**Similar pair document**

Background

 由于有些词在目标使用场景下会出现，却在训练数据中没有或较少出现（如新词，生僻词），导致训练出的语言模型不能很好识别这些词。在这种情况下，需要一种方法对现有语言模型扩充，以提高对这些词的识别效果。由于5gram\_1e-9的语言模型参与解码会消耗过大内存，需要使用biglm decoder。在这种情况下，扩充后的语言模型需要与biglm decoder匹配。

Introduction

 Similar pair是一个用给定的词扩充现有语言模型的工具。如果现有语言模型不能很好识别某些词，却能较好识别与它们用法相似的同类词，使用similar pair可以提高对这些词的识别效果。对于给定语言模型，需要在其中人工选择一个定位词，再让人工整理的新词词表中的词共享定位词在原语言模型中的位置和权值，构建新语言模型，新语言模型就能识别出新词词表中的词。本工具附带兼容similar pair 的biglm decoder，可以对较大语言模型解码进行similar pair变换并进行biglm解码。使用该biglm decoder解码不影响对新词词表中的词的识别。使用本工具需提供原语言模型及其发音词典，新词词表及其发音词典，定位词。本工具将输出新语言模型及其发音词典。Similar pair输入和输出的语言模型均为FST格式，所有输入输出FST语言模型均配有相应的label词表。对于较大的语言模型，本工具在test case中提供ARPA2FST语言模型转换和biglm decoder代码。需要额外输入ARPA格式3gram\_1e-5和5gram\_1e-9语言模型（test case中为3gram\_1e-5，3gram\_1e-6），经similar pair变换后，得到输出和biglm解码结果。

Detail

1. 用新词词表构造FST。在这个FST中，新词词表中的每个词构成一条边，所有的边共用一个初始状态和一个终止状态。
2. 在原语言模型中挑选合适的定位词，并在定位词所在的每一条边的位置插入一条新的边，作为tag label。其weight值等于当前定位词的weight值。定位词决定新加入词将要出现在语言模型中的位置和基准weight值，因此定位词应该与新词词表中的词用法尽量相似，其在训练数据中出现的频率应该与新加入词在目标场景下出现的频率尽量相等。
3. 设定一个merge weight。用新词词表FST替换原语言模型中的每一个tag label，并在新词词表FST和原语言模型连接处插入两条新的边，这两条边的weight值分别为merge weight和当前tag label的weight值。因此，对于语言模型中的每个新加入的词，它的等效weight值由对应定位词的weight值和merge weigh共同决定。Merge weight越小，新语言模型把一个词识别成新词词表中的词的概率越高。由于定位词在训练数据中出现的频率与新加入词在目标场景下出现的频率可能不同，因此需要通过调节merge weight来补偿这种偏差。当merge weight为0时，新语言模型识别成新加入词的概率等于定位词在训练数据中出现的频率，可以根据实际情况，用枚举法调节merge weight，直到识别效果最好为止。

Scripts

1. similar\_general.sh

用法：./similar\_general.sh --position\_word --auxiliary\_dir --auxiliary\_name --auxiliary\_lex --source\_dir --tagged\_source\_dir --source\_lex --merge\_weight --similar\_output\_dir

该脚本可对常规大小语言模型进行similar pair变换，加入新词词表，得到新的FST语言模型。positon\_word是包含定位词的给定文件。auxiliary\_dir是给定新词词表所在文件夹。auxiliary\_name是给定新词词表文件名。即{auxiliary\_dir}/{auxiliary\_name}为给定新词词表。auxiliary\_lex是给定新词词表对应的发音词典。source\_dir是存放给定FST语言模型的目录。tagged\_source\_dir是一个中间变量目录，用来存放和调取给定FST的加入tag label的中间体FST，不可与source\_dir相同。source\_lex是给定FST语言模型对应的发音词典。merge\_weight是调节新加入词在原语言模型中权重的参数。similar\_output\_dir是加入新词后的FST语言模型输出目录。

1. similar\_biglm.sh

用法：./similar\_biglm.sh --position\_word --auxiliary\_dir --auxiliary\_name --auxiliary\_lex --small\_source\_dir --big\_source\_dir --source\_lex --tagged\_small\_source\_dir --tagged\_big\_source\_dir --merge\_weight --small\_similar\_output\_dir --big\_simialr\_output\_dir

该脚本用于较大剪枝率语言模型（5gram\_1e-9）的情况。功能同similar\_general.sh。但它要求输入两个FST语言模型（如同一原始语言模型剪枝出的3gram\_1e-5和5gram\_1e-9），分别将新词词表中的词插入其中，得到两个新的FST语言模型。其中small\_source\_dir是存放给定小剪枝率FST语言模型的目录。big\_source\_dir是存放给定大剪枝率FST语言模型的目录。tagged\_small\_source\_dir是存放加入tag label的小FST的中间变量目录。tagged\_big\_source\_dir是存放加入tag label的大FST的中间变量目录。small\_simialr\_output\_dir是给定小FST加入新词后的FST输出目录。big\_similar\_output\_dir是给定大FST加入新词后的FST输出目录。其他参数与similar\_general.sh等价。

Other files

1. 定位词文件

保存定位词的文件，须事先准备好。文件名被保存在similar pair配置文件中，间接被相应脚本调用。格式参见testcase/general/position\_word.txt。

1. Path.sh

配置本工具的环境变量，本工具自带。

1. toolkit/

similar pair需要的支持文件，由本工具自带。

1. Decode/

Biglm decoder需要调用的文件，由本工具自带。

1. Testcase

本工具的testcase分为一般情况和biglm情况，使用时直接运行run\_testcase\_general.sh和run\_testcase\_biglm.sh文件。其中biglm的两个语言模型应为3gram\_1e-5和5gram\_1e-9。为节省演示时间，此处使用了3gram\_1e-5和3gram\_1e-6模型。数据和输出目录在testcase/中，每一种情况的主要输出结果都应该与expected\_output相同。

Notice

1. 本工具依赖kaldi\_201609，如果没有安装，请参照README进行安装。
2. 新词词表中的词可以是原语言模型词表中已经存在的词，只要其在训练数据中出现频率不够高，就可以加入新词词表。这部分词不用加入新词发音词典中。
3. 本工具会使用openfst的fstcompile和fstprint，这两个命令会打乱FST 原有state ID顺序，自行调试时请注意。
4. 整理新词词表选择定位词的原则可参考wiki中的Similar\_pair\_实验报告.docx。