 节。

(1) 特征提取

语言学特征提取方法如下：

```
# 进入程序所在文件夹，如 /home/jason/aes
cd /your/path/to/app
# 筛选词袋特征
perl extract_lingua_feats.pl --data [train|test] --thresh 0.25
```

其中，参数 *--data* 指定训练或测试数据，*--thresh* 为基于 word2vec 的词汇语义相似度的阈值。本研究将阈值设定为 0.25。提取后的语言学特征以 TSV 格式保存于 *feats/train/lingua* 和 *feats/test/lingua*。

(2) 特征筛选

语言学特征筛选方法如下：

```
# 进入程序所在文件夹，如 /home/jason/aes
cd /your/path/to/app
# 运行 R 软件
sudo R
# 运行 R 脚本，筛选语言学特征
source("code/bin/select_lingua_feats.r")
```

筛选后的语言学征位于 *feats/selection/lingua*。

三、评分模型构建和评测

1. 模型 I (baseline model)

模型 I 参照现有研究，直接合并词袋特征和语言学特征，使用 SVR 构建评分模型，方法如下：

```
# 进入程序所在文件夹，如 /home/jason/aes
cd /your/path/to/app
# 构建模型 I
perl baseline_model.pl --t 1 --p 3 --mi 16 --weight binary --norm 1
```

其中，参数 *--t* 和 *--p* 分别为单词和词性序列长度，*--mi* 为互信息阈值，*--weight* 为加权方式，*--norm* 为数据标准化开关。模型评测结果保存于 *code/temp/svm_measure*。

2. 模型 II (stacking model)

模型 II 采用集成学习构建评分模型，方法如下：

```
# 进入程序所在文件夹，如 /home/jason/aes
cd /your/path/to/app
```

```
# 构建模型 II
# 运行 10 折验证，将稀疏词袋特征转换为非稀疏特征
 perl stack_model.pl --t 1 --p 3 --mi 16 --weight binary
# 运行 Random Forests，评测系统性能
sudo R
source("code/bin/stack_model.r")
```