基于决策树的德语字素音素转换算法

王永生 李立贵
（同济大学外国语学院 上海 200092）

摘 要 字素音素转换是德语自然语言处理中的难点之一。提出一种基于决策树的字素音素转换的监督学习算法。在一个字素音素平行语料库的基础上，通过决策树进行字素音素转换的监督学习，生成字素音素转换规则。经交叉测试，平均转换正确率可达 98.03%。

关键词 德语字素音素转换 自然语言处理 决策树

中图分类号 TP391 **文献标识码** A DOI:10.3969/j.issn.1000-386x.2019.01.038

GERMAN GRAPHEME-TO-PHONEME CONVERSION ALGORITHM BASED ON DECISION TREE

Wang Yongsheng Li Ligui
(*School of Foreign Languages, Tongji University, Shanghai 200092, China*)

Abstract The conversion of grapheme to phoneme is one of the difficulties in German natural language processing. In this paper, we proposed a supervised learning algorithm based on decision tree for grapheme-to-phoneme conversion. On the basis of a parallel corpus of grapheme and phoneme, the rules of morpheme-to-phoneme conversion were generated by supervised learning of grapheme-to-phoneme conversion through decision tree. Through cross test, the average conversion accuracy can reach 98.03%.

Keywords German grapheme-to-phoneme conversion Natural language processing Decision tree

0 引 言

在德语自然语言处理的过程中，要通过计算机合成一个德语单词的语音，能想到的最简单直接的做法就是事先将德语单词的音标和数字格式的读音存储在数据库中，通过检索，从中提取要合成的德语单词的读音就可以了。可问题是，随着德语语言学的不断发展，会不断涌现大量的新词汇，加之德语的构词功能极为强大，只要符合构词法规则，几乎可以任意创造复合词。但数据库的容量毕竟有限，不可能存储所有词汇，对于那些不在词库里的词汇，即所谓的 OOV（Out-of-vocabulary words）单词，如何取得它们的读音就成了德语自然语言处理系统不得不解决的问题。

德语有 30 个字母，却有多达 48 个音素。很显然，德语中的字母与音素之间并不是完全一一对应的。同一个字母可能对应一到多个音素，如单词 Praxis 和 Text 中的字母 x 的读音是/ks/；同样地，同一个音素也可能与不同的字母或字母组合相对应，如 feucht 中的 ch 读/ç/，而 richtig 中的 g 也读/ç/；而其中的一些元音字母的读音就更为复杂。这显然给单词与其读音之间的转换带来了麻烦。而实际上，在德语单词中存在对应关系的是其中的字素和音素[1]。

表 1 所示的就是德语单词 schaffen 的字素和音素的对应情况。

表 1 schaffen 中字素与音素的对应关系

Sch	a	ff	e	n
ʃ	a	f	ə	n

如此一来，就将单词与其读音之间的转换简化为

收稿日期：2018－08－04。同济大学中央高校基本科研业务费资助项目（1100219123）。**王永生**，高工，主研领域：计算语言学。**李立贵**，副教授。

字素到音素的转换。字素音素转换是语音合成(Speech Synthesis)和语音识别 ASR(Automatic Speech Recognition)的一个重要组成部分和难点问题[2]。

在德语中,字素与音素之间的对应关系比较复杂,有一对一、一对多、多对一或是多对多。比如说,字素 b 和 bb 可与音素/b/对应(如单词 Bahn、Ebbe),而字素 b 还可以与音素/p/对应(如单词 Dieb);再比如音素/k/可能与字素 k、kk、g、ck、c 或 ch 对应(如单词 Komma、Akku、Tag、lecker、Cottbus、Christ)。另一方面,一些字素又可能有多个读音,如 g,可以读/k/,如 Berg,也可以/ç/,如 billig[3]。

字素音素转换是自然语言处理中的一个重要的研究课题,以前的研究人员已进行了大量的研究,如基于规则的方法(Rules-Based Approaches)[4],基于递归神经网络语言模型(Recurrent Neural Network Language Models)的转换算法[5-6],基于 n 元语法(n-gram)的转换算法[7-8],基于隐马尔科夫模型 HMM(Hidden Markov Model)的转换算法[9]等多种方法。但上述一些方法在德语字素音素转换中的效果并不理想[1-10]。

本项目已创建了一个有 10 300 个常用德语词汇的字素和音素对照的平行语料库。本文所要讨论的重点就是如何在这个语料库的基础上,通过构建决策树来解决字素音素的自动转换问题。

1 基于决策树的字素音素转换算法

1.1 选择属性

针对单词中某个需要进行转换的字素提取 4 个属性来组成一个实例,如表 2 所示。

表 2 定义属性

属性名称	G_-1	G_1	P_1	G_2
说明	前一个字素	后一个字素	G_1 代表的字素是元音或辅音	后第二个字素

实例用“属性-值”对来表示,以单词 adoptive 中的字素 v 为例,针对该字素提取的 4 个属性为(i,e,V,E)。

字素 v 的读音比较简单,只对应着 2 个音素,即/v/或/f/,这是个典型的二类问题,为方便算法描述,本文下面将之称为 v_f 问题。表 3 选取了字素 v 的 10 个比较典型的实例。

表 3 字素 v 的实例

No.	G_-1	G_1	P_1	G_2	对应的单词	字素 v 对应的音素
I1	i	e	V	E	adoptive	v
I2	i	E	N	E	akkusativ	f

续表 3

No.	G_-1	G_1	P_1	G_2	对应的单词	字素 v 对应的音素
I3	i	e	V	E	initiative	v
I4	i	E	N	E	intensiv	f
I5	r	e	V	E	konserve	v
I6	i	E	N	E	kreativ	f
I7	r	e	V	E	kurve	v
I8	i	E	N	E	objektiv	f
I9	i	e	V	E	perspektive	v
I10	i	E	N	E	relativ	f

其中 G_1 属性中的 E 表示单词尾,表明目标字素 v 已是单词的尾字素。P_1 属性中的 V、C、N 分别表示目标字素 v 后的第一个字素是一个元音字素、单辅音字素、空(目标字素 v 已是单词的最后 1 个字素)。G_2 属性中的 E 表示目标字素 v 后没有第二个字素。

1.2 分类函数

1.2.1 信息增益与实例分类的熵

构建决策树的关键是定义一个信息增益函数,以用于分类实例库。如公式所示[11]:

$$G(S,A) = E(S) - \sum_{v \in V(A)} \frac{|S_v|}{|S|} E(S_v) \qquad (1)$$

式(1)中的 S 表示目标概念的整个实例集,假定所有实例均属于同一个两类问题,则 $E(S)$ 表示这个两类问题分类的熵(Entropy):

$$E(S) = -P_+ \log_2 P_+ - P_- \log_2 P_- \qquad (2)$$

式中:P_+ 和 P_- 分别表示实例集中两类实例分布的比例。以表 2 为例,如果以目标字素 v 对应的音素为分类对象,10 个实例可分成 2 类,即 v 类和 f 类,各有 5 个,可表示为[5+,5-],所以说 P_+ 和 P_- 的值均为 0.5。

而在式(1)等号右边的第 2 项,即 $\sum_{v \in V(A)} \frac{|S_v|}{|S|} E(S_v)$ 中:S_v 是实例集 S 中属性 A 的值为 v 的子实例集,$V(A)$ 用于表示实例某个属性 A 的所有值的集合。比如,令 A = P_1,v = N,则 S_v = {I2,I4,I6,I8,I10},而且这 5 个实例均属于 f 类,因而 S_v 可表示为[0+,5-],则 $E(S_v)$ 表示 S_v 中 v 类和 f 类的熵:

$$E(S_v) = E([0+,5-]) = -0 \times \log_2(0) - 1 \times \log_2(1) = 0$$

所以说,式(1)等号右边的第 2 项表示用属性 A 分类实例集 S 熵的期望值。

信息增益函数是决策树每一步生长中确定属性分类效果好坏的判断标准,生长的每一步都必须计算所有属性的信息增益,信息增益最大的属性,就是分类集

合中的所有实例最优的属性[11]。

但仔细分析式(1),以属性 A 和 B 为例,可以看出,他们的熵的值是相等的,不相等的是等号右边的第 2 项。要判断属性 A 和 B 哪一个的分类效果更好,可以直接计算他们信息增益的差值,即:

$$G(S,A)-G(S,B)=\left[E(S)-\sum_{v\in V(A)}\frac{|S_v|}{|S|}E(S_v)\right]-\left[E(S)-\sum_{v\in V(B)}\frac{|S_v|}{|S|}E(S_v)\right]=-\left[\sum_{v\in V(A)}\frac{|S_v|}{|S|}E(S_v)-\sum_{v\in V(B)}\frac{|S_v|}{|S|}E(S_v)\right]$$

比如,令 $A=$G_-1,$v=i$,则 $S_v=\{$I1,I2,I3,I4,I6,I8,I9,I10$\}$,如果假设整个实例库就只有表 1 的 10 个实例,则在 $S_{G_-1=i}$中的 8 个实例中,属于 v 类的有 3 个,属于 f 类有 5 个,表示为 $f_{G\_-1=i}(v)=3$,$f_{G\_-1=i}(f)=5$。于是可以假定在 G_-1 $=i$ 这样的设定条件下 v 类与 f 类分布的比例为:

$$f_{G\_-1=i}(v):f_{G\_-1=i}(f)=3:5$$

则 $E(S_{G_-1=i})$的值为:

$$E(S_{G\_-1=i})=-P_{(i)+}\log_2P_{(i)+}-P_{(i)-}\log_2P_{(i)-}$$

其中:

$$P_{(i)+}=\frac{f_{G\_-1=i}(v)}{f_{G\_-1=i}(v)+f_{G\_-1=i}(f)}=\frac{3}{3+5}=0.375$$

$$P_{(i)-}=\frac{f_{G\_-1=i}(f)}{f_{G\_-1=i}(v)+f_{G\_-1=i}(f)}=\frac{5}{3+5}=0.625$$

于是可以求得:

$$E(S_{G\_-1=i})=-0.375\times\log_2(0.375)-0.625\times\log_2(0.625)\approx0.954$$

1.2.2　歧义类中音素类分布的比例

令:X、Y 表示某个字素对应的两类音素。$f(X)$ 表示在实例库中目标字素转换为 X 的实例出现的次数。$f(Y)$ 表示在实例库中目标字素转换为 Y 的实例出现的次数。则定义歧义类 X_Y 中音素为 X 和 Y 的实例分布的比例为:

$$P(X)=\frac{f(X)}{f(X)+f(Y)}\tag{3}$$

$$P(Y)=\frac{f(Y)}{f(X)+f(Y)}\tag{4}$$

而对于像元音字素 e、o 等对应不止两个音素的情况,即所谓的多类问题,可将实例库中标注为音素 X_i $(i=1,2,\cdots,N)$的实例的比例定义为:

$$P(X_i)=\frac{f(X_i)}{\sum_{j=1}^{N}[f(X_j)]}\tag{5}$$

1.2.3　实例的分类函数

最后定义用某个属性 A 分类整个实例集 S 的分类函数如下:

$$T(S,A)=\sum_{v\in V(A)}\frac{|S_v|}{|S|}E(S_v)\tag{6}$$

式(6)即为决策树生长过程中每一步用于选取最佳分类属性的函数,如果某一步中某个属性 A 的 $T(S,A)$ 值在所有值中最小,则说明使用它分类的效果最好。

以表 2 所示的两类问题 v_f 类为例,假定整个实例集 S 仅由这 10 个实例组成,即 $|S|=10$,令 $X=v$,$Y=f$。如果用属性 G_-1 来分类实例,则 $V(A)=\{i,r\}$,则:

$$T(S,\text{G\_}-1)=(8/10)\times E(S_{G\_-1=i})+(2/10)\times E(S_{G\_-1=r})=0.8\times0.952+0.2\times0\approx0.762$$

1.3　构建决策树

在构建决策树的初始阶段,所有实例组成根结点,计算每个属性的分类函数的值,选取分类函数值最小的属性作为这一步生长的分类属性。如果该属性有 N 个值,则决策树就向下产生 N 个分叉,然后再沿最左侧的分枝继续计算其他属性的值,再选取一个最佳的分类属性,依此类推,当这一条分枝全部分类完成或无法继续生长,则回到上一个未完全分类的结点继续树的生长,直到所有结点均完成分类或者树无法继续向下生长为止。

假设整个实例库只有表 2 的 10 个实例,在根结点处的 4 个属性对应的分类函数的值如下:

T(S,G_-1)=0.762

T(S,G_1)=0

T(S,P_1)=0

T(S,G_2)=1

可以看到,T(S,G_1)和 T(S,P_1)均为 0,是 4 个值中最小的,说明如果用属性 G_1 或 P_1 来分类实例库效果最好。由于这两个属性的分类函数的值均为 0,说明使用这两个属性来分类实例库,可以将所有实例完全分成两类,于是可以生成以下 v_f 类的转换规则:

if G_1 = e then P = v

if G_1 = E then P = f

或

if P_1 = V then P = v

if P_1 = N then P = f

当然在实际的决策树构建的过程中,不可能像表 1 这样简单,这里只是为了方便算法说明,故意选取了 10 个典型的实例。

2 规则测试和修剪

为了转换规则的学习和测试,本文采用 K 次迭代交叉验证法(K-Fold Cross Validation)[12-13]来进行测试。将预先创建的字素音素平行语料库中的 10 300 个样本平均分成 10 份,每份 1 030 个样本,需进行 10 轮规则验证,在每一轮验证中,将 9 份共 9 270 个样本用于规则学习,余下的 10% 即 1 030 个用于规则测试。10 轮交叉验证的平均结果如表 4 所示。

表 4 字素音素转换规则 10 轮交叉验证结果

学习		测试	
平均单词数	平均规则数	平均单词数	平均字素音素转换正确率
9 270	1 948	1 030	83.56%

可以看出,平均的字素音素转换正确率只有 83.56%,非常不理想。为了进一步提高转换正确率,需要采取以下改进措施:

1) *对单词进行词法预处理* 德语是一个构词功能非常强大的语言,比如说,数字 20 0433 对应的德语单词为 zweihunderttausendvierhunderdreiunddreiβig,很明显,它是由 8 个单词(zwei、hundert、tausend、vier、hundert、drei、und、dreiβig)组成的复合词,这么长的一个单词直接对其进行字素音素转换,出错的概率显然要比单独对 8 个单词分别进行字素音素转换要高。因而,对一些单词先基于构词法进行切分,然后再进行字素音素转换应该是提高转换正确率的一个可靠手段。

德语主要的构词形式有两类[14]:

(1) *复合词切分* 复合词是指由两个或多个词组成的新词,德语中的复合词非常多,只要符合一定的构词规则,就可以任意地构成复合词。如 Wortschatz(复合名词,可切分成 wort-schatz)、danksagen(复合动词,可切分成 dank-sagen)、kinderreich(复合形容词,可切分成 kinder-reich)等。

(2) *派生词* 所谓派生词,是指由一个现有的词汇(或词干)加特定前缀或后缀组成的新的词汇,如 ansprechen、arbeitslos 等。派生词的切分就是将派生词切分成词缀加另一个基本词汇(或词干),ansprechen 切分成 an-prech-en,freundlich 切分成 freund-lich,由于词缀的数量有限,其转换比较简单,只需将主要精力放在词干的转换上就可以了。

经过预处理后,再次使用决策树进行转换学习,其 10 轮的平均转换正确率可大幅提高到 94.67%,提高了大约 11%。

2) *规则修剪* 在决策树学习过程中很可能会出现过度拟合(Overfit)现象,从而导致经学习生成的规则在实际标注时的效果不佳。可以通过“规则后修剪法”[11]来解决这一问题。规则后修剪法包含两步:

(1) *先修剪每一条规则* 比如,以下是 v_f 类的一条规则:

if G_1 = e and P_1 = V then P = v

该规则有两个前提条件,即:“G_1 = e”和“P_1 = V”,所谓规则修剪,其实就是剔除其中的某个前提条件。上述规则如果分别剔除一个前提条件,就可以形成两条新规则:

if G_1 = e then P = v

if P_1 = V then P = v

接着就来判断原规则及经修剪后产生的两条新规则中究竟哪一个的分类效果最佳。将这 3 条规则分别进行测试,并计算它们的得分。计算方法为:如果某个规则将某个 v_f 类转换正确,得一分,反之扣一分。最后得分最高的规则胜出,替换其他规则。这 3 条规则的平均得分分别为 126.6、117.2 及 234.5,规则“if P_1 = V then P = v”的得分最高,其胜出,其他两条规则被抛弃。

(2) *规则排序* 将修剪后胜出的所有规则按照得分的高低降序排序,生成规则库,这样就保证转换效果好的规则先于效果差的规则应用于转换。

再次使用新生成的规则库进行 10 轮交叉测试,平均转换正确率进一步提高到 95.36%。

3) *将外来词另行处理* 通过分析错误样本,发现有一些转换错误的词是外来词,主要是一些来源自英语和法语的词,如 Internet、Computer、Restaurant 等。由于这些外来词的读音规则与德语单词的读音规则差别较大,且在德语中所占比例不大,因而没有必要在本课题中花费大量的时间专门研究它们的字素音素转换规则。最简单的办法就是,直接创建一个德语中常用外来词的语音库来解决这部分词的语音合成问题。10 300 中共有 237 个外来词,如果将这些外来词剔除,则整个转换正确率可提高到 98.03%。

3 结 语

由于德语的构词功能极为强大,加之随着语言学的发展,新的词汇不断涌现,因而不可能建立一个包含所有词汇的词库。如何取得那些不在词库的词的读音就必须借助字素音素转换技术。本文提出了一种基于

决策树的字素音素转换算法。通过对规则进行交叉验证,初期的转换正确率只有 83.56%。但如果预先对单词进行词汇预处理后再进行测试,则其转换正确率可大幅提高到 94.67%;如果再通过规则修剪和排序,则转换正确率可以进一步提高到 95.36%;如果再将所有外来词另行建库处理,只针对纯正的德语单词应用规则进行转换,则其正确率可达到 98.03%。

参 考 文 献

[1] Demberg V. Letter-to-Phoneme Conversion for a German Text-to-Speech System[D]. Universität Stuttgart, Institut für Maschinelle Sprachverarbeitung, 2006.

[2] Sayeski K L, Earle G A, Eslinger R P, et al. Teacher candidates' mastery of phoneme-grapheme correspondence: massed versus distributed practice in teacher education[J]. Annals of Dyslexia, 2017, 67 (1):26 - 41.

[3] 李媛. 德语语音教程[M]. 上海:上海外语教育出版社, 2014.

[4] Möbius B. German and Multilingual Speech Synthesis[J]. AIMS, 2001,7(4).

[5] Rao K, Peng F, Sak H, et al. Grapheme-to-phoneme conversion using Long Short-Term Memory recurrent neural networks[C]//IEEE International Conference on Acoustics. IEEE, 2015:4225 - 4229.

[6] Seng K, Katsurada K, Iribe Y, et al. Solving the Phoneme Conflict in Grapheme-to-Phoneme Conversion Using a Two-Stage Neural Network-Based Approach[J]. Ieice Transactions on Information & Systems, 2014, 97-D(4):901 - 910.

[7] Novak J R, Minematsu N, Hirose K. Phonetisaurus: Exploring grapheme-to-phoneme conversion with joint n-gram models in the WFST framework[J]. Natural Language Engineering, 2016, 22(6):907 - 938.

[8] Novak J, Minematsu M, Hirose K. Failure transitions for Joint n-gram models and G2P conversion[C]//Proceedings of InterSpeech 2013, Lyon, France, 2013: 1821.

[9] Rasipuram R, Magimai-Doss M. Acoustic Data-driven Grapheme-to-Phoneme Conversion using KL-HMM[C]// 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2012:4841 - 4844.

[10] Marchand Y, Damper R I. Can syllabification improve pronunciation by analogy of English? [J]. Natural Language Engineering, 2007,13(1):1 - 24.

[11] 孙亮, 黄倩. 实用机器学习[M]. 北京:人民邮电出版社, 2017.

[12] Yadav S, Shukla S. Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification[C]//IEEE International Conference on Advanced Computing. IEEE, 2016:78 - 83.

[13] Rodriguez J D, Perez A, Lozano J A. Sensitivity Analysis of k-Fold Cross Validation in Prediction Error Estimation[J]. IEEE Trans Pattern Anal Mach Intell, 2010, 32(3):569 - 575.

[14] 江山,龚剑琴. 形态学下的德语构词法研究[J]. 科技信息, 2010 (8):135 - 136.

(上接第 179 页)

多数据融合分析功能可以直观地反映大范围区域内适应工程施工小范围区域,帮助施工决策人员合理规划施工区域。帮助决策者有侧重地进行决策选择部署,合理规划施工区域,提高野外踏勘的目的性并减少踏勘次数。

6 结 语

本系统针对荒漠复杂地形条件下施工受限条件多、踏勘成本大、施工决策难等问题,将 GIS 技术与工程施工规划分析相结合,使施工相关数据可视化,是对传统施工规划方法的完善和提高。本文除了实现 GIS 系统中通用的管理、分析、查询显示、输出等功能外,还设计实现了平整度分析及多数据融合施工规划分析功能,通过多数据融合分析出大范围内适合多工程施工的小范围区域,对于工程的规划有着重要的参考作用。

参 考 文 献

[1] 陈元涛. 基于 GIS 的道路智能化选线方法研究[D]. 重庆:重庆交通大学,2012.

[2] 韩英, 赵宇鹏. GIS 地理信息系统的应用及其发展分析[J]. 科技创新与应用, 2013(35):73 - 73.

[3] 刘学军, 龚健雅, 周启鸣,等. 基于 DEM 坡度坡向算法精度的分析研究[J]. 测绘学报, 2004, 33(3):258 - 263.

[4] 沈杰杰, 刘维. 公路工程试验检测技术[M]. 济南:山东大学出版社, 2012.

[5] 牟乃夏. ArcGIS Engine 地理信息系统开发教程——基于 C#. NET[M]. 北京:测绘出版社,2015.

[6] 芮小平. 基于 C#语言的 ArcGIS Engine 开发基础与技巧[M]. 北京:电子工业出版社,2015.

[7] 崔铁军. 地理空间数据库原理[M]. 北京:科学出版社,2010.

[8] 徐仕琪,张晓帆,周可法,等. 关于利用七参数法进行 WGS-84 和 BJ-54 坐标转换问题的探讨[J]. 测绘与空间地理信息,2007,30(5): 33 - 42.