

## 语音合成中的文本归一化问题

冯志伟

(教育部语言文字应用研究所, 北京 100010)

**摘要:** 文本归一化是语音合成中一个与语言规划联系最为密切的问题。英语文本归一化的3个任务是: 句子的词例还原, 非标准词的处理, 同形异义词的排歧。根据汉语语音合成的实际, 汉语文本归一化应当解决的问题是: 汉语文本的词例还原, 汉语文本非标准词的处理, 汉语文本同形异义词的排歧。此外, 还应注意汉语语音合成中特殊韵律现象的处理。

**关键词:** 语音合成; 词例还原; 非标准词; 同形异义词; 排歧; 韵律。

### 一、语音合成技术的发展与应用

出生在斯洛伐克(当时属于匈牙利王国)的发明家 Wolfgang von Kempelen 于1769年在维也纳为玛利亚·泰莱撒女皇(Maria Theresa)制造了一个叫做图尔克的机器(mechanical Turk)。图尔克机是一个会下象棋的自动机器, 它的前端是一个布满了齿轮的大木箱, 在这个大木箱的后面, 坐着一个机器人, 这个机器人在下象棋的时候, 会用自己的机械手来移动棋子。数十年间, 这个图尔克机在欧洲和美国进行巡回比赛, 据说曾经打败了法国皇帝拿破仑(Napoleon Bonaparte), 甚至还和英国数学家巴贝奇(Charles Babbage)做过对弈, 名噪一时。

但是, 后来发现, 这尽然是一场恶作剧。原来这个图尔克机的全部动作都是由藏在木箱内部的一个会下象棋的活生生的人控制着的。不然, 这个图尔克机也许可以看成是人工智能的最早的一个成就呢!

Wolfgang von Kempelen 因此而声名狼藉, 不过, 我们也不能把他看扁了。这个 Wolfgang von Kempelen 确实具有发明的天分。在1769年至1790年间, 他还做了另外一件举世瞩目的大事: 他发明了第一台能够合成完整句子的语音合成器。他的这个装置包括一个模拟肺部的鼓风机, 一个橡胶制成的嘴, 一个鼻子孔, 一个模拟声带的簧片, 用于产生摩擦音的各种不同的哨子, 以及用于给塞音提供喷出气流的一个附加的小鼓风机。这种语音合成器实际上是一个共鸣箱。操作

员用双手移动操作杆来打开或关闭鼻子孔, 调节有弹性的皮制“声腔”, 就可以产生各种不同的元音和辅音。由于受当时技术水平的限制, Wolfgang von Kempelen 发明的这台语音合成器是用木头或皮革来制造的。材料虽然还比较简陋, 却开了语音合成这项技术的先河。<sup>①</sup>

两百多年过去之后, 我们不再使用木头或皮革来制造语音合成器了, 我们也不再需要人来亲自担任操作员了。现代语音合成(speech synthesis)的任务就是使用计算机从文本产生语音, 把可视的书面文本转换成可听的语音, 所以, 语音识别又叫做“文本—语音转换”(text-to-speech conversion)或简称“文语转换”(TTS)。这样的语音合成是用计算机来进行的, 与 Wolfgang von Kempelen 的语音合成器不可同日而语。

美国哈斯金(Hanskins)实验室、贝尔实验室、麻省理工学院、剑桥空军研究实验室、瑞典斯德哥尔摩皇家工学院、德国夫琅禾费研究院、中国科学技术大学都进行过语音合成的研究。语音合成已经进入实用化阶段。<sup>②③</sup>

现代语音合成有着多种多样的、非常广泛的用途。

首先, 语音合成器可以用于基于电话的会话智能代理系统(conversation agent system)中, 这种智能代理可以与人进行对话和交谈。目前国外的会话智能代理系统已经实用化了。

其次, 语音合成器还可以在那些不是会话的场合用来对人说话。例如, 用语音合成器来给盲

人大声朗读；将语音合成技术应用于视频游戏和儿童玩具的制作等。

此外，语音合成技术还可以用于帮助那些神经受损的病人说话。例如，英国著名天体物理学家 Steven Hawking 由于得了肌萎缩性脊髓侧索硬化症（ALS）而失去了使用自己语音的能力，现代语音合成技术给他帮了大忙，使他可以通过电脑打字把要说的词语传输给语音合成器，并让语音合成器说出单词的方式来进行说话。

目前，最先进的语音合成系统可以在各种不同的输入环境下产生优质的自然语音，尽管哪怕最好的系统产生出来的声音还显得有些呆板，并且只能局限于它们所使用的那些语音的范围之内。

笔者几年前患了黄斑前膜的眼病，双目视物不清，读书非常困难。2005年，我借助于英语和汉语的语音合成器让计算机给我朗读书面文字，克服了看不清书面文字的困难，完成了 588 页的《自然语言处理综论》的长篇巨著的英译汉翻译工作，中文译本已经由电子工业出版社正式出版了。

可见，现代语音合成技术确实给我们人类的生活带来了福音！

目前，语音合成技术已经走进了普通人的日常生活。在很多手机中，都有语音合成装置，可以正确地朗读出手机上的短信。

## 二、语音合成技术之文本归一化

语音合成的任务是把文本映射为波形。例如，我们有如下的文本：

PG&E will file schedules on April 20.

语音合成器要把这个文本映射为如下的波形：



语音合成器把这样的映射分为两个步骤来实现：首先把输入文本转换成语音内部表示（phonemic internal representation），然后再把这个语音内部表示转换成波形。我们把第一个步骤叫做文本分析（text analysis），把第二个步骤叫做波形合成（waveform synthesis）。本文只讨论

文本分析问题。文本分析中最重要的问题是文本归一化（text normalization），因此，我们着重讨论文本归一化的问题。

在语音合成中，为了生成语音的内部表示，首先我们必须对于形形色色的、自然状态的文本做前处理（pre-processing）或归一化（normalization）。我们需要把输入的文本分解为句子，处理缩写词、数字等特殊问题<sup>④⑤</sup>。

目前，英语的文本归一化研究已经取得不少成果。这里，我们首先主要根据国外文献<sup>②</sup>介绍英语文本归一化研究的成果，以此来加深对文本归一化的认识，然后再进一步结合汉语的特点，讨论汉语的文本归一化问题。

英语的文本归一化有 3 个任务：第一个任务是句子的词例还原，第二个任务是非标准词的处理，第三个任务是同形异义词的排歧。

“词例还原”（tokenization）这个术语，可能有的读者还不熟悉，我们有必要先解释一下。

“词例”（token）是文本中独立的词汇单元。所谓“词例还原”，就是自动地把句子中的单词作为独立的词例切分出来。英语文本中的单词一般是界限分明的，单词与单词之间存在空白，单词的切分不像汉语书面文本那样困难。但是，下列情况仍需要进行切分，把独立的“词例”找出来。

### 1. 缩写

（1）缩写式“字母 + 圆点 + 字母 + 圆点”算一个词例。例如，“U.S.”，“i.e.”，“U.K.”，都算一个词例。

（2）缩写式“字母串 + 圆点”算一个词例。例如，“Mr.”，“Mrs.”，“Eds.”，“Prof.”，“Dr.”，“Co.”，“Jan.”，“A.”，“b.”等，都算一个词例。

（3）含有非字母符号的缩写算一个词例。例如，“AT&T”，“Micro\$oft”，都算一个词例。

### 2. 连续的数字

一般连续的数字按一个词例来处理。例如，“123,456.78”是一个独立的词例。带百分号的数字如“90.7%”，也应该算一个独立的词例。分数如“3/8”算一个独立的词例。日期如“15/04/1939”也算一个独立的词例。

### 3. 其他情况

（1）带连字符的词串算一个词例。例如，“three-years-old”，“one-third”，“so-called”，都

算一个词例。

(2) 带空白的某些习用符号串算一个词例。例如, “and so on”, “ad hoc”, 都算一个词例。

(3) 带省略符号 “'” 的符号串, 要还原成不同的词例。例如:

Let's 还原成 let + us;

I'm 还原成 I + am;

it's、that's、this's、there's、what's、where's……均还原成 (it、that……) + is;

He's 还原成 (He + is) 或者 (He + has)。

经过词例还原之后, 句子中的符号串被转换成词例串。这样, 就为波形合成提供了方便。

下面的英语文本是从 Enron 语料库中抽取出来的, 我们来考察一下这个文本在处理上的困难究竟有多大。

He said the increase in credit limits helped B.C. Hydro achieve record net income of about \$1 billion during the year ending March 31. This figure does not include any write-downs that may occur if Powerex determines that any of its customer accounts are not collectible. Cousins, however, was insistent that all debts will be collected: “We continue to pursue monies owing and we expect to be paid for electricity we have sold.”

为了把上面这个文本的片段切分成彼此分开的话段以便语音合成, 我们需要知道: 第一个句子是在 March 31 后面的那个小圆点处结尾, 而不是在 B.C 后面的小圆点处结尾。因此, March 31 后面的那个小圆点要还原成句号, 单独切分出来, 而 B.C 后面的小圆点不能单独切分, 应当把 “B.C.” 作为一个单独的词例。我们还需要知道: 在单词 collected 处是一个句子的结尾, 尽管 collected 后面的标点符号是一个冒号, 而不是小圆点。因此, 这个冒号应当为一个单独的词例。

这些研究工作的目的是找出句子中的“词例”(token), 所以叫做“词例还原”(tokenization)。英语文本归一化的第一个任务就是句子的词例还原(sentence tokenization)。

英语文本归一化的第二个任务是处理非标准词(non-standard words)。非标准词是指那些在标准的发音词典(pronunciation dictionary)中没有

收录的单词, 包括数字、首字母缩写词、普通缩写词等等, 由于这些非标准词的数量几乎是无限的, 发音也没有明确的标准, 因而, 在标准的发音词典中难以注明它们的准确发音。例如, March 31 的发音应当是 March thirty-first, 而不是 March three one; \$1 billion 的发音应当是 one billion dollars, 在 billion 的后面应当加一个单词 dollars。它们都没有按照英语的一般习惯来发音, 需要特殊对待。

此外, 英语文本归一化还要研究同形异义词的排歧(homograph disambiguation)问题。

下面, 我们分别讨论英语文本归一化中的这些问题。

### 三、英语语音合成中的文本归一化问题

#### 1. 句子的词例还原

从前面所举的例子来看, 英语句子的词例还原是有一定难度的。英语句子的边界不总是用小圆点来标识的, 有时也可以用冒号等标点符号来标识。当以一个缩写词来结束句子的时候, 还会出现一个附带的问题, 这时, 缩写词结尾处的小圆点会起双重的作用。例如, 在句子 “The group included Dr. J. M. Freeman and T. Boone Pickens Jr.” 中, “Jr.” 最后的小圆点, 既可以表示 Junior 的缩写 (T. Boone Pickens Jr. 表示 “小 T. Boone Pickens”), 又可以表示句末的句号。这个小圆点产生了歧义。

英语句子的词例还原的一个关键部分就是小圆点的排歧问题。大多数英语句子词例还原的算法都比确定性算法(deterministic algorithm)要更加复杂一些, 特别是这些算法都是通过机器学习(machine learning)的方法来训练, 而不是用手工建立的。在进行这样的训练时, 我们首先要手工标注带有句子边界的一个训练集, 然后使用任何一种有指导的机器学习方法(supervised machine learning)训练一个分类器(classifier)来判定并标注句子的边界。

更加具体地说, 在开始的时候, 我们可以把输入文本还原成彼此之间用空白分隔开的词例, 然后, 选择包含 “!”、“.” 或者 “?” 三个符号中的任何一个符号(也可能包含 “:”)的词例作为句子的结尾。在手工标注了一个包含这样的词例

的语料库之后，我们就训练一个分类器，对于这些词例内的潜在句子边界字符，进行二元判定，判定某个词例是 EOS (end-of-sentence, 句子结尾)，还是 not-EOS (非句子结尾)。

这种分类器成功与否依赖于在分类时抽出的特征。

让我们来研究在给句子边界排歧的时候可能用得着的某些特征模板，其中的句子边界符号 **candidate** (候选成分) 表示在我们训练的少量数据中可能标注为句子边界的某个符号。

**Prefix**: 前缀 (处于 **candidate** 之前的候选词例部分);

**Suffix**: 后缀 (处于 **candidate** 之后的候选词例部分);

**PrefixAbbreviation** 或 **SuffixAbbreviation**: 前缀或后缀是不是 (一串符号中的) 缩写词;

**PreviousWord**: 处于 **candidate** 之前的单词;

**NextWord**: 处于 **candidate** 之后的单词;

**PreviousWordAbbreviation**: 处于 **candidate** 之前的单词是不是一个缩写词;

**NextWordAbbreviation**: 处于 **candidate** 之后的单词是不是一个缩写词。

我们来研究下面的例子:

ANLP Corp. chairman Dr. Smith resighed.

对照上面的特征模板，在 “Corp.” 中的 “.” 的特征值是:

PreviousWord = ANLP

NextWord = chairman

Prefix = Corp

Suffix = NULL

PreviousWordAbbreviation = 1

NextWordAbbreviation = 0

如果我们的训练集足够大，那么，我们也可以找到一些关于句子边界的词汇方面的线索。例如，某些单词可能倾向于出现在句子的开头，某些单词可能倾向于出现在句子的结尾。这样，我们又可以加进去如下的特征:

**Probability[candidate occurs at end of sentence]**: 表示 **candidate** 出现于句子结尾的概率;

**Probability[word following candidate occurs at beginning of sentence]**: 表示跟随在出现于句子开头的 **candidate** 的单词的概率。

上面所述的特征，大部分是与具体的语言无关的。此外，我们还可以使用一些针对具体语言的特征。例如，在英语中，句子一般是以大写字母开头的，所以，我们还可以使用如下的特征:

**Case of candidate**: **candidate** 的大小写情况。如: Upper、Lower、Allcap、Numbers。

**Case of word following candidate**: 跟随在 **candidate** 后面的单词的大小写情况。如: Upper、Lower、Allcap、Numbers。

类似地，我们还可以使用缩写词的某些次类的信息。例如，尊称或头衔 (Dr.、Mr.、Gen.)，公司名称 (Corp.、Inc.)，月份名称 (Jan.、Feb.)，等等。

任何的机器学习方法都可以用来训练 EOS 分类器。逻辑回归 (logical regression) 和决策树 (decision tree) 是两种最普通的方法。逻辑回归的精确度比决策树的精确度要高一些。

## 2. 非标准词的归一化

英语文本归一化的第二个任务是“非标准词” (non-standard words, 简称 NSWs) 的归一化。非标准词是诸如数字或缩写词之类的词例。在英语中专有名词的读音很特别，词典中一般查不出来，也可以算为非标准词。在语音合成中，在计算机读出它们之前，需要把它们扩充为英语单词的序列。

英语非标准词的处理是很困难的，因为它们总是在读音方面存在歧义。例如，在不同的上下文中，1750 这个数字至少可以有 4 种不同的读法:

**Seventeen fifty**: (在 “The European economy in 1750” 中);

**One-seven-five-zero**: (在 “The password is 1750” 中);

**Seventeen hundred and fifty**: (在 “1750 dollars” 中);

**One thousand, seven hundred, and fifty**: (在 “1750 dollars” 中)。

相似的歧义问题也发生在罗马数字 IV 或 2/3 等非标准词的读音中。

IV 可以读音为 four, 或者读为 fourth, 或者也可以按照字母 I 和 V 分别来读, 这时, IV 的含义是 “intravenous” (静脉内的)。

2/3 可以读为 two thirds, 或者读为 February

third, 或者读为 March second, 或者读为 two slash three.

某些非标准词是由字母构成的, 例如, 缩写词 (abbreviation), 字母序列 (letter sequences), 首字母缩写词 (acronyms) 等。

缩写词读音时, 一般都要进行扩充 (expanded), 所以, Wed 要读为 Wednesday, Jan 1 要读为 January first。像 UN、DVD、PC、IBM 这样的字母序列 (letter sequences) 读音时, 要按照字母在序列中的顺序, 一个一个地来读。像 IKEA、MoMA、NASA 和 UNICEF 这样的首字母缩写词读音时, 要把它们当做一个单词来读。这里也会出现歧义问题。Jan 按照一个单词来读音呢 (人名 Jan)? 还是扩充为月份名称 January 来读音? 这常常会为我们陷入举棋不定的困境。

我们可以把英语中数字和字母组成的非标准词归纳为字母非标准词和数字非标准词两大类, 每一个大类又可以进一步细分为若干个小类。

#### (1) 字母非标准词

① EXPN (Abbreviation, 缩写词)。如: adv、N.Y.、mph、gov't。

② LSEQ (Letter sequence, 字母序列)。如: DVD、D.C.、PC、UN、IBM。

③ ASWD (Read as word, 按一个单词读音)。如: IKEA, 未知词, 专有名词。

#### (2) 数字非标准词

① NUM (Number cardinal, 基数词)。如: 12、45、1/2、0.6。

② NORD (Number ordinal, 序数词)。如: May 7、3<sup>rd</sup>、Bill、Gates III。

③ NTEL (Telephone or part of telephone, 电话号码或电话号码的一部分)。如: 212-555-5423。

④ NDIG (Number as digit, 数字号码)。如: Room 101。

⑤ NIDE (Identifier, 识别号码)。如: 747、386、15、pc110、3A。

⑥ NADDR (Number as street address, 街道地址号码)。如: 747、386、15、pc110、3A。

⑦ NZIP (Zip code or BO Box, 邮政编码或信箱号码)。如: 91020。

⑧ NTIME (Time, 时间)。如: 3.20、11: 45。

⑨ NDATE (Date, 日期)。如: 2/28/05、

28/02/05。

⑩ NYER (Years, 年代)。如: 1988、80s、1900s、2008。

⑪ MONEY (Money, US or other, 美元或其他货币)。如: \$3.45、HK\$300、Y20,200、\$200K。

⑫ BMONEY (Money tr/m/billions, 万亿/百万/十亿的货币)。如: \$3.45 billion。

⑬ PRCT (Percentage, 百分比)。如: 75%、3.4%。

每种类型非标准词都有一个或几个特定的实际读法。例如, 年代 (NYER) 通常按“双对式读法” (paired method) 来读, 其中每一对数字按照一个整数来读音 (例如, 1750 读为 seventeen fifty); 而美国的邮政编码 (NZIP) 通常按“顺序式读法” (serial method) 来读, 序列中的每一个数字单独读音 (例如, 94110 读为 nine-four-one-one-zero)。货币 (BMONEY) 这种类型的读法要处理一些特异的表达形式。如 \$3.2 billion 在读音的时候要在结尾加一个单词 dollars, 读为 three point two billion dollars。对于字母非标准词的读法, 我们有 EXPN、LSEQ 和 ASWD 等类型。EXPN 用于诸如“N.Y.”这样的缩写词, 读的时候要进行扩充; LSEQ 用于读那些要按照字母序列来读音的首字母缩写词; ASWD 用于读那些要按照单词来读音的首字母缩写词。

非标准词的处理至少有三个步骤: 词例还原 (tokenization), 分类 (classification), 扩充 (expansion)。词例还原用于分割和识别潜在的非标准词; 分类用于给非标准词标上面所述的那些读音类型; 扩充用于把每一个类型的非标准词转换为标准词的符号串。

在词例还原这个步骤, 我们可以使用空白把输入文本还原成词例, 在词例与词例之间用空白分开, 然后假定在发音词典中没有的单词都是非标准词。一些更加细致的词例还原算法还可以处理某些词典中业已包含某些缩写词这样的事实。例如, CMU 发音词典就包含了缩写词 st、mr、mrs 的发音 (尽管这些发音不正确) 以及诸如 mon、tues、nov、dec 等日期和月份的缩写词。因此, 除了那些没有看到的单词之外, 我们还有必要给首字母缩写词标注发音, 并把单字母的词

例作为潜在的非标准词来处理。词例还原算法还需要对于那些包含两个词例的组合分隔成不同的单词，如 2-car 或 R Ving 等。我们可以使用简单的启发式推理方法来分隔单词。例如，把破折号作为分割的标志，把大写字母与小写字母转换之处作为分割的标志，等等。

下一个步骤是分类，也就是标注非标准词的类型。使用简单的正则表达式就可以探测出很多非标准词的类型。如 NYER 可以使用如下的正则表达式来探测：

`/(1[89][0-9][0-9])|(20[0-9][0-9])/`

其他类型的规则写起来比较困难，所以，使用带有很多特征的机器学习分类器来进行分类将会更加有效。

为了区分字母非标准词 ASWD、LSEQ 和 EXPN 等不同的类型，我们可以使用组成成分的字母的一些特征。我们在这里举例简单地说一说：全是大写字母的单词（如 IBM、US）可以归入 LSEQ 这一类，带有单引号的全是小写字母组成的一些比较长的单词（如 gov't、cap'n）可以归入 EXPN 这一类，带有多个元音的全是大写字母组成的单词（NASA、IKEA）可以归入 ASWD 这一类。

另外一个很有用的特征是相邻单词的辨识。像 3/4 这样的歧义字符串，它可以归入 NUM（three-fourths）或者归入 NDATE（march third）。归入 NDATE 时，它的前面可能出现单词 on，后面可能出现单词 of，或者在附近的上下文某个地方出现单词 Monday。与此不同，归入 NUM 时，它的前面可能是另外一些数字，后面可能出现诸如 mile 和 inch 之类的表示计量单位的单词。类似地，VII 这样的罗马数字，当前面出现 Chapter、part 或者 Act 等单词时，可能倾向于归入 NORD（seven），当在相邻单词中出现 king 或者 Pape 之类的单词时，就可能倾向于归入 NUM（seventh）。这些上下文单词可以通过手工的方式选择作为特征，也可以通过诸如决策表（decision list）算法这样的机器学习技术选择作为特征。

如果把上述的各种办法结合起来，建立一个机器学习的分类器，这样就能大大地提高分类的效能。例如，Sproat 等的非标准词分类器（NSW

classifier）使用了 136 个特征<sup>⑤</sup>，其中包括诸如“全是大写字母”，“含有两个元音”，“含有斜线号”，“词例长度”等基于字母的特征，还包括诸如 Chapter、on、king 等特殊的单词是否在附近的上下文中出现二元特征。Sproat 等还提出了一个基于规则的粗分类器（rough-draft classifier）<sup>⑤</sup>，其中使用手写的正则表达式来给很多表示数字的非标准词分类。这个粗分类器的输出可以在主分类器（main classifier）中作为另外的特征来使用。

为了建立这样的主分类器，我们需要一个手工标注的训练集，其中的每一个词例都标出它们的非标准词分类范畴。Sproat 等就建立了一个这样的手工标注数据库<sup>⑤</sup>。给出了标注训练集，我们就可以使用任何一种有监督的机器学习算法，例如前面讨论过的逻辑回归算法、决策树算法等。然后，我们训练分类器来使用这些特征，从而预测手工标注的非标准词的分类范畴。

非标准词处理的第三个步骤是把非标准词扩充为一般的单词。EXPN 这种非标准词的类型扩充起来是非常困难的。EXPN 这种类型包括缩写词和像 NY 这样的首字母缩写词。一般地说，扩充时需要借助于缩写词词典，并且要使用下一节将要讨论的同音异义词的排歧算法来处理歧义问题。

其他的非标准词类型的扩充一般都是确定性的。很多的扩充都是简单易行的。例如，LSEQ 把非标准词中的每一个字母扩充为单词序列；ASWD 把非标准词读为一个单词，等于把非标准词扩充为它自己；NUM 把数字扩充为表示基数词的单词序列；NORD 把数字扩充为表示序数词的单词序列；NDIG 和 NZIP 都分别把数字扩充为相应的单词序列。

其他类型的扩充要稍微复杂一些。NYER 把年代按两对数字来扩充，如果年代以 00 结尾，那么，年代的 4 个数字则按照基数词来读音（2000 读为 two thousand），或者按照“百位式读法”（hundreds method）来读音（1800 读为 eighteen hundred）。NTEL 把电话号码扩充为数字序列；也可以把电话号码的最后 4 个数字按照“双对式数字读法”（paired digit）来读音，每一对数字读为一个整数。电话号码还可以采用所谓的“跟踪单位读法”（trailing unit）来读音，以若干个零为

结尾的数字，非零的数字部分按顺序式读法来读音，零的部分按适当的进位制来读音。如，876-5000 的读音为“eight-seven-six five thousand”。

当然，这些扩充很多是与方言有关的。在澳大利亚的英语中，电话号码 33 这个数字序列通常读为 double three。在其他语言中，非标准词的归一化会出现一些特殊的困难问题。例如，在法语或德语中，除了上述的情况之外，归一化还与语言的形态性质有关。在法语中，1 fille（一个姑娘）这个短语归一化为 une fille，而 1 garçon（一个小伙子）这个短语却归一化为 un garçon。与此类似，在德语中，由于名词的格的不同，Heinrich IV（亨利四世）这个短语可以分别归一化为 Heinrich der Vierte, Heinrich des Vierten, Heinrich dem Vierten, 或者 Heinrich den Vierten 等。<sup>④</sup>

英语中的专有名词也属于非标准词。由于英语的发音词典中通常不收专有名词。在很多实际的应用中，这是一个很严重的问题。专有名词包括人名（人的名字和人的姓氏）、地理名称（城市名、街道名和其他的地名）和商业机构名称等。

我们这里仅考虑人名，Spiegel 估计，仅在美国，大约有 200 万个不同的姓氏和 10 万个名字。200 万是一个非常巨大的数字。正是由于这样的原因，大规模的语音合成系统都包含一部很大的专有名词的发音词典。<sup>[1] 175-178</sup>

究竟需要多少个专有名词才算足够呢？

Liberman 和 Church 公布了一个专有名词的词表，包含 1987 年从 Donnelly 市场组织收集的 150 万个专有名词（覆盖了美国的 7 千 2 百万个家庭）。<sup>④</sup>

他们发现，在容量为 4 千 4 百万单词的 AP newswire 语料库中，包含 5 万个专有名词的词典覆盖专有名词的词例数可以达到 70%。有趣的是，很多不包含在词典中的其他专有名词可以通过简单地修改这 5 万个专有名词而得到。例如，给词典中的专有名词 Walter 或 Lucas 加上带中重音的后缀，就可以得到新的专有名词 Walters 或 Lucasville。其他的发音还可以通过韵律类推的方法得到。例如，如果我们知道人名 Trotsky 的发音，而不知道人名 Plotsky 的发音，我们用词首的 /pl/ 来替换 Trotsky 词首的 /tr/，就可以得到

Plotsky 的发音。<sup>④</sup>

诸如此类的技术，包括形态分解、类推替换、以及把未知的专有名词映射到已经存储在词典中的拼写变体的技术（Fackrell 和 Skut, 2004），已经在专有名词的发音研究中取得了一定的成绩<sup>⑥</sup>。但是，总的说来，专有名词的发音仍然是一个困难的问题。

### 3. 同形异义词的排歧

前文所述的非标准词处理算法的目的在于给每一个非标准词 (NSW) 确定一个标准词的序列，以便把它们读出来。然而，有的时候，尽管是一个标准词，要想确定它的读音仍然还是非常困难的事情。同形异义词 (homograph) 的情况就是如此。同形异义词是拼写相同而读音不同的词。下面是英语同形异义词 use、live 和 bass 的几个例子：

It's no use (/y uw s/) to ask to use (/y uw z/) the telephone.

Do you live (/l ih v/) near a zoo with live (/l ay v/) animals?

I prefer bass (/b ae s/) fishing to playing the bass (/b ey s/) guitar.

为了发文排版的方便，我们这里没有采用国际音标 IPA 而采用了 ARPabet，这是目前计算语言学中经常使用的一种非常先进的标音方法，与 ASCII 码完全兼容，便于计算机进行信息交换，国内语言学界还不熟悉，关于 ARPabet 的详细介绍，可参看冯志伟和孙乐译的《自然语言处理综论》(2005)。

法语中的 fils 是同形异义词，含义为“儿子”时，读为 [fis]，含义为“线绳”时，读为 [fil]。法语的 fier 和 est 有多个发音，fier 的含义为“骄傲”或“信赖”时，发音不同；est 的含义为“是”或“东方”时，发音也各不相同。

幸运的是，同形异义词的排歧可以利用词类信息。在英语（以及法语、德语这些类似的语言）中，同形异义词的两个不同的意义往往倾向于分属不同的词类。例如，上例中 use 的两个形式分别属于名词和动词，live 的两个形式分别属于动词和名词。

Liberman 和 Church 说明，在 AP newswire 语料库的 4 千 4 百万单词中，出现频度最高的同

形异义词都可以使用词类信息来排歧。他们用来排歧的 15 个频度最高的单词是 use, increase, close, record, house, contract, lead, live, lives, protest, survey, project, separate, present, read<sup>④</sup>。

由于词类知识已经足够处理很多同形异义词的排歧问题，所以，在实际应用中，我们对标有词类信息的这些同形异义词存储不同的发音，以便进行同形异义词的排歧，然后，对上下文中给定的同形异义词，运行词类标注程序来选择正确的读音。

然而，还有一些同形异义词的不同发音词类相同。在上面的例子中，bass 有两个不同的发音 /b æ s/ 和 /b eɪ s/，它们都属于名词（一个语音形式表示“鱼”，一个语音形式表示“低音乐器”）。再比如 lead，两种语音形式对应两个名词，表示“导线”的名词发音为 /l iː d/，表示“金属”的名词的发音为 /l eɪ d/。我们也可以把某些缩写词的排歧（前面我们把这样的排歧看成是非标准词的排歧）看成是同形异义词的排歧。例如，“Dr.”具有 doctor（博士）或 drive（驾驶）歧义；“St.”具有 Saint（神圣）或 Street（街道）歧义。最后，还有一些单词的大写字母有差别，如 polish/Polish，这些单词仅只在句子开头或全部字母都大写的文本中才可以看成同形异义词。

在实际应用中，后面这几种同形异义词是不能使用词类信息来解决的，在语音合成系统中通常可以忽略。另外，我们也可以尝试使用词义排歧算法来解决这样的问题。例如，我们可以使用 Yarowsky（1997）的决策表（decision-list）算法来排歧<sup>[2] 157-172</sup>。

最后，数字的发音是一个特别复杂的问题。电话号码“947-2020”的最自然的读音大概应该是“nine”-“four”-“seven”-“twenty”-“twenty”，而不是“nine”-“four”-“seven”-“two”-“zero”-“two”-“zero”。

Lieberman 和 Church（1992）把英语数字串的读音归纳为如下五种方法<sup>④</sup>：

一是顺序式读法（Serial）。每个数字单独读音。如，8765 的读音为“eight seven six five”。

二是组合式读法（Combined）。数字串按照一个整数来读音，每个数字根据它所在的位置分别加读“thousand、hundred”等进位数。如，8765

的读音为“eight thousand seven hundred sixty five”。

三是双对式读法（Paired）。数字一一对地按一个整数来读音。如果数字有奇数个，则第一个数字单独读音。按照这一规则，8765 的读音应为“eighty-seven sixty-five”。

四是百位式读法（Hundreds）。四位的数字串可按百位记数方式来读音。如，8765 的读音为“eighty-seven hundred (and) sixty-five”。

五是跟踪单位读法（Trailing Unit）。以若干个零为结尾的数字，非零的数字部分按顺序式读法来读音，零的部分按适当的进位制来读音。如，8765000 读为“eight-seven-six-five thousand”。

以上是英语文本归一化的一些主要研究成果，下面，我们再来考察一下汉语语音合成中的书面文本归一化问题。

#### 四、汉语语音合成中的文本归一化问题

汉语书面文本的归一化实际上是自然语言信息处理中的语言规划问题。我们提出这个问题的目的，是为了引起我国的语言规划专家在关注社会生活中的语言规划问题的同时，也关注一下自然语言信息处理中的语言规划问题。

我们认为，汉语的文本归一化与英语的文本归一化是相似的，在汉语的文本归一化中，也存在词例还原，非标准词处理，同形异义词处理等问题。下面逐一说明。

##### 1. 汉语文本的词例还原

汉语的书面文本是一个连续的汉字流，除了标点符号之外，词与词之间没有空白。在语音合成中，为了识别汉语的词以便查询发音词典，必须把隐藏在汉语书面文本中的词找出来，也就是要进行“切词”（word segmentation）。“切词”是汉语书面文本归一化的关键问题，也是中文信息处理的一个困难问题。关于汉语书面文本的自动切词，很多文章都有介绍，这里就不赘述了。

在经过切词处理后输出的文件中，汉语词边界用空格(space)表示，要特别注意人名、地名和机构名以及术语的切词是否正确，应当遵照《汉语拼音正词法基本规则》《GB13725 信息处理用现代汉语分词规范》等规范进行判断，为波形合成做好准备。



## 2. 汉语文本的非标准词 (NSW) 处理

汉语书面文本中的非标准词是诸如数字或专有名词之类的词，它们的读音比较特殊，一般不会存储在发音词典中，在语音合成中，在计算机读出它们之前，需要注出它们的读音。

汉语的非标准词包括如下几种：

### (1) 具有特殊读音的姓氏字

英语中的专有名词是很重要的非标准词。在汉语中，姓氏字也可以看成表示姓氏的词，所以，也是一种非标准词，在语音合成时，要区别姓氏字的特殊读音。如，“曾国藩”和“曾经”中的“曾”字，前者是姓氏字，读作 zēng，后者是标准词中的一个语素，读作 céng。例如：

记者带着这个问题采访了中国食文化研究会会长曾老。这位 75 岁老人曾参加八路军，四面八方都到过。

第一个“曾”是姓氏，应读作 zēng，后一个“曾”是一个语素，应读作 céng。

又如，“仇为之”(人名)和“仇恨”中的“仇”，前者是姓氏，读作 qiú，后者是标准词中的语素，读作 chóu。例如：

它的地址在旃坛寺，老板姓仇。

这里的“仇”是姓氏，应读作 qiú。

### (2) 数字

汉语中的数字也是很重要的非标准词。

对于汉语书面文本中的数字串，应区分它们的进位制，按汉语习惯以亿、万、千、百、十为单位读出。如 1,254,000,000 应读成“十二亿五千四百万 (shíèr yì wǔqiānsìbǎi wàn)”。例如：

这片林子共有 14,000 棵树。

这里的“14,000”应读成“一万四千 (yí wàn sì qiān)”。

(3) 年代、时间、电话号码、百分比、分数和小数

要区分汉语书面文本中年代、时间、电话号码和特殊数字表示的顺序式读法和进位制读法以及某些特殊读法，并要处理全角的数字符号。例如：

食源开发和物种驯化，中国在 4,000 年前就开始进行。”

这里的“4,000 年”应读为“四千年 (sì qiān nián)”，采用进位制读法。

美联社 16 日报道了中国首位进入太空的宇航员安全返回地面。报道说，在环绕地球 21 个小时后，航天飞船按计划准时着陆。中国的指挥控制中心宣布：中国首次载人航天飞行获得圆满成功。报道说，这次飞行的圆满完成是中国 11 年载人航天计划取得的最高成就，也是中国赢得世界声望的象征。

这里的“16”应读为“十六 (shíliù)”，“21”应读为“二十一 (èrshíyī)”，“11”应读为“十一 (shíyī)”，都采用进位制读法。

秦朝建立于公元前 221 年。

“221 年”应读为“二百二十一年 (èrbǎi èrshíyī nián)”，采用进位制读法。

马克思生于 1818 年。

“1818 年”应读为“一八一八年 (yī-bā-yī-bā nián)”，采用顺序式读法。

研讨会定于 12 月 23 日上午 9:35 开幕。

这里的“12”，“23”都采用进位制读法，分别读为“十二 (shíèr)”和“二十三 (èrshísān)”，“9:35”表示时间，应读为“九点三十五分 (jiǔ diǎn sānshíwǔ fēn)”。

旅游投诉电话是 9258。

“9258”应读为“九二五八 (jiǔ-èr-wǔ-bā)”，采用顺序式读法。

有 80% 的家庭主妇对一日三餐感到头疼。

“80%”应读为“百分之八十 (bǎifēnzhī bāshí)”。(注意：这里的 80% 是全角的数字符号，语音合成系统除了合成半角数字之外，还应当具有合成全角数字的能力)

美国太空发展经费占全球约 80.2%。

“80.2%”应读为“百分之八十点二 (bǎifēnzhī bāshí diǎn èr)”。(注意：这里的 80.2% 是全角的数字符号。在全角数字中，小数点使用中间圆点)

他的年龄是我的 1/2。

“1/2”应读为“二分之一 (èrfēnzhī yī)”。

我将住 5~8 天。

“5~8”应读为“五到八 (wǔ dào bā)”或者“五至八 (wǔ zhì bā)”。

### (4) 符号与单位

对符号和单位，有中文法定计量单位的应给出相应的拼音形式，并按照汉语普通话读音，读

音应遵照《关于在我国统一实行法定计量单位的命令》(1984年)的规定。一般外文符号可按原文给出,按照原文读音。例如:

1987年七月肯德基前门餐厅开业,门脸儿招牌上KFC三个大字,远远儿就瞧见了。顾客排队最长达20m,中午就餐最长达3000-4000人,真有人驱车20km从通县来的,够火的吧!

其中的“20m”应读为“二十米(èrshí mǐ);“20km”应读为“二十公里(èrshí gōnglǐ)”。

中国选手获得男子举重60kg级冠军。

“60kg”应读为“六十公斤(liùshí gōngjīn)”。

声音在空气中传播的速度是340米/秒。

“340米/秒”应读为“三百四十米每秒(sānbǎi sìshí mǐ měi miǎo)”。

比热容单位(焦耳每千克开尔文)的国际符号是J/(kg·K)。

其中的J/(kg·K)应按英文字母读音。

(5)以西文字母开头的词语

以西文字母开头的词语有的是借词,有的是外语缩略语,其中的西文字母按西文读音,汉字按汉语普通话读音。例如,“α粒子”应读为(alfalìzǐ),“B超”应读为(Bchāo),“ATM机”应读为(ATMjī)。

(6)专有名词的读音

专有名词是文语转换中的一个困难问题。词典中不可能事先列举出汉语中的一切专有名词。专有名词还可能来自其他语言,而且还可能有不同的拼写方法。语音合成和文语转换的很多应用都是与专有名词分不开的。例如,在与电话有关的应用中,电话簿和打电话都离不开人名和地名。汉语专有名词有的读音很特殊,应该注意区别。例如,“单”作为姓时应读为(shàn),不能读为(dān)。地名“枞阳”中的“枞”应读为(zōng),不能读为(cōng)。

(7)专业术语的读音

把语音技术应用于不同的专业领域需要正确处理专业术语的读音。例如,地貌学术语“潟湖”(浅水海湾因湾口被淤积的泥沙封闭而形成的湖)中的“潟”应读为(xì),不读为(xiè)。

3. 汉语文本的同形异义字(多音字)处理

同形异义词在汉语书面文本中表现为同形异义字。在汉语中,同形异义字也就是多音字。在语音合成中,要根据上下文条件的不同,在输出的拼音文件中对多音字给出不同的拼音。例如,“参加”和“参差”中的“参”,前者读为(cān),后者读为(cēn);“行军”和“银行”中的“行”,前者读为(xíng),后者读为(háng);“长江”和“局长”中的“长”,前者读为(cháng),后者读为(zhǎng)。

## 五、汉语语音合成中特殊韵律现象的处理

以上我们讨论了汉语书面文本归一化中的主要问题。此外,我们还要注意汉语的韵律(prosody)。韵律与汉语文本的归一化有密切联系。汉语是有声调的语言,在汉语语音合成中,必须注意变调、轻声等汉语的特殊韵律现象的处理。儿化具有区别意义和表达感情的作用,与韵律有关,在汉语书面文本归一化时也要注意处理。

1. “一”、“不”的读音

现有的用于语音处理的汉语发音词典还没有很好的模型来处理“一”“不”等字的读音。这是因为这些字发音变化的语音上下文环境很复杂。一般在发音词典中只包含某些最基本的形式,如“一”的发音为(yī)。在语音合成中,要使用相应的算法根据上下文推出它们的发音变体。

(1)“一”在非去声前变为去声。例如:

在阴平前:一天(yìtiān) 一般(yìbān)

在阳平前:一时(yìshí) 一齐(yìqí)

在上声前:一手(yìshǒu) 一起(yìqǐ)

(2)“一”、“不”在去声前变为阳平。例如:

一半(yíbān) 一定(yídìng)

一再(yízài) 一贯(yíguàn)

不论(búlùn) 不但(búdàn)

不幸(búxìng) 不愧(búkù)

(3)“一”、“不”夹在词语中间时变为轻声。

例如:

想一想(xiǎngyixiǎng) 看一看(kànyikàn)

差不多(chànbudūo) 好不好(hǎobuhǎo)

2. 上声变调

上声在语流中发生音变,在语音合成中,这种语流音变十分复杂,也要使用相应的算法根据上下文推出它们的发音变体,主要应处理如下的

现象。

(1) 上声在非上声(阴平、阳平、去声)前一律变为平上,调值由原来的 214 变为 21,只降不升。如,“影星”、“影评”、“影印”中的“影”应读为平上。

(2) 上声在上声前(上上相连),前一个上声变得像阳平,调值由 214 变为 24,只升不降。如,“本领”、“讲解”、“导演”中的“本”、“讲”、“导”的调值均为 24。

### 3. 轻声的读音

普通话的轻声具有区别意义的作用,在语音合成中,应当注意如下要点:

(1) 辨义轻声。同一个汉字,由于是否读轻声而导致语义不同。例如,“老子”读轻声时表示骄傲的自称,不读轻声时表示古代人名或书名。

(2) 连接词“和”读为轻声。

(3) 助词“的、地、得”读为轻声。

(4) 方位结构中的非中心音节读为轻声。例如,“眼里”、“手上”、“乡下”中的“里”、“上”、“下”读为轻声。

(5) 双字重叠的指人名词,后一个音节读为轻声。例如,“哥哥”、“妈妈”、“婆婆”中的后一个音节“哥”、“妈”、“婆”读为轻声。

(6) 单音节动词重叠式的后一个音节读为轻声。如,“看看”、“洗洗”、“说说”中的后一个音节“看”、“洗”、“说”读为轻声。

### 4. 儿化的读音

儿化音对于语音合成的自然度有重要的作用,在语音合成中,应当对儿化进行系统化的处理,这也是汉语文本归一化应当注意的问题。

(1) 对于有区别意义作用的儿化词,必须按儿化读音。例如:

信(信件)——信儿(消息)

头(脑袋)——头儿(领头的人)

(2) 对于有区别词性作用的儿化词,必须按儿化读音。例如:

盖(动词)——盖儿(名词)

尖(形容词)——尖儿(名词)

(3) 对于表示感情色彩的儿化词,尽量按儿化读音。例如:

小孩——小孩儿 好玩——好玩儿

语音词典中,应当对上述儿化词一一标注其拼音,儿化词中的音节数等于汉字字数减一。例如,“花儿”应标注为(huār),其音节数为 1。

当自动切词得到后缀“儿”时,将“儿”与前面的单词合并,并把前面单词的最后一个音节儿化,语音合成时“儿”不再发音。

非儿化词中的“儿”,应当单独读成一个音节。例如,“孤儿”、“男儿”、“混血儿”中的“儿”,都应当读成一个音节,不能儿化。

对于汉语语音合成中文本归一化问题,我国语言规划界似乎还没有进行过深入研究,本文介绍英语文本归一化中的某些研究成果,并结合汉语语音合成,列举了一些值得关注的问题,这个问题涉及到自然语言处理、语音学、汉字学、词汇学、语言规划等领域,作者知识有限,很可能是以蠡测海,以管窥豹,肯定有许多片面或不妥之处,请批评指正。

### 注释:

① 详见冯志伟《自然语言处理的形式模型》(中国科学技术出版社,2009年版)和冯志伟、孙乐译《自然语言处理综论》(译自 Jurafsky, D and Martin, H. J. *Speech and Language Processing, First Edition*),电子工业出版社,2005年版)的相关内容。

② 主要参照 Jurafsky, D and Martin, H. J. *Speech and Language Processing, Second Edition*, 2008.

③ 关于德语语音合成的研究主要参照 Demberg, V. *Letter-to-phoneme conversion for a German text-to-speech system, Diplomarbeit Nr. 47, Universität, Stuttgart*, 2006.

④ 参照 Liberman, M. Y. and Church, K. W. *Text analysis and word pronunciation in text-to-speech synthesis*, In Furui, S. and Sondhi, M. M. (Eds.), *Advances in Speech Signal Processing*, 1992.

⑤ 参照 Sproat, R., Black, A. W. et al, *Normalization of non-standard words*, *Computer Speech & Language*, 15(3), 2001.

⑥ 参照 Fackrell, J. and Skut, W. *Improving pronunciation dictionary coverage of names by modeling spelling variation*, In *Proceedings of the 5<sup>th</sup> Speech Synthesis Workshop*, 2004.

### 参考文献

[1] Spiegel, M. F. *Proper name pronunciation for speech technology application*, In *Proceedings of IEEE Workshop*

on Speech Synthesis, 175-178, 2002.

et al, (Eds.), Progress in Speech Synthesis, 157-172,

[2] Yarowsky, D. Homograph disambiguation in Springer, 1997.

text-to-speech synthesis, In van Santen, J. P. H., Sproat, R.

## **Text Normalization in Speech Synthesis**

FENG Zhi-wei

(Institute of Applied Linguistics, The Ministry of Education, Beijing, 100010)

**Abstract:** The text normalization is a topic which closely correlated with language planning. This paper firstly introduces 3 main tasks in English text normalization: sentence tokenization, non-standard words (NSW) processing, homograph disambiguation. Then this paper discusses the main problems in Chinese text normalization: tokenization of Chinese text, non-standard words processing of Chinese text, homograph disambiguation of Chinese text. In the end, some special prosody phenomena in Chinese speech synthesis are discussed.

**Key words:** Speech synthesis; Tokenization; Non-standard words (NSW); Homograph; Disambiguation; Prosody