

用户型德语初学者词典 DeutschLernen 设计理念

计 41 牛行知数 33 赵丰

June 16, 2017

摘要

在国内外语学习的热潮下双语词典的编纂一直是为人们所忽视的主题，传统词典一直由少数专家把关，编一部词典的周期长，门槛高；现代信息技术的进步为使用计算机网络技术进行协作式编辑，充分考虑用户需求，缩短编辑周期和提高实用性方面提供了可能。本次项目试图以用户型德语初学者词典为突破口探索这种可能性。目前已经实现了词典的浏览和在线编辑模块，背单词模块已经完成理论上的初步分析。

目录

1	问题背景	2
2	设计目标描述	2
3	单词查询	2
3.1	数据视角	2
3.2	界面视角	2
4	用户编辑	2
5	背单词	3
5.1	词汇量统计方法	3
5.2	用户词汇量测量方法	3
5.3	具体实施步骤	4
5.4	单词抽样方法	4
5.5	Score 算法	5
5.5.1	算法描述	5
5.5.2	精度分析	5
6	致谢	6
A	实现细节	7
A.1	基本信息	7
A.2	单词格式定义	7
A.2.1	名词格式定义	7
A.2.2	动词格式定义	8
A.3	定义选项	9
A.4	在线编辑词条功能的实现细节	9
B	Second Appendix	10
B.1	JML	10
B.2	CML	10
B.3	安全概率	11
B.4	能力参数 β 满足的方程	11
B.5	Rasch 模型 Fisher 信息量	11
B.6	Rasch 模型 Bias 数值计算	13

1 问题背景：我国双语学习型词典的设计缺位

尽管英汉词典学是我国外语人才培养的一个二级学科，但我国真正流行的词典几乎都是所引进或合作编纂出版的外来词典，其编纂设计者与使用者形成了主客二分的疏离关系，设计者对于需求的认知主要源于编者主体的专业知识判断，而非对实际用户需求调研的结果。[1]

我国原创英汉汉英类词典，在国内词典市场的份额缺失严重，这和脱离用户需求，盲目照翻单语词典不无关系。

从表面上看，我国电子词典呈现出一片繁荣景象，所涉语言从英汉延伸至其他小语种，但实际情况是电子产品公司与计算机软件开发公司对电子词典表现出极大的热情，辞书出版机构将纸制词典的电子版权转让给电子出版商，而后者只是简单地把印在纸上的东西搬进芯片。

欧美电子词典以辞书为本体来开发电子词典，注重词典数据库的建设，而我国内地则是以电子为本体，先是由 IT 公司开发，出现问题后又转向引进权威词典。[2]

2 设计目标描述

电子词典的核心功能是单词查询的浏览功能。电子词典的界面设计一方面要继承纸制词典的风格，另一方面又要发挥出新媒介的优势。考虑到外语学习者的实际需求，我们在词典数据的基础上设计了背单词的模块，希望利用计算机辅助测试和数据分析的方法帮助学习者更好地入门。

为实现上述设计目标，我们以网页为平台开发了 Klick Auf Deutsch Hilfer(点击德语助手)。目前点击德语助手包含以下三个模块：单词查询，用户编辑和背单词，关于该项目的源代码可以从附录 [3] 给的链接进行下载，下面的部分是对各模块的简要综述。

3 单词查询

3.1 数据视角

我们设计词典抛弃了一般电子词典采用的急键值加表项的方式，而采用超文本标记语言 xml 组织每一个词项。单独的 xml 文件以词条的编号命名，如 1.xml 表示名词类中的第一个，词项是 Abend(晚上)；V100.xml 表示动词类中的第 100 个，词项是 wissen(知道)，比如：

```
<Entry category="Substantiv">
  <Stichwort>Abend</Stichwort>
</Entry>
```

xml 可以自定义元素和属性，为此我们采用了文档类型定义 dtd 的方法，考虑到不同词性的词有不同的属性，每一个词性我们单独定义了一个 dtd 文件，

```
<!ELEMENT Entry (Stichwort, Einheit, Anteil,
Genus, Pluralform, GenitivSingular,
zusammengesetzteWörter, Synonymgruppe,
Antonymgruppe, Kollokationen,
AllgemeineErläuterungen, Comment?)>
<!ELEMENT Stichwort (#PCDATA)>
```

在 xml 文档的头部，显示指明了它被哪一个 dtd 所约束，如 NounModel.dtd。虽然不同词性的词有不同的 dtd，但其大致结构相同，关于我们设计的词项的具体结构，见附录 [A.2]。

考虑到目前大多数在线的外语词典都支持例句查询功能，我们也尝试了通过提前对单词在语料库中的位置建立索引，使得用户可以用一个词查找它出现的上下文，这在语料库分析中叫 concordance，但由于自然语言十分复杂，一个词有多种不同的形式以及语义，因此能否对简单匹配词形的 concordance 进行改进是我们下一步研究的一个内容。< *internal>

3.2 界面视角

</internal>

4 用户编辑

< *internal> 此段是为了赶工作量临时写的，很多地方凭个人感觉，没有出处，待完善。</internal> 世界上最大的百科全书维基百科是以多人协作的方式积少成多发展起来的，维基百科的成功使得在线编辑引擎 wiki 十分流行，相比于维基百

科, 维基词典 y 虽然知名度不是那么高, 但也是利用 wiki 发展起来的多语词典。wiki 也可以看成一种赋码语言, 可以看成 html 的简化版。但对于一般文职人员来说, 学习 wiki 的语法仍有一定难度, 而维基百科或者维基词典本身在线编辑的界面就是提交 wiki 代码的一个 TextArea, 这种设计在一定程度上限制了一些想参与但不是很懂 wikis 代码的人的参与。目前已经有一些相当成熟的 wiki 可视化编辑器, 但一方面需要用户下载安装, 另一方面对于特定需求来说, 编辑效率不如掌握 wiki 语法后直接写的效率高。

针对这两个问题, 我们尝试在我们设计的 xml 词条格式框架下开发用户在线编辑词条的功能, 设计的目标是让用户无需了解技术背景也能轻松地在线编辑词条。具体实现细节可以参考 [(A.4)]。

5 背单词

我们打算基于已有的词典内容开发背单词的模块, 传统的背单词往往基于简单的随机因子和分数累加, 在开发之前, 我们仔细学习了 Item Response Theory, 并以此理论为基础, 建立单词测试的数学模型, 用统计学中分析数据的方法, 在数据库和客户端-服务器架构的技术基础上尝试实现背单词模块, 但由于时间所限, 目前该模块只完成了初步的理论分析, 主要结果由下文内容给出。

5.1 词汇量统计方法

<*internal> [4] 中给出了估计正常人遇到的词汇量的基本方法, 首先统计出 Alphabetical Type (N) 与语料库规模 (M) 的关系, Herdan's Law 指出 N 是 M 的幂函数且幂指数小于 1, 其次通过抽样调查的方法统计某一特定人群在一段时间内 exposed to 语言输入的 tokens, 对这些 tokens 进行统计去重再外推即得到 encountered AT 值。</internal> [4] 指出词汇统计应该基于 word family 而不是对 lemma (词项) 统计, word family 是把一些在词形上有派生关系的词也归为一类, 比如 Lehrer 和 Lehrerin(男老师和女老师) 属于同一个 word family, 但却是两个 lemma。

基于 word family 的统计思想和语料库 tokenization 的初步处理, 我们尝试通

过人工的方法建立一个 wordfamily 的资料库, 考虑到数据传输和处理的简易性, 这个资料库使用了 json 的数据结构。

5.2 用户词汇量测量方法

传统的词汇测量方法是基于 Classical Test Theory 对于每个 test item, 一般结果是二值的, 即只有对和错两个 result, 累加后的结果可以作为 observed score。CTT 认为 $X=T+E$, T 是 true score, E 是 error. CTT 还有三个基本假设:

1. E 的期望为 0
2. T 和 E 不相关
3. 不同次测量结果 (对不同 participants) 相互独立

该测量的性能评估用 reliability 表示, 其数学定义为

$$\rho_{XT}^2 = \frac{\sigma_T^2}{\sigma_X^2} \quad (1)$$

在心理学领域, 要测量某个 latent variable, 不能采用独立重复试验的方法, 如果要基于单次测量估计 ρ_{XT} , 该单次测量应该在 subscale 上应该由同样能反应出被测测量 latent variable 水平的 item 组成。基于不同 participants, 可以得到 item 矩阵, 分析该 item 矩阵可估计 ρ_{XT} 。该方法的数学模型如下: 假设一 test 有 k 个 items $u_j, j = 1, \dots, k$, 对第 i 个 participant 其总得分分为:

$$X_i = \sum_{j=1}^k U_{ij} \quad (2)$$

上式中 U_{ij} 表示对第 i 个参与者在第 j 个 item 上 observed score。可以证明, Cronbach's alpha 是 ρ_{XT} 的下界。

$$\alpha = \frac{k}{k-1} \left(1 - \frac{\sum_{i=1}^k \sigma_{U_j}^2}{\sigma_X^2} \right) \quad (3)$$

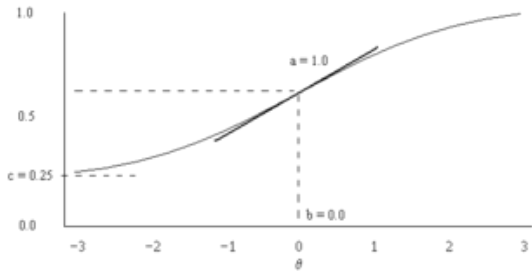
α 介于 0,1 之间, 用于评估 Test 性能, 一般认为 α 值在 0.9 以上会有 redundancy of items, 但对于 individual high-stakes testing 这又是必要的。在 CTT 框架下, α 只能用于评估 Test 总体性能, 如果要做 item analysis, 对于每个 item, 计算 p value(表征 item difficulty) 和 item-total correlation(表征 discrimination). CTT 的问题在于评估 Test 性

能和被试者特征 (examinee characteristics) 有关, 而且对于不同被试者假定 true score 均值不同方差相等。在 psychometrics, 一般不使用 CTT 而用 IRT 方法 (item response theory)。这是一种基于 item 而不是 test(由许多 item 组成) 的方法, IRT 要估计 latent variable θ , 假设各个 item 彼此独立, 被试者对某个 item 的回答正确的概率用 IRF(item response function) 建模。一般 θ 会做一个归一化, 使得其均值为 0 标准差为 1, 这样 $\hat{\theta}$ 作为 θ 的估计值一般在 -3 到 3 之前, 非常接近 0 表示水平中等。这种归一化给不同测试集之间相互比较提供了方便。IRF 函数有多种不同的建模方式, 一般常用的有 Logistic model:

$$p_i(\theta) = c_i + \frac{1 - c_i}{1 + \exp(-a_i(\theta - b_i))} \quad (4)$$

上式中 i 表示被试者的编号, a, b, c 是 item 的参数, 分别表征 discrimination, difficulty 和 pseudo guessing, 可以从下图 (ICC 曲线, item characteristic curve) 形象地说明这三个参数

Figure 1: 三个参数的 IRF



由上图可以看出 $\theta = -3$ 时被试者仍有概率 c 答对, 对 4 选 1 的 multiple choice, $c = \frac{1}{4}$. b 是 $p(\theta) = \frac{c+1}{2}$ 的点, 即最大值 1 和最小值 c 的平均值的点, 同时也是 LRF 曲线最陡的点, 可以衡量 difficulty. a 和 $p'(b)$ 成正比, a 越大曲线两级分化越严重, 即 ability θ 小于某一个阈值答对的正确率为 c , 大于此阈值答对的正确率为 1. 此外 LRF 曲线还可以从标准正态分布的 cdf 建模。三个参数的 IRF 虽然精确, 但实际中估计参数比较繁琐, 一般常用的是 1 个参数 (b) 的 Rasch Model, 其可以简化表述为第 k 个 person 在第 i 个 item 上答对的概率为

$$P(X_{ki} = 1) = \frac{\exp(\beta_k - \delta_i)}{1 + \exp(\beta_k - \delta_i)} \quad (5)$$

上式中 β_k 表示 ability, δ_i 表示 difficulty. 在获得 person \times item 的二维表格数据后, 要先根据数据估计 Rasch Model 的参数 $\vec{\delta} = (\delta_1, \dots, \delta_I)$, 常用的方法有极大似然法, CML, EM 等, 关于这三种方法在 Rasch Model 参数估计的具体讨论, 见 B.1。

5.3 具体实施步骤

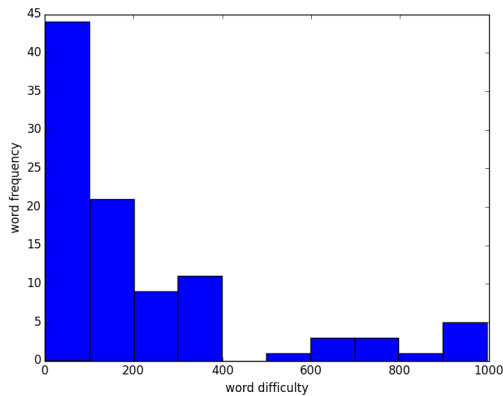
下面的列表给出了我们基于 Rasch Model 关于 CAT (computer adapted testing) 背单词的实施方案:

1. 根据课本内容统计词频, 归一化后作为每个单词难度的近似替代量
2. 每一个用户初始化背单词能力为 0, 每一次背单词后保留其该次背单词能力的估计值, 在下一次背单词时采用之前能力值的加权平均值, 对于该平均值-单词难度 > 3 的单词则不予考虑, 在其他单词中按单词难度进行重要度抽样, 样本数量为 N 个, 作为该次背单词的测试集。每次用户的有效测试 (没有中途退出和缺失值) 保存到服务器的数据库用来更新单词难度。
3. 定期更新单词难度之前集齐一定数量的测试结果, 应考虑到用户的能力变化曲线, 有选择地剔除某一部分数据再用 CML 全局计算单词难度, 将计算值与原有的频率值做平均。

5.4 单词抽样方法

考虑到总单词数在几百左右, 如果分的 bins 太多, 每个 bins 内单词过少, 因为难词多而被抽样的概率低, 则大量的难词无法被抽到; 简单词少而被抽样的概率高, 则简单词几乎必然被抽到。因此在实际操作中, 我们把单词分为 easy, middle 和 difficult 三类。分类的标准是假设单词难度归一化后是近似指数分布的随机变量 X (我们先把单词按从易到难排序, 同时用随机模拟的方法生成同样长度的指数分布的随机数, 将该随机数序列从小到大排序后作为给定单词总体归一化后的难度), $P(X < 1) \approx 64\%$, $P(X > 1 \wedge X < 2) \approx 23\%$, $P(X > 2) \approx 13\%$, 基于此, 我们设定简单词的采样率为 $\frac{10}{13}$, 中等词的采样率为 $\frac{5}{23}$, 难词的采样率为 $\frac{3}{64}$, 则可以算出三

类词在样本中的比率为 10 : 5 : 3, 对于给定的单词总体, 先将其分成三类; 对于给定的样本数 N , 计算出在简单词类中要无放回的抽取 $\frac{10}{18}$, 中等词类中抽取 $\frac{5}{18}$, 难词类中抽取 $\frac{3}{18}$. 下图是从 1000 个单词中按上述方法抽取 100 个单词的模拟结果:



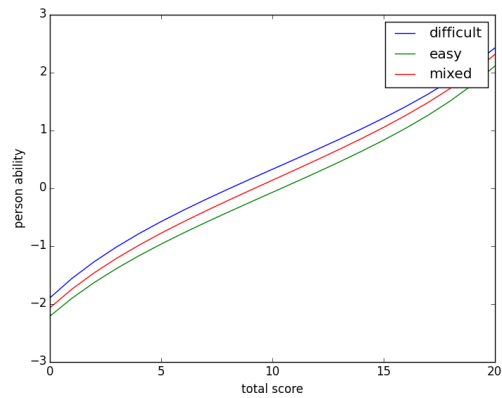
上图中纵轴表示每个 bins 的大小, 横轴是将 1000 个单词按难度排序, 最难的序号为 1000, 最易的为 1。由上图可以看出, 该方法抽到的大部分样本集中在难度排名前 20% 的单词, 占到样本的 60% 以上。由上面的描述可知, 该方法不依赖于单词的绝对难度差异而是根据排名进行的抽样, 对于规模为几百的总体, 效果较好。该方法的局限性在于由于每类抽样比固定, 对于给定的总体 ($len=N$), 样本数量要受到限制, 一个比较保守的上限是 $\frac{117N}{1000}$, 根据概率论 B.3 的知识可以计算出当 $N=200$ 时临界安全概率为 99.8%, 即当抽样数恰好等于上界时无放回抽样不会报错的概率为 99.8%, 这对于一般的情形 ($N \geq 200$) 可认为不安全是小概率事件。

5.5 Score 算法

5.5.1 算法描述

Rasche Model 算出的 ability 在测试题目给定的情况下和总分具有非线性的一一对应关系, 通过极大似然的方法可以推导出 person 的 ability 代数方程 B.4, 用牛顿法求解方程即得到 ability 参数。

实际实现时发现牛顿算法在分数接近满分和接近零分时误差较大, 改用优化的方法求 β 在 $[-3, 3]$ 区间的极大值则无此问题, 下图是利用仿真数据得到的分数-能力曲线:



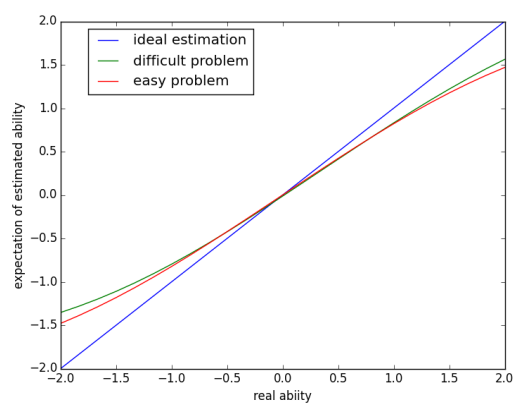
上图中三组数据分别是 20 个难题, 20 个容易题, 和 10 个难题和 10 个容易题的混合组, 由上图可看出每一条曲线有如下特点:

1. 能力和分数的关系在高分和低分段斜率比较大, 非线性性比较明显, 而在中部接近线性。
2. 平均水平能力为 0 对应答出来一半的题目, 曲线具有对称性。

• 比较不同的曲线也符合直观, 答出同样数量的题, 对于难题组能力高, 混合组能力次之, 最末为简单题组。通过用 Ability 而不是 score 来衡量从而消除了某一次 Test 题目的影响而在一个统一的 scale 上比较。

5.5.2 精度分析

对于单次测试而言, 可以用总的 Fisher 信息量 $I(\beta)$ 衡量精度, 一般而言, 对于某一个估计的值 $\hat{\beta}$, $I(\hat{\beta})$ 越大则表明估计统计量的方差越小, 能力估计的精度越高。B.5 基于 MLE 的方法一般都是有偏的, 即估计统计量 $\hat{\beta}$ 的均值不等于 β , 由于引入了先验的分布, 这种偏差在两级状态会非常明显 B.6, 下图是比较两组题目 bias 的结果: 如果 person 的能力接近平均水平 ($ability \approx 0$), 则几乎没有 bias, 但两级的 bias 会比较大。由于采用了先验的正态总体假设, 对于高分, 会低估 person 的能力而对于低分则会高估 person 的能力。如果题目较难, 高分段的 bias 减小而低分段的 bias 增大。



6 致谢

感谢这些年来一直陪伴 DeutschLernen 成长的新老同学，大家的需求是软件前进的动力，大家的反馈是软件提高的机会。

热烈欢迎各位到 DeutschLernen Github 主页贡献！

A 实现细节

A.1 基本信息

Need to be perfected.

A.2 单词格式定义

A.2.1 名词格式定义

```
1 <(*NounModel.dtd)
2 <?xml version="1.0" encoding="UTF-8"?>
```

每一个合法的 xml 首先都有一个根元素名为 Entry。Entry 下必须依次出现如下元素：

1. Stichwort
2. Einheit
3. Anteil
4. zusammengesetzteWörter
5. Synonymegruppe
6. Antonymegruppe
7. Kollokationen
8. AllgemeineErläuterungen

```
3 <!ELEMENT Entry (Stichwort, Einheit, Anteil, Genus, Pluralform, GenitivSingular, zusammengesetzteWörter,
4 Synonymegruppe, Antonymegruppe, Kollokationen, AllgemeineErläuterungen, Comment?)>
```

这里有的子元素结构比较简单，比如 Stichwort 下只包含了词形的信息。

```
5 <!ELEMENT Stichwort (#PCDATA)>
```

Einheit/Anteil 元素是适配考虑到清华大学德语教学正在使用的教材为每个单词提供的其所在单元和所在单元具体模块的信息。

```
6 <!ELEMENT Einheit (#PCDATA)>
7 <!ELEMENT Anteil (#PCDATA)>
```

对于德语名词的性、第二格、复数分别用元素来表示。

```
8 <!ELEMENT Genus (#PCDATA)>
9 <!ELEMENT Pluralform (#PCDATA)>
10 <!ELEMENT GenitivSingular (#PCDATA)>
```

zusammengesetzteWörter 元素提供德语中的和该词有关的复合词的信息，

```
11 <!ELEMENT zusammengesetzteWörter (KompositaCollection, abgeleiteteWörter)>
```

由于德语中复合词数量更多，相应的我们在 zusammengesetzteWörter 下设了 KompositaCollection 和 abgeleiteteWörter 两个子元素，分别包含合成词类和派生词类。

```
12 <!ELEMENT KompositaCollection (_K|K_)*>
13 <!ELEMENT abgeleiteteWörter (hierzu)*>
```

在合成词类下，为支持后期多种检索方式，我们将其主要分为 K_ 和 _K 型，分别表示这个词项在该合成词的位置。是在前面还是后面，派生词类下对每个由该词项派生的词必须注明它的词性，否则按照 dtd 的语法检查规则，整篇文档就是不合法的。

```
14 <!ELEMENT _K (#PCDATA)>
15 <!ELEMENT K_ (#PCDATA)>
16 <!ELEMENT hierzu (#PCDATA)>
```

整个词条中最重要的部分是 AllgemeineErläuterungen，其结构也最复杂。

```
17 <!ELEMENT AllgemeineErläuterungen (Eintrag+)>
```

考虑到一词多义的可能性，该“一般性释义”下设若干个 Eintrag，

```
18 <!ELEMENT Eintrag (Chinesisch, BeispielSammlung)>
```


并且至少要有一个 Eintrag. 每一个 Eintrag 有 Chinesisch 和 BeispielSammlung 两个子元素, 分别是汉语释义和例句集,

```
19<!ELEMENT Chinesisch (#PCDATA)>
20<!ELEMENT BeispielSammlung (Beispiel*)>
```

每一个例句集是由若干个 Beispiel 组成的

```
21<!ELEMENT Beispiel (Satz,Übersetzung)>
```

而每一个 Beispiel 由 Satz 和 Übersetzung 组成。该部分处于整个文档树最深的位置。

```
22<!ELEMENT Satz (#PCDATA)>
```

```
23<!ELEMENT Übersetzung (#PCDATA)>
```

对于 Entry 下最后面的三个元素, 分别表示同义词集合、反义词集合和词组集合, 其中我们在编辑的过程中发现, 同反义词集合具有稀疏性。

```
24<!ELEMENT Synonymgruppe (Sym*)>
```

```
25<!ELEMENT Sym (#PCDATA)>
```

```
26<!ELEMENT Antonymgruppe (Anm*)>
```

```
27<!ELEMENT Anm (#PCDATA)>
```

```
28<!ELEMENT Kollokationen (K*)>
```

```
29<!ELEMENT K (#PCDATA)>
```

对每一个单词设置注释项容纳用法等其他信息。

```
30<!ELEMENT Comment (#PCDATA)>
```

使用名词 (Substantiv) 作为超链接的单词的默认词性。

```
31<!ATTLIST hierzu category (Substantiv | Adjektiv | Verben) "Substantiv">
```

使用名词 (Substantiv) 作为德语单词的默认词性。

```
32<!ATTLIST Entry category (Substantiv | Adjektiv | Verben) "Substantiv">
```

我们考虑到单词之间错综复杂的语义关系, 在 Eintrag 的相关子元素下设置了 link 属性, 其值为相对应单词的文件名称, 如 essen 词项下某个合成词 Abendessen 的 xml 表示为

```
<K link="1.xml">Abendessen</K>
```

所以 Abendessen 可以链接到 Abend 这种方法不仅为展示数据提供了统一接口, 还为用网络的方法做关联分析提供了数据基础。下面是属性列表,

```
33<!ATTLIST _K link CDATA #IMPLIED>
34<!ATTLIST K link CDATA #IMPLIED>
35<!ATTLIST K_ link CDATA #IMPLIED>
36<!ATTLIST Sym link CDATA #IMPLIED>
37<!ATTLIST Anm link CDATA #IMPLIED>
38<!ATTLIST hierzu link CDATA #IMPLIED>
39<!ATTLIST Stichwort link CDATA #IMPLIED>
40<!ATTLIST Stichwort Band CDATA #IMPLIED>
41<!ATTLIST Stichwort Bild CDATA #IMPLIED>
42<!ATTLIST Stichwort Audio CDATA #IMPLIED>
43</NounModel.dtd>
```

A.2.2 动词格式定义

<*VerbenModel.dtd>

```
44<!ELEMENT Entry (Stichwort, Einheit, Anteil, Konjugation, zusammengesetzteWörter, Synonymgruppe,
45 Antonymgruppe, Kollokationen, AllgemeineErläuterungen, Comment?)>
```

```
46<!ELEMENT Stichwort (#PCDATA)>
```

```
47<!ELEMENT Einheit (#PCDATA)>
```

```
48<!ELEMENT Anteil (#PCDATA)>
```

```
49<!ELEMENT zusammengesetzteWörter (KompositaCollection, abgeleiteteWörter)>
```

```
50<!ELEMENT KompositaCollection (_K|K_|K_)*>
```

动词 dtd 模板文件与名词大同小异, 但我们加入了动词变位这一类属性。

```
51<!ELEMENT Konjugation (Präsen, Präteritum, Perfekt)>
```

三类时态各对应 6 个人称。

```
52<!ELEMENT Präsen (ich, du, er_sie_es, wir, ihr, sie_Sie)>
```

```
53<!ELEMENT Präteritum (ich, du, er_sie_es, wir, ihr, sie_Sie)>
```

```
54<!ELEMENT Perfekt (ich, du, er_sie_es, wir, ihr, sie_Sie)>
```



```

55<!ELEMENT ich (#PCDATA)>
56<!ELEMENT du (#PCDATA)>
57<!ELEMENT er_sie_es (#PCDATA)>
58<!ELEMENT wir (#PCDATA)>
59<!ELEMENT ihr (#PCDATA)>
60<!ELEMENT sie_Sie (#PCDATA)>

61<!ELEMENT Comment (#PCDATA)>
62<!ELEMENT abgeleiteteWörter (hierzu*)>
63<!ELEMENT _K (#PCDATA)>
64<!ELEMENT K_ (#PCDATA)>
65<!ELEMENT _K_ (#PCDATA)>
66<!ELEMENT hierzu (#PCDATA)>
67<!ELEMENT AllgemeineErläuterungen (Eintrag+)>
68<!ELEMENT Eintrag (Chinesisch,BeispielSammlung)>
69<!ELEMENT Chinesisch (#PCDATA)>
70<!ELEMENT BeispielSammlung (Beispiel*)>
71<!ELEMENT Beispiel (Satz,Übersetzung)>
72<!ELEMENT Satz (#PCDATA)>
73<!ELEMENT Übersetzung (#PCDATA)>
74<!ELEMENT Synonymgruppe (Sym*)>
75<!ELEMENT Sym (#PCDATA)>
76<!ELEMENT Antonymgruppe (Anm*)>
77<!ELEMENT Anm (#PCDATA)>
78<!ELEMENT Kollokationen (K*)>
79<!ELEMENT K (#PCDATA)>
80<!ATTLIST hierzu category (Substantiv | Adjektiv | Verben) "Substantiv">
81<!ATTLIST Entry category (Substantiv | Adjektiv | Verben) "Substantiv">
82<!ATTLIST _K link CDATA #IMPLIED>
83<!ATTLIST K link CDATA #IMPLIED>
84<!ATTLIST K_ link CDATA #IMPLIED>
85<!ATTLIST Sym link CDATA #IMPLIED>
86<!ATTLIST Anm link CDATA #IMPLIED>
87<!ATTLIST hierzu link CDATA #IMPLIED>
88<!ATTLIST hierzu chinesisch CDATA #IMPLIED>
89<!ATTLIST hierzu trennbar CDATA #IMPLIED>
90<!ATTLIST K_ chinesisch CDATA #IMPLIED>
91<!ATTLIST _K chinesisch CDATA #IMPLIED>
92<!ATTLIST _K_ chinesisch CDATA #IMPLIED>
93<!ATTLIST K_ category CDATA #IMPLIED>
94<!ATTLIST _K category CDATA #IMPLIED>
95<!ATTLIST _K_ category CDATA #IMPLIED>
96<!ATTLIST Perfekt hilfsverb CDATA #REQUIRED>
97<!ATTLIST Eintrag transitiv CDATA #IMPLIED>
98<!ATTLIST Stichwort Band CDATA #IMPLIED>
99<!ATTLIST Stichwort Bild CDATA #IMPLIED>

<VerbenModel.dtd>

```

A.3 定义选项

A.4 在线编辑词条功能的实现细节

```

100 <*cfg>
101 预留给 Django Server setting 的说明和代码。
102 </cfg>

```

当用户点击编辑按钮后，词条的 xml 代码被转成 html 的 inputBox, radioButton 等控件，用户只需点击按钮即可添加或删除某一个特定的子条目，并在相应的输入框中填写子条目相应的信息。当编辑完成后，用户提交后便可以直接看到已经排版过的编辑结果，但最终在服务器端是否修改相应的词条版本需要管理员再确认。

wiki 有一套成熟的历史版本管理机制，由于我们的代码 (包含 xml) 使用的是 git 版本控制系统，

所以每一个词条编辑的历史记录可以直接作为版本控制的一部分被记录下来。

B Second Appendix

B.1 JML

首先讨论 JML(joint maximum likelihood) 的方法,observed data matrix 联合概率似然函数为

$$\begin{aligned} \log(\Lambda) = & \sum_{k=1}^N \beta_k r_k - \sum_{i=1}^I \delta_i s_i + \\ & \sum_{k=1}^N \sum_{i=1}^I \log(1 + \exp(\beta_k - \delta_i)) \end{aligned} \quad (6)$$

其中 $r_k = \sum_{i=1}^I x_{ki}$, 表示第 k 个 person 的总分, $s_i = \sum_{k=1}^N x_{ki}$, 表示第 i 个 item 的总分。对对数似然函数关于 δ_i 和 β_k 求偏导, 得到含 β_k 和 δ_i 的非线性方程组为

$$s_i = \sum_{k=1}^N p_{ki}, i = 1, \dots, I \quad (7)$$

$$r_k = \sum_{i=1}^I p_{ki}, k = 1, \dots, N \quad (8)$$

• 上式中 p_{ki} 即为 (5)

B.2 CML

对实际应用来说, 一般 N 很大, 直接求解 (7) 计算量太大。故一般先求只含 item 的边缘概率分布, 在 item 的参数 δ_i 求出的情况下, 由于各个 person 之间相互独立, 只需分别对只含一维参数 β_k 的函数求极大值点即可。对第 k 个 person, 其各 item 得分的 joint distribution 为

$$\begin{aligned} P(\vec{x}_k | \beta_k, \vec{\delta}) &= \prod_{i=1}^I \frac{\exp(x_{ki}(\beta_k - \delta_i))}{1 + \exp(\beta_k - \delta_i)} \\ &= \frac{\exp(r_k \beta_k) \exp(-\sum_{i=1}^I x_{ki} \delta_i)}{\prod_{i=1}^I (1 + \exp(\beta_k - \delta_i))} \end{aligned} \quad (9)$$

由 \vec{x}_k 的联合分布可以求出 r_k 的分布为

$$\begin{aligned} P(r_k | \beta_k, \vec{\delta}) &= \sum_{\vec{y}_1=r_k} P(\vec{y}_1 | \beta_k, \vec{\delta}) \\ &= \frac{\exp(r_k \beta_k) \sum_{\vec{y}_1=r_k} \exp(-\sum_{i=1}^I y_i \delta_i)}{\prod_{i=1}^I (1 + \exp(\beta_k - \delta_i))} \end{aligned} \quad (10)$$

定义 $\gamma_{r_k | \vec{\delta}} = \sum_{\vec{y}_1=r_k} \exp(-\sum_{i=1}^I y_i \delta_i)$, 为 elementary symmetric function, 则条件似然函数 $P(x_k | r_k, \vec{\delta})$ 为

$$\begin{aligned} P(x_k | r_k, \vec{\delta}) &= \frac{P(x_k | \beta_k, \vec{\delta})}{P(r_k | \beta_k, \vec{\delta})} \\ &= \frac{\exp(-x_{ki} \delta_i)}{\gamma_{r_k | \vec{\delta}}} \end{aligned} \quad (11)$$

上式不含 β_k , 说明 r_k 是参数 β_k 的充分统计量。由于各 person 得分相互独立, 只需把 N 个对数似然函数相加即可。

$$\log(\Lambda(\vec{x} | \vec{r}, \vec{\delta})) = \sum_{k=1}^N \frac{\exp(-x_{ki} \delta_i)}{\gamma_{r_k | \vec{\delta}}} \quad (12)$$

B.3 安全概率

问题可转化为 $P(X > 2) = 0.136, Y_i i.i.d \sim \text{Bernoulli}(p = 0.136)$, 求 $P(\sum_{i=1}^N Y_i > K)$ 当 N 较大时, 可以用中心极限定理近似计算:

$$P\left(\frac{\sum_{i=1}^N Y_i - Np}{\sqrt{Np(1-p)}} > \frac{10K/18 - Np}{\sqrt{Np(1-p)}}\right) \quad (13)$$

上式左边为标准正态分布的随机变量, 由上式右边可以看出抽样率 K/N 必须小于 23.4% (这是该抽样方法最大允许的通过率), 否则随着 N 的增大概率趋近于 0. 在实际操作中取 $K = \frac{117N}{1000}$ 满足这个约束, 代入 $N=100$ 可算出概率为 98%, 而且 N 越大安全概率越大。

B.4 能力参数 β 满足的方程

用 β 表示某人的能力, β 的先验分布记成 $p(\beta)$, 一般是正态分布, 表示在没有考试成绩的时候对其能力的估计, 设此人参加了有 $i = 1, 2, \dots, I$ 个 item 组成的测试, 得分为 x_i , 每道题的难度为 δ_i , 由贝叶斯公式, 对其成绩的后验估计为:

$$p(\beta|\vec{x}) = \frac{p(\beta)p(\vec{x}|\beta)}{p(\vec{x})} \propto p(\beta)p(\vec{x}|\beta) \quad (14)$$

β 最有可能的取值为:

$$\underset{\beta}{\operatorname{argmax}} p(\beta)p(\vec{x}|\beta) \quad (15)$$

其中 $p(\vec{x}|\beta)$ 由 Rasch Model 给出:

$$p(\vec{x}|\beta) = \prod_{i=1}^I \frac{\exp(x_i(\beta - \delta_i))}{1 + \exp(\beta - \delta_i)} \quad (16)$$

对含先验分布的对数似然函数 $\log(p(\vec{x}|\beta))$ 关于 β 求得:

$$\frac{p'(\beta)}{p(\beta)} + \sum_{i=1}^I x_i = \sum_{i=1}^I \frac{\exp(\beta - \delta_i)}{1 + \exp(\beta - \delta_i)} \quad (17)$$

上方程的解 β 对各项得分的依赖仅仅通过总分 $\sum_{i=1}^I x_i$ 的形式, 因此总分是参数 β 的充分统计量。由于 Rasch dichotomous Model 对 person 的能力只有一个维度的假定, 在 items 一定的情况下, 相同能力与相同总分一一对应。如果不加先验分布, 在全对和全错两种极端情况下方程 (17) 无解, 因此适当的先验分布是必要的, 有用户 ability 数据的情况下可以拟合正态分布的参数, 在缺少用户数据的初始阶段可以用标准正态分布代替, 此时上式第一项化为 $-\beta$ 。

B.5 Rasch 模型 Fisher 信息量

$I(\beta)$ 的计算公式为:

$$I(\beta) = -E\left(\frac{\partial^2 \log p(\vec{x}|\beta)}{\partial \beta^2}\right) \quad (18)$$

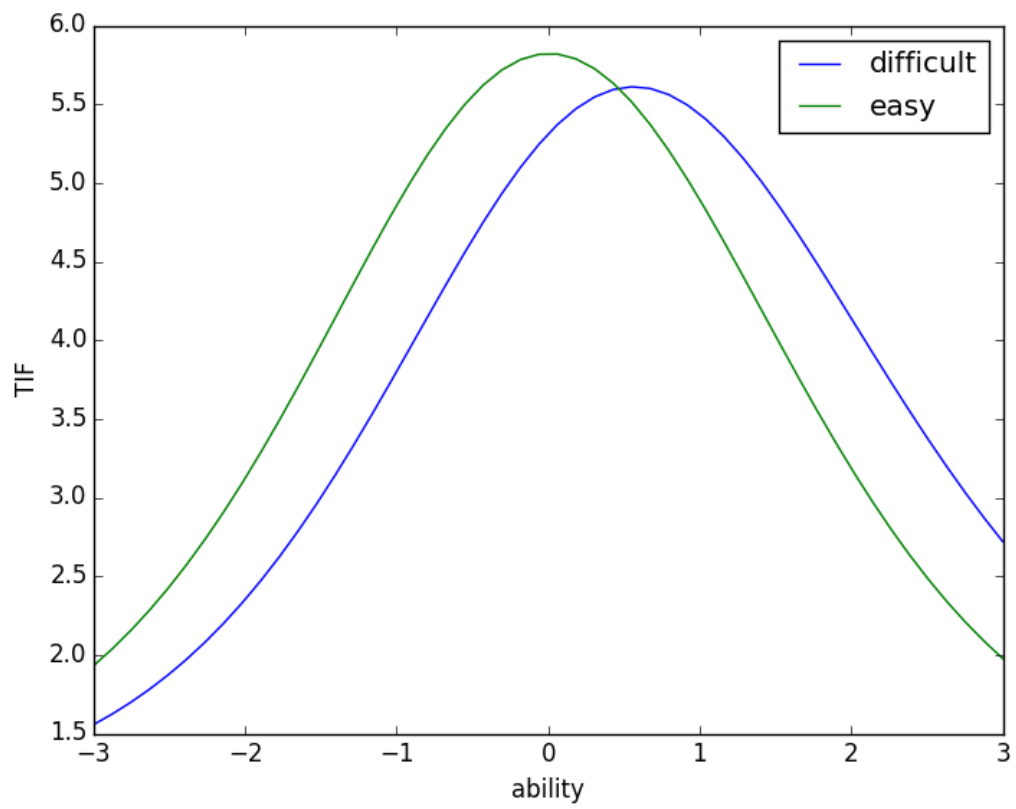
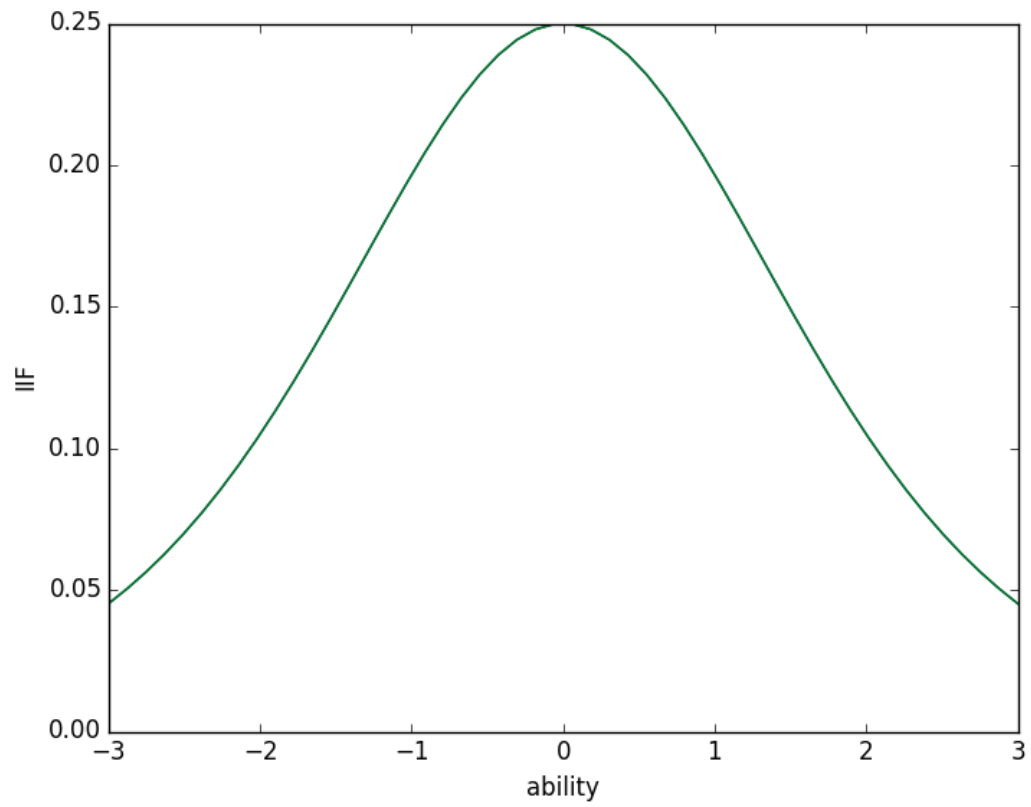
对于 Rasch Model, 代入似然函数表达式, 取先验分布为正态分布, 则有

$$I(\beta) = 1 + \sum_{i=1}^I \frac{\exp(\beta - \delta_i)}{(1 + \exp(\beta - \delta_i))^2} \quad (19)$$

上式中第一项是先验分布的信息量, 后面分别是每一个 item 的信息量 (item information function), 它们彼此独立因而可以相加。上式的和也被称为 Test Information Function(TIF). 对于每一个 item, 其 IIF 有下图所示的形式:

上图是假定 item difficulty 为 0, 当 $\beta - \delta_i > 3$ 时 (题目过难或过易), IIF 已经小于 0.05, 在这种情况下能力估计的误差为比较大。由于先验分布对 TIF 的贡献是常数, 可以在一定程度上减轻能力的极端情况造成的信息量过少的问题。下图是 20 道题的 TIF 结果:

由上图可看出, 对于难题组, 全对的信息量要比简单题大, 而全错的信息量比简单题小, 这符合一般经验。



B.6 Rasch 模型 Bias 数值计算

计算 bias 需要计算给定真实能力后计算关于估计量的期望值, 在对模型没有任何了解的情况下可以采用 Monte Carlo 模拟, 但对于 Rasch Model 由 (17) 式得 $r_I = \sum_{i=1}^I x_i$ 是 β 的充分统计量, 于是首先计算总分 r_I 的分布, 再由

$$E(\hat{\beta}(r_I)) = \sum_{i=1}^I \hat{\beta}(i) P(r_I = i) \quad (20)$$

计算出期望值。在上式中 $\hat{\beta}(i)$ 可以通过对 (15) 求解极大值得到, 而 r_I 的分布可以类比组合数的计算方法递推得到。记 $A_n^k = \sum_{k=1}^n x_k$, 则有如下递推公式:

$$A_n^k = P(x_n = 1)A_{n-1}^{k-1} + P(x_n = 0)A_{n-1}^k \quad (21)$$

上式中 $P(x_n = 1)$ 为第 n 道题的答对概率。该方法计算规模为 $O(n^3)$, 其中 n 为 item 数量, 相比 Monte Carlo 模拟需要大量重复才能得到比较精确的结果在 n 不大时效率比较高。信息量和能力估计的标准差 $\sigma_{\hat{\beta}}$ 之间的关系很难给出理论表达式, 但可以用 $\frac{1}{\sqrt{I(\hat{\beta})}}$ 近似作为标准差的下界。

References

- [1] 双语学习型词典设计特征研究外研社 2013 年版
- [2] 计算词典学上海辞书出版社 2011 年版
- [3] <https://github.com/Leidenschaft/Deutsch-Lernen>
- [4] How Many Words Do We Know? Practical Estimates Of Vocabulary Size Dependent on Word Definition, the Degree of Language Input and the Participant's Age, *frontiers in Psychology*, 2016,V7,Article 1116