



文章编号:1001-9081(2018)04-0971-07

DOI:10.11772/j.issn.1001-9081.2017092149

基于 TensorFlow 的俄语词汇标音系统

冯伟, 易绵竹*, 马延周

(战略支援部队信息工程大学(洛阳), 河南 洛阳 471003)

(*通信作者电子邮箱 mianzhu@ gmail.com)

摘要:针对俄语语音合成和语音识别系统中发音词典规模有限的问题,提出一种基于长短时记忆(LSTM)序列到序列模型的俄语词汇标音算法,同时设计实现了标音原型系统。首先,对基于 SAMPA 的俄语音素集进行了改进设计,使标音结果能够反映俄语单词的重音位置及元音弱化现象,并依据改进的新音素集构建了包含 20 000 词的俄语发音词典;然后利用 TensorFlow 框架实现了这一算法,该算法通过编码 LSTM 将俄语单词转换为固定维数的向量,再通过解码 LSTM 将向量转换为目标发音序列;最后,设计实现了具有交互式单词标音等功能的俄语词汇标音系统。实验结果表明,该算法在集外词测试集上的词形正确率达到了 74.8%,音素正确率达到了 94.5%,均高于 Phonetisaurus 方法。该系统能够有效为俄语发音词典的构建提供支持。

关键词:俄语;词汇标音;长短时记忆网络;序列到序列;TensorFlow

中图分类号:TP391.1 **文献标志码:**A

Russian phonetic transcription system based on TensorFlow

FENG Wei, YI Mianzhu*, MA Yanzhou

(The PLA Strategic Support Force Information Engineering University (Luoyang), Luoyang Henan 471003, China)

Abstract: Focusing on the limited pronunciation dictionary in Russian speech synthesis and speech recognition system, a Russian grapheme-to-phoneme algorithm based on Long Short-Term Memory (LSTM) sequence-to-sequence model was proposed, as well as a phonetic transcription system. Firstly, a new Russian phoneme set based on Speech Assessment Methods Phonetic Alphabet (SAMPA) was designed, making transcription results can reflect the stress position and vowel reduction of Russian words, and a 20 000-word Russian pronunciation dictionary was constructed according to the new phoneme set. Then, the proposed algorithm was implemented by using the TensorFlow framework, in which the Russian word was converted into a fixed-length vector by encoding LSTM, and then the vector was converted into the target pronunciation sequence by decoding LSTM. Finally, the Russian phonetic transcription system was designed and implemented. The experimental results on out-of-vocabulary test set show that the word correct rate reaches 74.8%, and the phoneme correct rate reaches 94.5%, which are higher than those of Phonetisaurus method. The system can effectively support the construction of the Russian pronunciation dictionary.

Key words: Russian; phonetic transcription; Long Short-Term Memory (LSTM); sequence-to-sequence; TensorFlow

0 引言

发音词典是语音信息处理研究中的重要基础资源,在语音合成和语音识别系统中发挥了关键作用。俄语作为一种拼音文字,在语言发展中不断有新词和外来词产生,发音词典必然难以包括所有俄语单词的发音。字音转换(Grapheme-to-Phoneme conversion, G2P)技术可以对俄语单词及其变化形式进行自动注音,有效解决集外词(Out-Of-Vocabulary, OOV)的注音问题,为俄语发音词典的构建提供支持。

字音转换可分为两类:

1) 基于规则的方法,即通过对俄语正字法和发音规律的总结,人工制定出俄语的字音转换规则,然后根据规则实现对单词发音的预测。俄罗斯圣彼得堡大学的 Karpov 等^[1-2]在俄语语音识别系统的开发过程中对基于规则的俄语字音转换算法进行了研究。该算法利用俄语辅音变化和元音弱化等

规则,借助大规模俄语重音词典、形态词典以及同形词词典,经过 7 个步骤、2 次循环完成。Karpov 等^[1-2]利用该算法构建了俄语语音识别系统需要的发音词典,但并没有对算法性能进行严格的测试。由于俄语发音特征复杂多变,正字法的约束也在逐渐减弱,规则中难免会出现无法覆盖到的例外情况,这些都会对字音转换的准确率造成影响。

2) 数据驱动的方法,是目前主流的字音转换方法。典型的数据驱动方法基于如下思想:首先对训练集中的字素和音素建立对齐关系,然后利用概率统计方法建立发音模型,最后通过解码算法计算概率最大的标音结果。例如,Jaipojamarn 等^[3]提出了多对多的对齐方法,并将隐马尔可夫模型(Hidden Markov Model, HMM)应用于发音模型建模;Bisani 等^[4]提出了联合序列的建模方法,并在英语、德语和法语测试集上进行了测试;Novak 等^[5]将加权有限状态转化器(Weighted Finite-State Transducer, WFST)运用于算法的对

收稿日期:2017-09-04;修回日期:2017-11-18。 基金项目:洛阳市社会科学规划项目(2016B285)。

作者简介:冯伟(1993—),男,陕西西安人,硕士研究生,主要研究方向:自然语言处理; 易绵竹(1964—),男,四川营山人,教授,博士,主要研究方向:计算语言学、语言信息处理; 马延周(1977—),男,河南洛阳人,副教授,博士,主要研究方向:计算语言学、语言信息处理。



齐、建模、解码过程,提出了基于循环神经网络语言模型(Recurrent Neural Network Language Model, RNNLM)的 N-best 解码算法,以及最小贝叶斯风险(Lattice Minimum Bayes-Risk, LMBR)词图解码算法,并在三个英语测试集上进行了对比测试。

神经网络近年来被广泛应用于深度学习的相关问题。Graves^[6]提出了基于长短时记忆(Long Short-Term Memory, LSTM)网络的序列到序列(sequence-to-sequence)模型,该模型可以将一个长度可变的输入序列翻译为目标输出序列,突破了传统的固定长度输入问题的限制,成功地将神经网络运用于序列型任务。目前该模型已在谷歌翻译、人机对话、计算机视觉等系统上得到了广泛的应用,并表现出了出色的效果^[7-8]。Yao 等^[9]将该方法应用于解决英语字音转换问题,并在卡内基梅隆大学(Carnegie Mellon University, CMU)、NetTalk、Pronlex 数据集上进行了测试,音素正确率达到了 92% 以上,词形正确率达到了 70% 以上。

综上所述,俄罗斯学者已经对基于规则的俄语词汇标音方法进行了一些研究,但基于规则的方法对语言学知识要求较高,规则的撰写和维护难度较大,实现起来有一定的困难。数据驱动的方法是目前自然语言处理领域的主流方法,但已有研究都以英语为主要目标,还没有俄语方面的有关研究和实验。不同的语种在模型训练过程中难免存在差异,且国内针对俄语语音处理的研究尚处于探索阶段,基础资源相对匮乏,有必要以俄语语音学知识为基础,完善俄语语料资源,对俄语字音转换算法的实现与应用作进一步研究。本文对数据驱动的俄语词汇标音方法进行探索,尝试运用基于 TensorFlow 的 LSTM 序列到序列模型算法,利用端到端(end-to-end)的思想实现单词到发音的转换。相对于传统算法,该算法不需要预先创造字素音素的对齐关系,可以直接对任意长度的序列进行处理,避免了对齐过程出现错误的可能性。

1 改进的俄语音素集设计

音素集就是音素的集合。由于国际音标书写复杂、机读性差,在俄语语音处理系统中,需要依据计算机可读的 SAMPA(Speech Assessment Methods Phonetic Alphabet)符号设计俄语音素集,从而构建俄语发音词典并训练俄语声学模型。俄语音素集中应尽可能包括俄语全部的音素,但如果音素集过大,单词注音结果的不确定性将会显著增加,大大提高解码过程的计算复杂度;若音素集太小,则会降低单词标音的精确度,影响语音处理系统的性能。为了体现俄语重音变化和元音弱化现象,本文对原始 SAMPA 俄语音素集进行了改进,设计了新的俄语音素集。

目前国际上俄语音素集的设计有多种方案。IPA(International Phonetic Alphabet)俄语音素集共包含 55 个音素和 1 个重音符号(音素分为 38 个辅音和 17 个元音,元音包括 11 个重读元音和 6 个非重读元音)^[10]。SAMPA 俄语音素集共包含 42 个音素,分为 36 个辅音和 6 个元音,其元音音素没有重读与弱化之分,仅仅将弱化的元音[e]和[o]分别用[i]和[a]表示^[11];卡内基梅隆大学(CMU)设计的俄语音素集共包含 50 个音素和 1 个无音符号(音素分为 36 个辅音和 14 个元音,并将元音分为 6 个重读元音和 8 个非重读元音)^[12]。

通过对以上三个俄语音素集的研究,结合俄语音素的发音规则,重点对元音音素从一级弱化和二级弱化的角度进行区分^[13],本文在原有俄语 SAMPA 音素集的基础上,增加了 4 个弱化后的元音和一个重音符号“!”,设计了共包含 46 个音素的俄语音素集。音素包括 36 个辅音和 10 个元音,元音又细分为 6 个重读元音和 4 个非重读元音。新增的 4 个元音如表 1 所示。

表 1 俄语弱化元音表

Tab. 1 Table of Russian unstressed vowels

| SAMPA 符号 | IPA 符号 | 单词示例 | 释义 | 单词音标 |
|----------|--------|---------|-----|------------------|
| 6 | ə | авто | 自动的 | 6 ft ! o |
| @ | ɛ | авиатор | 飞行员 | 6 v' i ! a t @ r |
| I | ɪ | истинно | 真实地 | ! i s t' I n 6 |
| } | ɯ | стимул | 刺激 | s t' ! i m ɿ l |

为了验证新音素集的有效性,本文从发音词典中随机抽取了 200 个俄语单词,分别用原始 SAMPA 音素集和新音素集进行标音,交由俄语语音学专家进行人工比对验证。

验证结果表明,本文设计的新音素集能够清晰地标明俄语单词的重音位置,有效地区分元音一级弱化和二级弱化后的读音区别,相对于原始的 SAMPA 音素集标音更加准确,可读性更强。表 2 以部分单词为例,对改进的音素集与原始音素集的标音结果进行了对比。

表 2 改进的音素集标音示例

Tab. 2 Examples of transcription by improved phoneme set

| 单词 | 释义 | 原始音素集 | 改进的音素集 |
|---------|----|------------------|--------------------|
| август | 八月 | a v g u s t | ! a v g ɿ s t |
| банана | 香蕉 | b a n a n a | b 6 n ! a n @ |
| вечер | 晚上 | v e tS' I r | v' ! e tS' I r |
| девочка | 女孩 | d' e v a tS' k a | d' ! e v @ tS' k 6 |

2 长短时记忆神经网络

2.1 循环神经网络

自然语言处理中的大部分问题本质上都是序列化的。例如,段落是由句子构成的序列,句子是由单词构成的序列,在机器翻译、人机对话、语音识别等应用中,模型的输入和输出都是序列数据。类似地,单词和发音也可以看作是由字素和音素构成的序列。循环神经网络就是专门用于处理序列数据的深度学习模型。循环神经网络(Recurrent Neural Network, RNN)出现于 20 世纪 80 年代,因为实现困难,其发展早期并没有得到合适的应用。最近几年,由于神经网络结构的进步和 GPU 上深度学习训练效率的突破,RNN 变得越来越流行,在人工智能的多个领域中得到应用。

RNN 的前向传播过程可用公式^[14]表示为:

$$a_t = b + Wh_{t-1} + Ux_t \quad (1)$$

$$h_t = \tanh(a_t) \quad (2)$$

$$o_t = d + Vh_t \quad (3)$$

$$y_t = \text{softmax}(o_t) \quad (4)$$

其中:输入层与隐藏层之间通过参数矩阵 U 连接;不同时刻的隐藏层之间以参数矩阵 W 连接;隐藏层与输出层之间以参数矩阵 V 连接。 b, d 为偏置向量; x_t 为 t 时刻的输入; a_t 为决定 t 时刻隐藏层状态的参数,包括现有的输入和对过去记忆的总结;



h_t 表示隐藏层状态; o_t 表示 t 时刻的输出; y_t 为经过归一化后的预测概率。

2.2 长短时记忆模型

循环神经网络虽然可以记忆整个时间序列中的信息,但记忆中影响最大的还是最后输入的一些信号,而较早的信号强度将越来越弱,即决定循环神经网络输出的还是最后输入的信号。这就造成了 RNN 的长时依赖 (Long-term Dependencies) 问题。

长短时记忆模型就是专门为解决长时依赖问题而对循环神经网络的改进。通过将 RNN 中的普通神经元替换为可以存储记忆的 LSTM 单元 (Cell),可以有效利用数据中的长距离依赖信息,由 Hochreiter 等^[15]在 1997 年提出。不同于原始 RNN 单一的 tanh 循环体结构,LSTM 模型在短期记忆单元 c_t 的基础上,增加了记忆单元 C_t 用来保留长时记忆,以及三个门控制器,分别是:输入门 (input gate)、输出门 (output gate) 和遗忘门 (forget gate)。标准循环神经网络与 LSTM 模型的结构对比如图 1 所示。

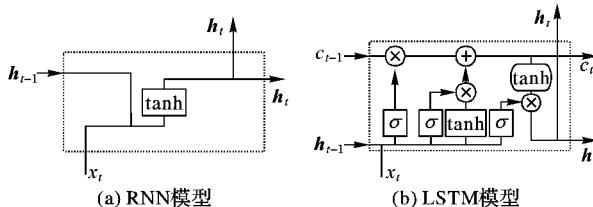


图 1 RNN 与 LSTM 模型结构对比示意图

Fig. 1 Structure comparison of RNN and LSTM

门结构运算由非线性激活函数 Sigmoid 和点乘运算控制。Sigmoid 函数的取值范围为 $[0,1]$,描述了信息传递的比例。取值为 0 时表示不允许所有信息传递,即删除之前的记忆;取值为 1 时表示所有信息可以通过,完全保留这一分支的记忆。

在每一个时刻,遗忘门会根据当前输入 x_t 、上一时刻输出 h_{t-1} 和上一时刻状态 c_{t-1} 控制上一时刻长期记忆的保留程度^[14]:

$$f_t = \sigma(\mathbf{T}_{sf}x_t + \mathbf{T}_{hf}h_{t-1} + \mathbf{T}_{cf}c_{t-1} + \mathbf{b}_f) \quad (5)$$

输入门会根据 x_t 、 c_{t-1} 和 h_{t-1} 控制新记忆写入长期记忆的程度,决定当前状态 c_t :

$$i_t = \sigma(\mathbf{T}_{si}x_t + \mathbf{T}_{hi}h_{t-1} + \mathbf{T}_{ci}c_{t-1} + \mathbf{b}_i) \quad (6)$$

$$c_t = f_t c_{t-1} + i_t \tanh(\mathbf{T}_{sc}x_t + \mathbf{T}_{hc}h_{t-1} + \mathbf{b}_c) \quad (7)$$

输出门则会根据最新状态 c_t ,以及 h_{t-1} 和 x_t ,基于长时记忆和短期记忆综合决定该时刻的输出 h_t :

$$o_t = \sigma(\mathbf{T}_{so}x_t + \mathbf{T}_{ho}h_{t-1} + \mathbf{T}_{co}c_t + \mathbf{b}_o) \quad (8)$$

$$h_t = o_t \tanh(c_t) \quad (9)$$

其中: σ 表示 Sigmoid 函数; i 、 f 、 o 和 c 分别表示输入门、遗忘门、输出门和记忆单元,其向量维数与隐藏层中向量相同。权值矩阵 \mathbf{T} 的下标描述了其含义,例如 \mathbf{T}_{hi} 为隐藏-输入门权值矩阵, \mathbf{T}_{so} 为输入-输出门权值矩阵。

LSTM 通过门结构维护和控制神经网络中每个时刻的状态信息,凭借对状态信息的存储和修改,从而解决了长时依赖的难题。基于以上运算机制,LSTM 对于长序列问题的理解分析能力相对于 RNN 得到了大幅提高,因此可以有效应用于俄语单词序列到发音序列的预测问题。

3 基于 LSTM 的序列到序列模型

3.1 字素序列到音素序列

LSTM 神经网络因其出色的长距离序列化信息处理能力,可以有效应用于序列到序列的问题。在词汇标音过程中,LSTM 序列到序列模型不需要预先创造字素音素的对齐关系,可以直接对任意长度的序列进行处理,避免了对齐过程出现错误的可能性。

基于 LSTM 的序列生成过程可以概括性地描述为条件概率 $p(y_1, y_2, \dots, y_n | x_1, x_2, \dots, x_m)$ 最大值的估算过程。 (x_1, x_2, \dots, x_m) 表示输入的字素序列, (y_1, y_2, \dots, y_n) 表示对应的输出序列,两个序列的长度 n 和 m 不一定相等。LSTM 在计算条件概率时,首先对不定长序列 (x_1, x_2, \dots, x_m) 进行学习,根据最后一个隐藏层的状态,将序列表示为固定维数的向量 s ,然后将标准 LSTM 语言模型的初始隐藏层状态设置为向量 s ,根据语言模型利用 2.2 节描述的 LSTM 公式(5)~(9) 计算序列 (y_1, y_2, \dots, y_n) 的概率。用公式表示为:

$$p(y_1, y_2, \dots, y_n | x_1, x_2, \dots, x_m) = \prod_{t=1}^n p(y_t | s, y_1, y_2, \dots, y_{t-1}) \quad (10)$$

其中:每个输出的概率分布 $p(y_t | s, y_1, y_2, \dots, y_{t-1})$ 通过激活函数 softmax 映射到音素集中的音素。

在数学定义中,softmax 指一种归一化指数函数,它将一个 k 维向量 z 中的每个元素转换为 $(0,1)$ 的区间,计算公式如下:

$$\delta(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad (11)$$

机器学习中常用这种方法将类似判别函数的置信度值转换为概率形式。softmax 函数常用于输出层,用于指定唯一的分类输出^[16]。

3.2 LSTM 编码-解码模型

基础的序列到序列模型由编码器-解码器 (Encoder-Decoder) 结构组成,该结构的特点是:输入序列经过编码器网络得到向量表示后,解码器网络基于这个向量生成新的序列。LSTM 编码-解码模型两个 LSTM 网络,分别是处理输入的编码器网络和生成输出的解码器网络。编码-解码过程如图 2 所示^[17]。

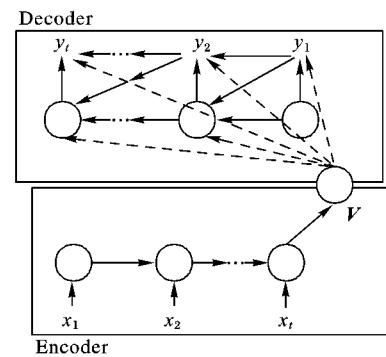


图 2 编码-解码过程示意图

Fig. 2 Schematic diagram of encode-decode process

3.2.1 编码器

编码器网络按照 LSTM 单元结构进行运算。每个时间



步,编码器的输入为单词的一个字素(在机器翻译等任务中,也可以是一个字或单词),当遇到终止符 <s> 时输入结束,编码器根据最后一个隐藏层的状态,将该单词序列表示为固定长度的向量 v 。依赖于 LSTM 对长距离信息处理的能力,向量 v 能够包含整个单词序列的字素信息。在每个时间 t ,隐藏层的状态 h_t 可用公式表示为:

$$h_t = f(x_t, h_{t-1}) \quad (12)$$

其中: f 表示非线性激活函数,为编码 LSTM 单元结构; h_{t-1} 表示上一时刻隐藏层状态; x_t 为当前时刻的输入。向量 v 为最后一个隐藏层或多个隐藏层的加权和,运算符号用 φ 表示:

$$v = \varphi(h_1, h_2, \dots, h_t) \quad (13)$$

3.2.2 解码器

在解码过程,向量 v 将作为隐藏层的初始状态输入解码 LSTM 网络。解码器通过 t 时刻的隐藏层状态 h_t 、前一个音素 y_{t-1} 以及向量 v ,逐步计算当前时刻音素 y_t 的概率分布,当遇到终止符 </os> 时结束预测,得到整个输出序列。这一过程用公式表示如下:

$$h_t = f(h_{t-1}, y_{t-1}, v) \quad (14)$$

$$P(y_t | v, y_1, y_2, \dots, y_{t-1}) = g(h_t, y_{t-1}, v) \quad (15)$$

其中: f 表示解码 LSTM 单元结构; g 一般为 softmax 函数。解码过程,解码器使用启发式集束搜索(beam search)算法^[18]在序列输出前检索大量词汇,选择后验概率最高的候选序列为最优解,作为解码器最终输出的音素序列。LSTM 编码-解码模型的训练使用随时间反向传播(Backpropagation Through Time, BPTT)算法,利用解码过程中产生的误差更新网络的权值参数^[19]。

3.2.3 俄语单词编码-解码过程

俄语单词序列转换到发音序列的基本思想为:编码 LSTM 逐步读取俄语单词的每个字素,将序列映射为一个固定维数表示的向量,解码 LSTM 本质上是一个基于输入序列的 LSTM 语言模型,结合向量、隐藏层状态和上一时刻的音素,逐个预测音素,输出发音序列。

以俄语单词“пай”(天堂)的第二格形式“пая”[r ! a j 6]为例,LSTM 编码-解码模型的示例如图 3 所示。图 3 中神经网络由两层组成,虚线左侧为编码 LSTM,右侧为解码 LSTM。编码 LSTM 按照时间逆序读取输入序列“<s> яар”,根据最后一个隐藏层的状态,将序列“пая”表示为固定维数的向量 v 。解码 LSTM 在遇到起始符 <os> 后被激活,将向量 v 作为隐藏层的初始状态,逐个计算下一个音素产生的概率,通过集束搜索算法得到最终输出的音素序列“r ! a j 6 </os>”。<s> 表示输入序列的起始符,<os> 和 </os> 分别表示输出音素的起始符和终止符,起始符与终止符使模型可以对任意长度的序列进行编码和解码,解码 LSTM 在 </os> 后终止预测。另外,编码器按照逆序读取字素,可以在数据中引入短期依赖关系,简化了训练优化的过程^[9]。

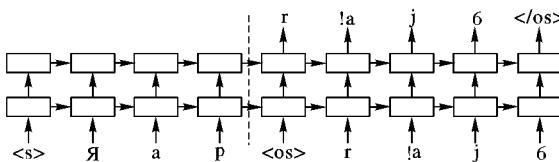


图 3 LSTM 编码-解码网络示意图

Fig. 3 Schematic diagram of LSTM Encode-Decide network

4 基于 TensorFlow 的俄语词汇标音系统

4.1 TensorFlow

Google 经过长期的研究,在内部使用了第一代分布式机器学习框架 DistBelief 之后,于 2015 年 11 月推出了目前最优秀的深度学习框架之一 TensorFlow,并在 GitHub 上开源。TensorFlow 的官方定义为:TensorFlow 是一个基于数据流图(data flow graph)的数值计算开源软件库,其灵活的架构设计可以让用户以单机或分布式的方式将计算部署在台式机、服务器甚至是手机上。Tensorflow 广泛支持包括计算机视觉、语音识别、人机对弈和自然语言处理等大量功能^[20]。

TensorFlow 的数据计算过程可以表示为数据流图,也称计算图(Computational Graph)。计算图是一个有向图,其中每一个运算操作(operation)作为一个节点(node),节点与节点之间的连接称为边(edge),在边中流动(flow)的多维数组数据称为张量(tensor)。计算图的执行可以看作张量按照图的拓扑顺序,从输入节点逐步流过所有中间节点,最后流到输出节点的过程。

4.2 系统开发环境

该系统的开发基于 Ubuntu 操作系统,使用 Python 程序语言,在 TensorFlow 深度学习框架的支持下进行。具体的开发环境如下:

操作系统:Ubuntu 14.04-amd64-LTS。

开发语言:Python 2.7。

深度学习框架:TensorFlow 1.0.0。

Python 开发平台:Qt 4.8.4 + PyQt 4.12 + SIP 4.19 + QScintilla 2.8 + Eric 6.1.11。

4.3 系统框架结构

基于 TensorFlow 框架的俄语词汇标音系统整体框架如图 4 所示。

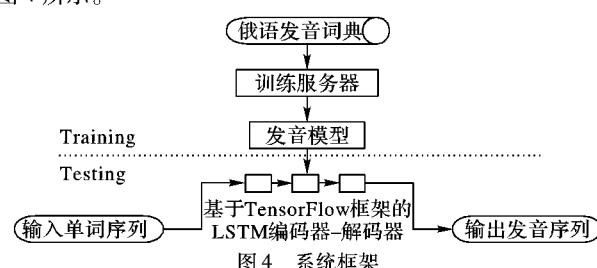


Fig. 4 Framework diagram of the system

4.4 系统功能与实现

系统的设计基于 PyQt 工具集、QtDesigner 界面设计器,以及 Eric 开发环境,并利用 QSS(Qt Style Sheets)语言进行 GUI 界面美化。系统主要包括模型训练和单词标音功能,其实现方法如下:

1) 发音模型训练功能。

模型训练功能以发音词典为训练语料,通过调用 TensorFlow 的 tf.contrib.rnn 接口实现 LSTM 网络的定义,调用 tf.contrib.legacy_seq2seq 接口的 model_with_buckets 方法进行模型训练。优化参数使用 sgd 算法,通过调用 tf.train 接口的 GradientDescentOptimizer 方法实现,并通过 tf.nn.sampled_softmax_loss 方法降低梯度更新时的计算复杂度。

模型训练时的主要参数包括:

source_vocab_size: 输入序列词表大小。



`target_vocab_size`:目标序列词表大小。

`buckets`:处理不同长度的序列的方法,由一对 (I, O) 表示, I, O 分别表示该bucket处理的最大输入和输出序列的长度,一般为 $(2, 4)、(4, 8)、(8, 16)$ 。

`size`:模型每一层的单元数。

`num_layers`:模型的网络层数。

`max_gradient_norm`:梯度最大修剪规范。

`batch_size`:训练时批处理的大小。

`learning_rate`:初始的学习率。

`learning_rate_decay_factor`:学习率衰减因子。一定的阶段之后,学习率按照衰减因子进行衰减。

`use_lstm = True`:是否使用LSTM单元。True表示使用LSTM, False表示使用GRU单元。

`num_samples = 512`:执行采样softmax函数的临界值。使用softmax函数处理输出序列时,若输出词表较大会影响计算效率。因此,当输出词表大于512时使用采样softmax函数;当输出词表小于512时使用softmax函数。

`optimizer = "sgd"`:自适应梯度调节器,使用sgd算法。

`dtype = tf.float32`:存储内部变量的数据类型为float32。

2) 单词标音功能。

系统的标音功能为交互式,根据载入的发音模型对输入的单词进行编码-解码操作,将单词序列转换为发音序列。这一过程将调用`tf.contrib.legacy_seq2seq`接口的`basic_rnn_seq2seq`方法。首先通过编码LSTM将输入序列转换为向量表示,然后将编码器最后一个隐藏层的状态作为输入,激活解码LSTM。方法的主要参数包括:

`encoder_inputs`: $[batch_size \times input_size]$ 表示的二维张量列表,每一个二维张量代表某一时刻的输入,`batch_size`具体指某一时刻输入的字素个数,`input_size`指编码器的长度。

`decoder_inputs`: $[batch_size \times output_size]$ 表示的二维张量列表。

`cell`:类`tf.contrib.rnn.LSTMCell`的实例,表示使用LSTM单元。

`dtype = tf.float32`:LSTM单元的数据类型为float32。

`basic_rnn_seq2seq`方法的返回值为二维张量表示的二元组(`outputs, state`)。`outputs`对应每个时间步中解码器的输出,形式为 $[batch_size \times output_size]$;`states`表示每个时间步中解码器的内部状态,形式为 $[batch_size \times cell.state_size]$ 。

解码完成后,调用`tf.nn`接口的softmax方法将张量转换为对应的音素,输出音素序列。

5 实验测试

5.1 实验准备

系统的实验准备工作包括以下内容:发音词典语料准备、实验环境搭建以及评测指标制定。

5.1.1 词典语料准备

在实验阶段,本文首先完成了俄语发音词典的构建工作。原始语料的主要来源包括维基百科、CMU资源库以及一些开源的俄语语料库。数据的获取通过编写爬虫程序实现,并人工进行适当的判别和整理。词典的整理过程主要包括:

1)过滤和筛选操作。去除原始语料中存在乱码、格式错

误等问题的数据,删除词组、句子或不构成单词的样例。

2)音素集归一操作。将原始语料中的IPA和CMU音素全部用对应的改进SAMPA音素集替换,实现发音词典音素集的统一。

3)音素分隔操作。对改写后音标中的每个音素进行识别,并以空格符为分隔标志将其隔开。

4)去重操作。去除词形和发音都相同的重复样例,保留词形相同但发音不同的样例。

5)排序操作。按照单词的字母顺序对发音词典进行排序。

经过以上整理过滤,最终形成了使用改进的SAMPA音素集标注并包含重音信息的俄语发音词典,词典共包含20 000词条样例。

5.1.2 实验环境

实验的模型训练和测试工作在服务器上进行,服务器的硬件配置为:曙光云图W760-G20高性能服务器,16核i7至强CPU,128 GB内存,4×600 GB硬盘。

5.1.3 评测指标

衡量字音转换算法的评测指标分别是音素正确率和词形正确率^[21]。音素错误一般存在三类,分别是插入错误、删除错误,以及替换错误。音素正确率的计算公式如下:

$$\text{音素正确率} = \frac{\text{正确转换的音素数}}{\text{音素总数}} \times 100\% \quad (16)$$

$$\begin{aligned} \text{正确转换的音素数} &= \\ \text{音素总数} - (\text{插入错误数} + \text{删除错误数} + \text{替换错误数}) & \end{aligned} \quad (17)$$

词形正确率的计算公式如下:

$$\text{词形正确率} = \frac{\text{正确标注的词形数}}{\text{词形总数}} \times 100\% \quad (18)$$

5.2 实验结果

本文将20 000词发音词典分为两部分,90%作为训练数据,10%作为测试数据。在模型训练阶段,通过对LSTM网络的层数(layers)和单元数(units)进行调整,观察模型参数对系统性能的影响。测试阶段,本文使用了对比验证的方法,将训练得到的4个发音模型与文献[6]方法提出的Phonetisaurus工具进行了对比测试。此外,为了衡量数据来源对系统性能的影响,分别使用了训练集语料(集内词)和测试集语料(集外词)作为测试数据。

从表3的实验结果可以看出,LSTM模型的层数和单元数会对系统性能造成显著影响。当层数为3,单元数为512时系统的性能最佳,在集内词测试的音素正确率达到了99.2%,词形正确率达到了95.8%;在集外词测试的音素正确率达到了94.5%,词形正确率达到了74.8%,均高于Phonetisaurus方法。

表3 Phonetisaurus与LSTM模型的正确率对比

Tab. 3 Comparison of accuracy of Phonetisaurus and LSTM

| 模型参数 | 集内词正确率/% | | 集外词正确率/% | |
|----------------------|----------|------|----------|------|
| | 音素 | 词形 | 音素 | 词形 |
| Phonetisaurus 8-gram | 96.8 | 81.5 | 92.2 | 65.3 |
| LSTM 层数=2,单元数=64 | 97.2 | 89.5 | 92.5 | 67.6 |
| LSTM 层数=2,单元数=512 | 98.8 | 94.7 | 94.1 | 74.2 |
| LSTM 层数=3,单元数=64 | 97.4 | 89.5 | 92.2 | 67.4 |
| LSTM 层数=3,单元数=512 | 99.2 | 95.8 | 94.5 | 74.8 |



表 4 显示了 4 种不同模型的运行效率对比。模型的层数和单元数会对训练时间、模型大小和解码速度造成影响。尽管当 LSTM 模型的层数为 3, 单元数为 512 时系统性能最优, 但训练时间的提高和解码速度的变慢导致系统效率发生了显著下降。

表 4 不同参数的 LSTM 模型效率对比

Tab. 4 Comparison of efficiency of the model of LSTM with different parameters

| 模型参数 | 训练时间/min | 模型大小 | 每词解码时间/ms |
|------------------------|----------|----------|-----------|
| LSTM 层数 = 2, 单元数 = 64 | 28 | 528.6 KB | 35 |
| LSTM 层数 = 2, 单元数 = 512 | 155 | 42.2 MB | 242 |
| LSTM 层数 = 3, 单元数 = 64 | 37 | 738.1 KB | 41 |
| LSTM 层数 = 3, 单元数 = 512 | 169 | 42.4 MB | 260 |

为了分析训练数据规模对系统性能的影响, 本文通过改变训练集的规模, 使用 3 层 512 单元的 LSTM 网络对发音模型进行训练, 并分别在集内词和集外词测试集上验证词形正确率, 结果如图 5 所示。随着训练数据规模的逐渐增大, 词形正确率逐渐提高; 在相同训练规模的情况下, 集内词的标音正确率高于集外词的标音正确率。

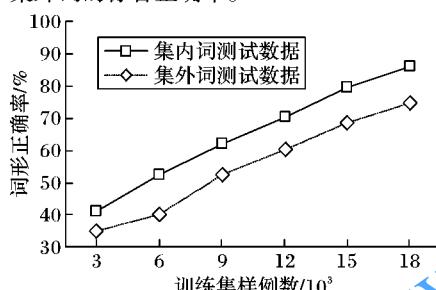


图 5 训练规模与系统性能的关系

Fig. 5 Relationship between training data and system performance

5.3 实验分析

实验结果表明, 增加 LSTM 模型的层数和单元数能够提升系统性能, 当网络层数由 2 层增加至 3 层时对模型的影响较小; 当单元数由 64 增加至 512 时, 模型的大小、训练时间、系统性能都会大幅提高, 但同样会导致系统效率大幅下降。当 LSTM 模型的层数为 3, 单元数为 512 时系统性能最佳, 与 Phonetisaurus 方法相比, 音素正确率提升了 2.3 个百分点, 词形正确率提升了 9.5 个百分点。此外, 训练语料的规模也会对系统性能造成影响, 随着训练规模的增大, 系统性能会逐渐提升。但同时可以发现, 在训练语料有限的情况下, 集外词的标音正确率始终和集内词存在 10% 个百分点的差距。因此, 为了提升标音系统的总体性能, 还需进一步扩充俄语发音词典, 提高模型准确率, 扩大集内词的覆盖范围。

6 系统应用前景

本系统基于 TensorFlow 框架, 实现了基于 LSTM 序列到序列模型的交互式俄语词汇自动标音功能。本系统的开发是理论方法到工程应用的实践过程, 可在诸多实际问题上得到应用:

1) 我国与俄罗斯等地区在军政、外交、文化等领域的沟通交流日益密切, 本系统可以为不懂俄语的使用者提供语言帮助。

2) 本系统可以帮助俄语学习者拼读单词, 判断单词的重音位置, 掌握辅音软硬、元音强弱等变化规律, 在俄语学习中起到辅助支持的作用。

3) 本系统可以嵌入俄语语音识别和语音合成系统, 通过快速、准确、实时的俄语字音转换, 摆脱对发音词典的依赖, 降低内存空间的占用率。

此外, 本系统还具有以下特点:

1) 系统基于 TensorFlow 框架和 Python 语言开发, 具有操作系统的移植性, 可以在 Linux、Windows、Android、IOS 系统间实现移植。

2) 系统基于序列到序列模型算法, 可根据训练数据进行语种间的移植, 实现多语种词汇标音的功能。

3) 系统可根据训练数据的质量优化发音模型, 为标音准确率的提升提供可能性; 并且能够根据音素集的选择, 改变发音的标注形式。

7 结语

词汇标音技术能够为俄语语音合成和语音识别系统的构建提供关键支持。本文首先设计了基于 SAMPA 的俄语音素集, 在原音素集的基础上增加了重音符号及 4 个弱化元音, 并基于此音素集构建了包含 20000 词的俄语发音词典。在此基础上, 本文设计并实现了基于 TensorFlow 的俄语词汇标音系统, 系统使用了基于 LSTM 序列到序列的模型算法。在实验测试中, 集内词和集外词的音素正确率分别达到了 99.2% 和 94.5%, 词形正确率分别达到了 95.8% 和 74.8%, 均高于 Phonetisaurus 方法。实验结果表明, 基于 LSTM 的序列到序列模型在俄语字音转换问题上取得了出色的表现, 该系统能够有效应用为俄语发音词典的建设提供支持。但在训练语料有限的情况下, 系统对集外词的标音正确率与集内词存在一定差距, 还有待进一步提高。因此在以后的工作中, 需要进一步扩充俄语发音词典, 扩大训练语料的规模, 为模型准确率的提高探寻途径。

参考文献(References)

- [1] KARPOV A, MARKOV K, KIPYATKOVA I, et al. Large vocabulary Russian speech recognition using syntactico-statistical language modeling[J]. Speech Communication, 2014, 56(1): 213–228.
- [2] KIPYATKOVA I, KARPOV A, VERKHODANOVA V, et al. Analysis of long-distance word dependencies and pronunciation variability at conversational Russian speech recognition[J]. Computer Science and Information Systems, 2012, 11(6): 719–725.
- [3] JIAMPOJAMARN S, KONDRAK G, SHERIF T. Applying many-to-many alignments and hidden Markov models to letter-to-phoneme conversion[C]// Human Language Technologies: Proceedings of the North American Chapter of the Association of Computational Linguistics. New York: NAACL-HLT, 2007: 372–379.
- [4] BISANI M, NEY H. Joint-sequence models for grapheme-to-phoneme conversion[J]. Speech Communication, 2008, 50(5): 434–451.
- [5] NOVAK J R, MINEMATSU N, HIROSE K. WFST-based grapheme-to-phoneme conversion: open source tools for alignment, model-building and decoding[EB/OL]. [2017-05-10]. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.361.9764>.
- [6] GRAVES A. Generating sequences with recurrent neural networks[EB/OL]. [2017-05-10]. <https://arxiv.org/pdf/1308.0850.pdf>.



- [7] BAHDANAU D, CHO K, Bengio Y. Neural machine translation by jointly learning to align and translate[EB/OL]. [2017-05-10]. <https://arxiv.org/abs/1409.0473>.
- [8] SUTSKEVER I, VINYALS O, LE Q V. Sequence to sequence learning with neural networks[C]// NIPS 2014: Proceedings of the 27th International Conference on Neural Information Processing Systems. Cambridge, MA: MIT Press, 2014, 2: 3104–3112.
- [9] YAO K, ZWEIG G. Sequence-to-sequence neural net models for grapheme-to-phoneme conversion[EB/OL]. [2017-05-10]. <https://arxiv.org/abs/1506.00196>.
- [10] Wikipedia . IPA symbol for Russian pronunciations [EB / OL] . [2017-10-17]. https://en.wikipedia.org/wiki/Help:IPA_for_Russian.
- [11] WELLS J C. SAMPA computer readable phonetic alphabet [C] // Handbook of Standards and Resources for Spoken Language Systems. Berlin: Walter de Gruyter, 1997.
- [12] OTANDER J. CMU sphinx [EB/OL]. (2017-04-26) [2017-10-17]. <https://cmusphinx.github.io/wiki/download/>.
- [13] 信德麟, 张会森, 华勤. 俄语语法[M]. 2 版. 北京: 外语教学与研究出版社, 2009: 1 – 92. (XIN D L, ZHANG H S, HUA S. Russian Grammar(Second Edition) [M]. Beijing: Foreign Language Teaching and Research Press, 2009: 1 – 92.)
- [14] 喻俨, 莫瑜. 深度学习原理与TensorFlow实践[M]. 北京: 电子工业出版社, 2017: 128 – 139. (YU Y, MO Y. Deep Learning Principle and TensorFlow Practice [M]. Beijing: Publishing House of Electronics Industry, 2017: 128 – 139.)
- [15] HOCHREITER S, SCHMIDHUBER J. Long short-term memory [J]. Neural Computation, 1997, 9(8): 1735 – 1780.
- [16] GIMPEL K, SMITH N A. Softmax-margin CRFs: training log-linear models with cost functions [C] // Human Language Technologies: Proceedings of the North American Chapter of the Association of Computational Linguistics. Los Angeles: DBLP, 2010: 733 – 736.
- [17] CHO K, van MERRIENBOER B, GULCEHRE C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[EB/OL]. [2017-05-10]. <https://arxiv.org/abs/1406.1078>.
- [18] KOEHN P. Pharaoh: a beam search decoder for phrase-based statistical machine translation models[C]// AMTA 2004: Proceedings of the 6th Conference of the Association for Machine Translation in the Americas. Berlin: Springer, 2004: 115 – 124.
- [19] WILLIAMS R J, PENG J. An efficient gradient-based algorithm for on-line training of recurrent network trajectories[J]. Neural Computation, 1990, 2(4): 490 – 501.
- [20] ABADI M, BARHAM P, CHEN J, et al. TensorFlow: a system for large-scale machine learning [C] // Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation. Savannah, GA: USENIX, 2016: 265 – 283.
- [21] PETERS B, DEHDARI J, van GENABITH J. Massively multilingual neural grapheme-to-phoneme conversion[C] // Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems. Copenhagen: EMNLP, 2017: 19 – 26.
- [22] 滕飞, 郑超美, 李文. 基于长短句记忆多维主题情感倾向性分析模型[J]. 计算机应用, 2016, 36(8): 2252 – 2256. (TENG F, ZHENG C M, LI W. Multidimensional topic model for oriented sentiment analysis based on long short-term memory[J]. Journal of Computer Applications, 2016, 36(8): 2252 – 2256.)
- [23] HANNEMANN M, TRMAL J, ONDEL L, et al. Bayesian joint-sequence models for grapheme-to-phoneme conversion [EB/OL]. [2017-05-10]. http://www.fit.vutbr.cz/research/groups/speech/publi/2017/hannemann_icassp2017_0002836.pdf.
- [24] TSVETKOV Y, SITARAM S, FARUQUI M, et al. Polyglot neural language models: a case study in cross-lingual phonetic representation learning[EB/OL]. [2017-05-10]. <https://arxiv.org/abs/1605.03832>.
- [25] MILDE B, SCHMIDT C, KÖHLER J. Multitask sequence-to-sequence models for grapheme-to-phoneme conversion [EB/OL]. [2017-05-10]. http://www.isca-speech.org/archive/Interspeech_2017/pdfs/1436.PDF.

This work is partially supported by the Project of Social Science Planning of Luoyang (2016B285).

FENG Wei, born in 1993, M. S. candidate. His research interests include natural language processing.

YI Mianzhu, born in 1964, Ph. D., professor. His research interests include computational linguistics, language information processing.

MA Yanzhou, born in 1977, Ph. D., associate professor. His research interests include Computational Linguistics, phonetic information processing.

(上接第 959 页)

- [12] ZHANG Y, FU P, LIU W, et al. Imbalanced data classification based on scaling kernel-based support vector machine [J]. Neural Computing and Applications, 2014, 25(3/4): 927 – 935.
- [13] GUO H, LIU H, WU C, et al. Logistic discrimination based on G-mean and F-measure for imbalanced problem [J]. Journal of Intelligent and Fuzzy Systems, 2016, 31(3): 1155 – 1166.
- [14] ALCALÁ-FDEZ J, FERNANDEZ A, LUENGO J, et al. KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework [J]. Journal of Multiple-Valued Logic and Soft Computing, 2011, 17(2/3): 255 – 287.
- [15] OLIVEIRA G V, COUTINHO F P, CAMPELLO R J G B, et al. Improving k -means through distributed scalable metaheuristics [J]. Neurocomputing, 2017, 246: 45 – 57.
- [16] BERKHIN P. A survey of clustering data mining techniques [J]. Grouping Multidimensional Data, 2006, 43(1): 25 – 71.
- [17] RUI XU, DONALD C. WUNSCH II. Survey of clustering algorithms[J]. IEEE Transactions on Neural Networks, 2005, 16(3): 645 – 678.
- GUO Huaping**, born in 1982, Ph. D., associate professor. His research interests include machine learning, data mining.
- ZHOU Jun**, born in 1984, M. S. candidate. His research interests include machine learning.
- WU Chang'an**, born in 1959, M. S., professor. His research interests include pattern recognition, image processing.
- FAN Ming**, born in 1948, M. S., professor. His research interests include machine learning, data mining, database.