**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，本工具提供ARPA2FST语言模型转换脚本和biglm decoder。需要额外输入ARPA格式3gram\_1e-5和5gram\_1e-9语言模型，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. gen\_fst.sh

用法：./gen\_fst.sh <similar\_config>

该脚本可以用给定的新词词表构造FST。<similar\_config>是给定的similar pair配置文件，其中的auxiliary\_dir是新词词表路径，auxiliary\_name是新词词表文件名。利用这个文件可以构造一个FST，命名为{auxiliary\_name}.txt，并保存在上述路径下。

1. add\_tag.sh

用法：./add\_tag.sh <similar\_config> <source\_dir> <tagged\_source\_dir>

该脚本可以在原语言模型中定位词位置上插入tag label。给定语言模型需为FST格式，并需与其附属的label词表一起保存在<source\_dir>中。<similar\_config>是给定的similar pair配置文件，其中的position\_word是存储了定位词的文件，tag\_label是tag label样式。在给定语言模型中指定定位词位置上插入指定tag label，得到带tag label的FST语言模型，并生成相应的label词表，保存在<tagged\_source\_dir>中。

1. g\_merge.sh

用法：./g\_merge.sh <similar\_config> <merge\_weight> <tagged\_source\_dir> <output\_dir>

该脚本可以将新加入词的FST插入到语言模型中tag label位置处，得到新语言模型。带tag label的语言模型已存在于<tagged\_source\_dir>中。<similar\_config>是给定的similar pair配置文件，其中tag\_label是tag label，tag\_eps是每个插入的新词词表FST与原语言模型连接处没有发音的两条边上的label，auxiliary\_dir和auxiliary\_name是保存新词词表FST的路径和新词词表FST的文件名，source\_lex是原语言模型对应的发音词典，auxiliary\_lex是新词词表的发音词典。利用这些文件和参数可以生成合并后的新FST，并生成相应的发音词典和label词表，保存在<output\_dir>中。

1. g\_merge\_biglm.sh

用法：./g\_merge\_biglm.sh <similar\_config> <merge\_weight> <tagged\_source\_dir> <output\_dir>

该脚本用法与功能和g\_merge.sh相同，适用于原语言模型较大的情形。输出新FST有两个，分别保存在<output\_dir>和<output\_dir>/G\_to\_build\_hclg中。两个FST对应的发音词典和label词表相同，保存在<output\_dir>中。

1. lm\_format.sh

用法：./lm\_format.sh <biglm\_config> <small\_source\_dir> <big\_source\_dir>

该脚本可以将输入的两个不同剪枝率的ARPA语言模型转换为相应的FST格式。<biglm\_config>是较大语言模型格式转换的配置文件，其中small\_lm是给定的ARPA格式3gram\_1e-5语言模型，big\_lm是5gram\_1e-9的语言模型，lm\_lex是语言模型对应的发音词典。<small\_source\_dir>是小语言模型FST格式输出路径，<big\_source\_dir>是大语言模型FST格式输出路径，两个语言模型的输出均为一个FST和相应的label词表。

1. similar\_biglm\_decoder.sh

用法：./similar\_biglm\_decoder.sh <biglm\_config> <small\_output\_dir> <big\_output\_dir> <biglm\_decode\_dir> <testset>

该脚本可以将两个不同剪枝率的语言模型输入g\_merge\_biglm.sh后的输出结果利用biglm decoder解码。<biglm\_config>是较大语言模型解码的配置文件，其中ph是phone表，am是解码用声学模型。<small\_output\_dir>是小语言模型经过g\_merge\_biglm.sh的输出路径，<big\_output\_dir>是大语言模型经过g\_merge\_biglm.sh的输出路径，其中都有FST格式的经过similar pair变换的语言模型及其对应的label词表和发音词典。利用这些可以分别构造解码图，然后利用biglm decoder解码。测试集是<testset>，解码路径是<biglm\_decode\_dir>。

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example/selected\_dish.txt。

1. similar pair配置文件

配置similar pair的一些变量，须事先准备好，并在使用相应脚本时提供。格式参见example/menu.cfg。

1. Biglm配置文件

配置针对较大语言模型格式转换和解码的一些变量，须事先准备好，并在使用相应脚本时提供。格式参见example/test.cfg。

1. Path.sh

配置本工具的环境变量，本工具自带。本工具需要声明kaldi下的多处环境变量，如果kaldi的路径与文件中不符，请修改此文件。

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。