

華中科技大學

課程報告

題目： 基于英语自动评分
作文的部分原理探究

院（系）： 光学与电子信息
课程名称： 信息检索（公选课）
学 号： U201813654
姓 名： 马笑天
任课老师： 魏 巍
报告日期： 2020 年 5 月 5 日

目录

1 绪论	3
2 对于英语自动评分作文的部分原理探究	4
2.1 自然语言处理技术	4
2.2 聚类技术	4
2.2.1 整体思想	4
2.2.2 实现步骤	5
2.3 多项式朴素贝叶斯技术	5
3 总结与展望	6
3.1 总结	6
3.2 展望	6
4 致谢	7
参考文献	8
附录	8

1 绪论

随着计算机科学，特别是人工智能在近十几年的迅猛发展，英语教学模式正在悄然发生改变。英语作文由于其体现作者写作情感、需要考察行文的语法正确性、语义连贯性、上下文点题等综合因素，一直需要老师进行手动批阅。自动作文评分(Automated Essay Scoring, AES)近年来已逐渐成为自然语言处理的热点问题^[1]。

当前自动作文评分中的内容研究，根据对作文中语言和内容评价的侧重，可以划分为3类：侧重语言形式的自动评分，如PEG；侧重语义，或者说内容的自动评分，如IEA；二者兼顾的自动评分，如E-rater。其他研究基本上也可以划归为其中的一类，如IntelliMetric属于二者兼顾；基于文本分类技术的自动评分系统BETSY虽然可以说可以兼顾包括语言使用的各方面特征，但目前研究的分类方法都是基于词汇，而且主要是实义词汇，所以应该划归第二类，即偏重内容的自动评分。真正能够而且目前正用于外语作文自动评分的只有语言质量和内容兼顾的AES系统，如E-rater和IntelliMetric^[1]。

如今，在中国的高等院校中，越来越多的老师选择使用例如批改网、句酷作文等作文进行作文批阅，笔者就曾在大学期间使用过其中的批改网，惊叹于其功能之丰富：5秒出分，显示文章的综合分数、用词分数、流畅度分数、词汇丰富程度分析等；语病检测、好词好句推荐等。这些平时被学生看似非常“玄乎”的概念竟然能被量化为百分制，使笔者不禁好奇起来。借此《信息检索》课程总结的机会，笔者以一些文献中的英语批改软件的设计为例，将从多方面来简略分析这类软件的算法，并提出一些关于这类软件未来的总结与展望。



About the *e-rater*[®]
Scoring Engine

2 对于英语自动评分作文的部分原理探究

由于构建一个完整的英语作文自动批改软件涉猎的范围较广，涉及到服务器开发技术、前端开发技术、数据库技术等多方面，由于笔者能力有限，以及本课程的重点，所以下面着重介绍有关信息检索和自然语言处理相关的细节。

2.1 自然语言处理技术

在自动批阅系统还有推荐系统中都用到了自然语言处理技术。自然语言处理(Natural language processing, NLP)关注计算机与人类语言之间相互作用的领域。在陈的论文中，他设计研究的个性化推荐算法以 Python 语言为框架，并用到自然语言工具包(Natural Language Toolkit, NLTK)是一套开源的程序集，包括象征性和统计自然语言处理，提供超过 50 个语料库和词汇集资源和一套文本处理工具^[2]。

2.2 聚类技术

2.2.1 整体思想

聚类技术已是文本挖掘和信息检索等多个领域的研究热点，其“目标是将一组对象划分为若干组或类别，简单地说就是相似元素同组、相异元素不同组的划分过程^[3]”，它是一种无指导学习的基本方法。在葛和陈的研究中^[1]，利用文档聚类的流程（图 1），将同一题目的多篇作文按内容或者说所用词汇的相似程度聚类，只以文章所用词汇为标准进行层级聚类，最终根据文章所用词汇的相似程度形成一棵类别树，该树上与大多数节点相似度较低的个别节点会被反馈给教师做最后的判断，因此，该种聚类标准是统一的，属于硬聚类中的层级聚类。

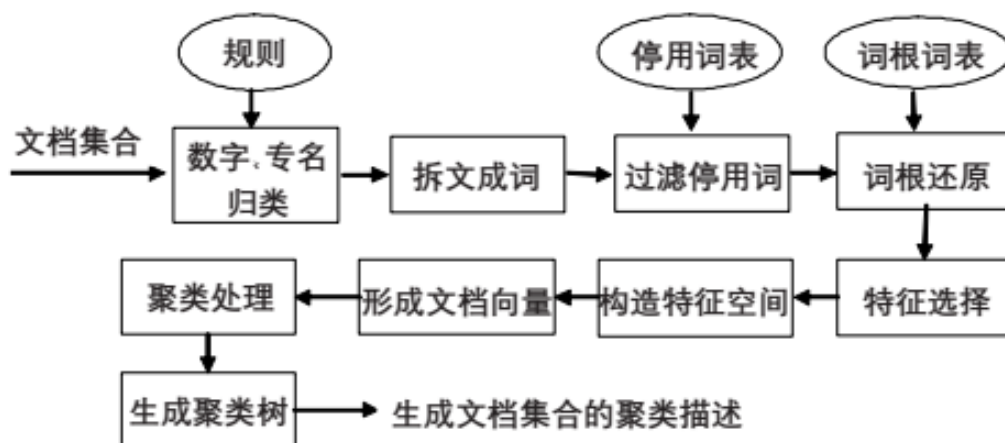


图 1 聚类流程

2.2.2 实现步骤

基于向量空间模型的常见文档聚类算法有分割算法中的 k-means 算法和层级算法中的凝聚层级算法(Hierarchical Agglomerative Clustering, HAC)

k-means 算法以 k 为参数，把 n 个对象分为 k 个簇，以使簇内具有较高的相似度。

其基本的思路是

1. 从数据集 $\{x_n\}_{n=1}^N$ 中任意选取 k 个赋给初始的聚类中心 c_1, c_2, \dots, c_k ;
2. 对各样本点，计算其与各聚类中心的欧氏距离并获取其类别标号：

$$label(i) = \arg \min_j \|x_i - c_j\|^2;$$

3. 重新计算 k 个聚类中心 $c_j = \sum_{s: label(s)=j} X_s / N_j, j=1, 2, \dots, k$ [4]

这种算法实质是一种多次迭代的方法，把每篇文档看成一个对象，利用文档与聚类中心之间的相似度来进行聚类。

2.3 多项式朴素贝叶斯技术

基于多项式模型的朴素贝叶斯分类算法在信息检索领域被广泛使用，并且被国外的 BETSY 等评分系统所使用，并取得了较好的效果，在李和刘的文章中，

他们采用了该算法，详细描述如下^[5]。

假定 d 表示用空间向量表示的一篇英文作文， $d = \{w_1, w_2, \dots, w_m\}$ ，其中 m 是所选出的相邻二元词组特征的个数， w_i 表示第 i 个相邻二元词组特征在作文文档 d 中的权重值。对测量集中的某个文档 d ，预测它属于某个分数档 $c_i (i \geq 1, i \leq k)$ 的概率为 $P(c_i | d)$ ，其中 k 为类别个数，表示不同的分数档。依据贝叶斯定理，
$$P(c_i | d) = \frac{P(c_i)P(d | c_i)}{P(d)}$$
，在基于多项式模型的朴素贝叶斯分类算法中，每篇作文被看成是有所提取的相邻二元词组特征的一个抽样，对于给定的一篇作文，其得到某个分数的概率为包括在该作文中属性的概率乘积。

3 总结与展望

3.1 总结

对自己的工作做个总结，主要工作如下：

- (1) 对当今国内外的英语自动评分作文简要进行了回顾
- (2) 对其部分原理进行了探究，并提供了三种不同的思路

3.2 展望

在调查的这些文献中，笔者发现它们大多以国外的成熟系统举例，而很少提及国内的软件。诚然，与国外作文自动评价系统的研究已经有60多年的历史相比，国内在这一领域的研究相对滞后，且以汉语作文的自动评分系统研究为主流。国内关于英语作文自动评价系统的研究起步更晚，数量更少^[6]。

在不远的将来，希望业内可以推出一套具有自主知识产权、比肩E-Rater的系统，这不单是技术上实力的体现，更将能推动整个中国英语教育事业向智能教学迈出一大步！

另外，我们不妨可以设计出具有自动评分语文作文的机器，这样能够在高考、中考等重大考试中进一步的实现公平、提高效率。

4. 致谢

感谢巍巍老师在疫情期间的坚持付出！虽然一开始报错了信息检索（图书馆），但是后来发现这里讲得也很有趣，就坚持下来了。我觉得这门公选课提供的思想是很重要的，虽然没有计算机编程的相关基础，但是仍能从中获益匪浅。再次感谢巍巍老师！

参考文献

- [1] 葛诗利, 陈潇潇. 文本聚类在大学英语作文自动评分中应用[J]. 计算机工程与应用, 2009, 45(06):145-148.
- [2] 陈贻东. 计算机辅助英语作文批阅系统的研究与设计[D]. 中国科学技术大学, 2016.
- [3] Manning C D, Schutze H. 统计自然语言处理基础[M]. 苑春法, 译. 北京: 电子工业出版社, 2005:310.
- [4] 《信息检索》课件.
- [5] 李霞, 刘建达. 适用于中国外语学习者的英文作文全自动集成评分算法[J]. 中文信息学报, 2013, 27(05):100-106.
- [6] 雷晓东. 英语作文自动评价系统技术的国内研究与应用[J]. 科技视界, 2015(35):43-45.

附录

统计文章中每篇文章的词频

```
def get_count_vectors(essays):  
    vectorizer = CountVectorizer(max_features=10000, ngram_range=(1, 3),  
stop_words='english')  
    count_vectors = vectorizer.fit_transform(essays)  
    feature_names = vectorizer.get_feature_names()  
    return feature_names, count_vectors  
feature_names_cv, count_vectors=get_count_vectors(data[data['essay_set']])
```

```
] == 1]['essay'])
```

构建训练集和特征集。利用线性回归和岭回归进行训练，两者差距不大。

```
X_cv = count_vectors.toarray()
y_cv = data[data['essay_set'] == 1]['domain1_score'].values
print(X_cv.shape)
print(y_cv.shape)
X_train, X_test, y_train, y_test = train_test_split(X_cv, y_cv,
test_size = 0.3)

linear_regressor = LinearRegression()
linear_regressor.fit(X_train, y_train)
y_pred = linear_regressor.predict(X_test)

# The coefficients
print('LinearRegression Coefficients: \n', linear_regressor.coef_)

# The mean squared error
print("LinearRegression Mean squared error: %.2f" %
mean_squared_error(y_test, y_pred))

# Cohen's kappa score: 1 is complete agreement
print('LinearRegression Cohen's kappa score: %.2f' %
cohen_kappa_score(np rint(y_pred), y_test))

print('LinearRegression MAPE: %.2f' %
np.average(np.abs((y_test-y_pred)/y_test)))

print('-' * 50)
```

```
ridge = Ridge(alpha=1.0)
ridge.fit(X_train, y_train)
y_pred = ridge.predict(X_test)
print('Ridge Coefficients: \n', ridge.coef_)
print("Ridge Mean squared error: %.2f" % mean_squared_error(y_test,
y_pred))
print('Ridge MAPE: %.2f' %
np.average(np.abs((y_test-y_pred)/y_test)))
```

转载自：nlp 的一个小测试：英文作文自动评分

https://blog.csdn.net/shange19/article/details/103860548?ops_request_misc=&request_id=&biz_id=102&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-0