软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2012,23(12):3101−3114 [doi: 10.3724/SP.J.1001.2012.04208]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

一种面向统计机器翻译的协同权重训练方法*

刘树杰[1+], 李志灏[2], 李 沐[2], 周 明[2]

[1](哈尔滨工业大学 计算机科学与技术学院,黑龙江 哈尔滨 150001)

[2](微软亚洲研究院,北京 100080)

Co-Training Framework for Feature Weight Optimization of Statistic Machine Translation

LIU Shu-Jie[1+], LI Chi-Ho[2], LI Mu[2], ZHOU Ming[2]

[1](School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

[2](Microsoft Research Asia, Beijing 100080, China)

+ Corresponding author: E-mail: j4ckl1u@gmail.com

Liu SJ, Li CH, Li M, Zhou M. Co-Training framework for feature weight optimization of statistic machine translation. *Journal of Software*, 2012,23(12):3101−3114 (in Chinese). http://www.jos.org.cn/1000-9825/4208.htm

Abstract: In this paper, based on the investigation of domain adaptation for feature weight, the study proposes to use a co-training framework to handle domain adaptation for feature weight, i.e. The study uses the translation results from another heterogeneous decoder as pseudo references and adds them to the development data set for minimum error rate training to bias the feature weight to the domain of test data set. Furthermore, the study uses a minimum Bayes- Risk combination for pseudo reference selection, which can pick proper translation results from the translation candidates from both decoders to smooth the training process. Experimental results show that this co-training method with a minimum Bayes-Risk combination can yield significant improvements in target domain.

Key words: statistical machine translation; minimum error rate training; domain adaptation; co-training; minimum Bayes-risk combination

摘 要: 分析了统计机器翻译中的特征权重的领域自适应问题,并针对该问题提出了协同的权重训练方法.该方法使用来自不同解码器的译文作为准参考译文,并将其加入到开发集中,使得特征权重的训练过程向测试集所在的领域倾斜.此外,提出了使用最小贝叶斯风险的系统融合方法来选择准参考译文,进一步提高了协同权重训练的性能.实验结果表明,使用最小贝叶斯风险系统融合的协同训练方法,可以在一定程度上解决特征权重的领域自适应问题,并显著地提高了在目标领域内机器翻译结果的质量.

关键词: 统计机器翻译;最小错误率训练;领域自适应;协同训练;最小贝叶斯风险系统融合

中图法分类号: TP391 **文献标识码**: A

机器翻译(machine translation)指使用计算机自动地将一种语言的文字翻译到另一种语言的文字,是人工智能领域重要且具有挑战性的分支之一[1].在 1991 年,Brown 等人提出了基于统计的机器翻译方法[2],并在实验中取得了初步的成功,引起了广泛的关注.从此,统计机器翻译(statistical machine translation)成为了机器翻译的主

* 收稿时间: 2011-09-01; 修改时间: 2012-01-16; 定稿时间: 2012-03-15

流方法.近几年,逐渐出现了一些实用的统计机器翻译系统,如微软在线翻译、谷歌在线翻译、百度在线翻译.

目前流行的统计机器翻译方法是由 Och 和 Ney 在 2002 年提出的对数线性模型(log-linear model),该模型融合了翻译模型、语言模型等各种特征[3],其中,翻译模型用来刻画译文对源语言句子的忠诚程度,即译文是否违背了源语言句子本来的意思;而语言模型则用来刻画译文是否流畅.该对数线性模型的形式如下:

$$p(e \mid f) = \frac{\exp(\sum_{i=1}^{N} \lambda_i h_i(e, f))}{\sum_{e'} \exp(\sum_{i=1}^{N} \lambda_i h_i(e', f))} \tag{1}$$

其中,f 表示源语言句子,e'和 e 表示任意可能的目标语言句子,h 表示使用的特征,λ表示特征权重.

从公式(1)中我们可以看到,整个模型中有两部分变量需要训练和优化:

1) 特征 h,包括互译概率、语言模型概率、调序模型概率等.这些概率通常从大规模的双语语料(用来获取翻译概率和调序概率)和大规模的目标语言单语语料(用来获取语言模型概率)统计得到,我们把获取特征的数据集称为训练集;
2) 特征权重λ,通常使用小规模的双语语料优化得到.

为使自动评价系统更加可靠和稳定,该小规模双语语料一般是一个源语言句子对应多于一个不同的目标语言句子.我们把优化特征权重λ的小规模双语语料称为开发集.除了训练集和开发集,还会有一份或多份类似于开发集的数据集(一个源语言句子对应若干个不同的目标语言句子)用来测试模型的最终性能,称为测试集.

当训练集与测试数据(或者开发集)来自不同的领域时,在训练集上统计得到的特征 h 可能并不适合测试集(或者开发集);同样的,在开发集上优化得到的特征权重λ也会存在不适合测试集的情况.因为领域不同导致的训练好的模型在测试集上的性能不佳现象称为领域自适应问题(domain adaptation problem).在本文中,我们称训练集跟测试集的领域自适应问题为特征的领域自适应问题,并称开发集同测试集的领域自适应问题为特征权重的领域自适应问题.

特征的领域自适应问题和特征权重的领域自适应问题,是统计机器翻译领域自适应问题的两个子问题,这两个子问题相辅相成,共同影响着统计机器翻译在目标领域内的翻译性能[4−7].

针对统计机器翻译的领域自适应问题,很多学者提出了各自的解决方法,这些方法主要集中在特征的领域自适应问题上,并大致可以分为两类:

- 一类方法是领域内的子模型同领域外的子模型相融合的方法[5−8]:这类方法使用领域内的训练数据获得领域内翻译模型和/或语言模型,并使用领域外的训练数据获得领域外翻译模型和/或语言模型,然后将这些翻译模型和/或者语言模型融合在一起,最后通过优化不同训练集上获取的翻译模型和/或者语言模型的权重实现领域自适应的目的;
- 另一类方法则是使用直推式学习生成领域内子模型[9]:该类方法使用已经训练好的模型对大规模领域内的源语言单语语料、开发集或测试集进行解码,从得到的译文中重新训练一个新的翻译模型或者语言模型,并将这个翻译模型或者语言模型作为新的特征融合到公式(1)里,从而将整个模型向测试集所在的目标领域倾斜.

尽管关于特征的领域自适应问题已经存在很多的研究工作,然而对特征权重的领域自适应问题的探索却比较少.Li,Zhao 等人通过开发集选择的方法训练特征权重,从而解决特征权重的领域自适应问题:该方法通过特征空间内的相似度从开发集中选择领域相近的句子构成领域内的开发集,并用该领域内的开发集来训练特征权重,从而实现领域自适应的目的[10].

不同于开发集选择的方法,本文并没有选择部分开发集参与特征权重的训练,而是将测试集的准参考译文添加到开发集中,以增加目标领域内训练样本的数量,使得训练过程偏向于测试集所在的目标领域.在协同训练的框架下,两个不同解码器使用来自对方解码器的译文作为准参考译文,将测试集添加到开发集中,并重新进行特征权重的训练,两个解码器相互协同,逐步提高测试数据集的译文质量.为选择更适合作为准参考译文的译文候选添加到开发集中,我们使用最小贝叶斯风险的系统融合方法来将两个解码器的译文进行系统融合,并将重排序后的最好译文候选作为准参考译文,进一步提高了目标领域内机器翻译结果的质量.

本文的创新点如下:

1) 首次使用了协同训练的方法解决统计机器翻译特征权重的领域自适应问题;

2) 使用最小贝叶斯风险系统融合的方法来选择准参考译文,使得协同训练的训练过程更加平稳.

1 特征权重的领域适应问题

在这一节,我们通过一组中-英翻译的实验,来说明特征权重的领域自适应问题是如何影响统计机器翻译系统在测试集上的性能的.在实验中,我们使用了两个不同的主流解码器:第 1 个是基于括号转换文法(bracketing transduction grammar,简称 BTG)的解码器[11],其使用了最大熵词汇调序模型[12];第 2 个是基于层次短语的解码器(hierarchical phrase based decoder,简称 Hiero)[13].我们使用的双语训练集是 NIST 训练数据的一部分(具体 LDC 号见表 1).翻译模型是通过使用 Giza++对双语语料进行双向词汇对齐,并使用 Grow-Diag-Final 的方法进行对称融合[14],最后采用标准短语抽取方法获取的[11];(BTG 解码器的)词汇调序模型[12]是使用张乐博士的最大熵训练工具包(http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html)在同样的双语训练集上训练得到的;语言模型是在 Gigaword 数据集训练得到的 5-Gram 语言模型.

Table 1 LDC numbers of training data

表 1 训练数据 LDC 号列表

LDC	语料描述
LDC2003E07	Ch/En treebank par corpus
LDC2003E14	FBIS multilanguage texts
LDC2005T06	Ch news translation text part 1
LDC2005T10	Ch/En news magazine par text
LDC2005E83	GALE Y1 Q1 release—Translations
LDC2006E26	GALE Y1—En/Ch par financial news
LDC2006E34	GALE Y1 Q2 release—Translations V2.0
LDC2006E85	GALE Y1 Q3 release—Translations

我们使用的开发集有两个:一是 NIST 2003 年的评测集(NIST03),另一个是 NIST 2006 年的评测集的 Web 部分(NIST06Web).我们使用的测试集是 NIST 2005 年的评测集(NIST05)和 NIST 2008 年的评测集的 Web 部分(NIST08Web).NIST03 和 NIST05 都是新闻数据,属于同一领域;而 NIST 2006/2008 年的评测集包含了一部分新闻数据和一部分网络挖掘的数据(Web 数据).为了保持数据集领域的单一性,我们只选择了 NIST 2006/2008 的 Web 数据部分进行实验.训练集、开发集和测试集的统计结果见表 2.

Table 2 Data set statistic

表 2 数据集统计

	训练集	NIST03	NIST05	NIST06Web	NIST08Web
句子数	497 996	919	1 082	479	666
单词数	12 042 230	23 959	29 423	9 246	14 711

NIST03 和 NIST06Web 分别作为开发集,并使用最小错误率训练[15]来训练特征权重;然后,用 NIST05 和 NIST08Web 来测试统计机器翻译的性能(评价指标为 BLEU4).两个解码器的翻译结果见表 3.

Table 3 Performance of BTG/Hiero on different test data set

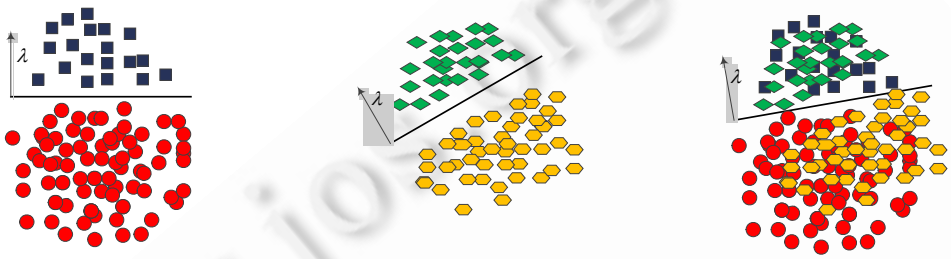
表 3 在不同开发集和测试集上的两种解码器的性能

开发集	BTG		Heiro	
	NIST05	NIST08Web	NIST05	NIST08Web
NIST03	36.06	18.47 (–3.29)	36.51	19.82 (–1.51)
NIST06Web	33.90 (–2.16)	21.76	33.23 (–3.28)	21.33

从表 3 中我们可以看到:对于 NIST05 的测试集,来自同一领域的 NIST03 更适合作为开发集;当我们使用来自不同领域的 NIST06Web 作为开发集时,其翻译的性能会有明显的下降(在 BTG 解码器上下降了 2.16,在 Hiero

解码器上下降了 3.28).同样地,当采用来自不同领域的 NIST03 作为开发集时,对于 NIST08Web 测试集,则分别下降了 3.29 和 1.51.

我们使用图 1 解释这种特征权重的领域自适应问题.在图 1(a)中:正方形点表示正例,在统计机器翻译中通常指的是候选译文中的最佳译文(在这里,最佳译文指的是能够得到最高 BLEU 的译文);圆形点代表负例,在统计机器翻译中通常指的是候选译文中去掉最佳译文的那些译文;无箭头线段表示分类面,带箭头线段表示在开发集上训练得到的特征权重向量λ.而图 1(b)显示了测试集上的正负例分布情况(棱形表示正例,六边形表示负例).我们发现,由于领域不一致,造成了测试集上的样本分布跟开发集上的样本分布显著不同,从而在测试集上的最优分类面也与开发集上的最优分类面存在较大的差异.所以我们需要寻找一个折衷的解决方案,使得在尽可能正确处理开发集的情况下能够照顾测试集的领域不一致情况,比如图 1(c)中的折衷分类面.



(a) 开发数据集上的最佳分类面 (b) 测试数据集上的最佳分类面 (c) 平衡开发数据集和测试数据集的分类面

Fig.1 Decision boundary for development and test data sets when domains are different

图 1 开发集和测试集来自不同领域造成的分类面不一致情况

为了解决特征权重的领域自适应问题,我们将在第 2 节介绍一种协同训练的学习方法,该学习方法使用两个分类器来相互辅助,从而可以利用测试集上的信息来实现领域自适应的目的.

2 协同训练方法

协同训练方法是由 Blum 和 Mitchell 在 1998 年提出的[16],并被广泛应用于自然语言处理的各个领域:Pierce 和 Cardie 将协同训练算法应用于名词短语识别,将识别错误率降低了 36%[17];Sarkar 和 Steedman 将协同训练方法应用于统计句法分析,句法分析的准确率和召回率均得到了显著的提高[18];Hwa 等人提出了一种基于协同训练的主动半监督句法分析方法,该方法可以减少一半的人工标注量[19].

协同训练方法是多视角学习(multi-view learning)的一种,它假设数据集有两个充分冗余(sufficient and redundant)的视角(view),即两个满足下述条件的属性集:

第一,每个属性集都足以描述该问题.也就是说,如果训练样例足够,在每个属性集上都足以训练一个强分类器;

第二,在给定标记时,每一个属性集都条件独立于另一个属性集.

针对第 2 个条件,Wang 和 Zhou 作了进一步的分析,他们证明了:只要两个分类器具有较大的差异,就可以通过协同训练提高性能,而不需要两个属性集的完全条件独立[20].

协同训练方法是一种包容器算法,该包容器算法需要两个分类器 $f^{(1)}$和 $f^{(2)}$.$f^{(1)}$和 $f^{(2)}$可以是任意一种分类算法,只要这两种分类算法存在较大差异,且能够给样本一个置信度即可.该置信度被用来从没有标注的样本中选择那些标注比较可靠的样本添加到其他分类器的开发集中.协同训练的基本思想是,参与协同训练的每个分类器均可以从另一个分类器获得有用的信息(通过来自另一个分类器标注好的样本)帮助自己提高性能,从而使得参与协同训练的两个分类器可以相互学习和更新.

协同训练算法的基本框架如图 2 所示:算法的输入是一个标注的数据集合 $\{(x_i,y_i)\}_{i=1}^{l}$、一个未带标注的数

据集合 $\{x_j\}_{j=l+1}^{l+u}$、一个学习率 k 和每个样本的两个视角:*view*^{1} 和 *view*^{2};首先,我们初始化分类器 $c^{(1)}$和 $c^{(2)}$的开发集 L_1 和 L_2 为标注数据集合 $\{(x_i,y_i)\}_{i=1}^{l}$(第 1 行);然后分别使用 L_1 和 L_2 来训练针对 *view*^{1} 和 *view*^{2} 的两个分类器 $c^{(1)}$和 $c^{(2)}$(第 3 行),并用训练好的 $c^{(1)}$和 $c^{(2)}$对未带标注的数据集合 $\{x_j\}_{j=l+1}^{l+u}$ 作预测(第 4 行),分别从 $c^{(1)}$/$c^{(2)}$的预测结果中选择前 k 个置信度比较高的结果作为准训练样本添加到 L_2/L_1 中,并将对应的样本从未标注数据集合中移除(第 5 行);最后,我们使用新的开发集 L_1 和 L_2 重新训练分类器 $c^{(1)}$和 $c^{(2)}$,并重复这个过程,直到未标注的数据集合为空(第 2 行).

Input: Labeled data: $\{(x_i,y_i)\}_{i=1}^{l}$, unlabeled data: $\{x_j\}_{j=l+1}^{l+u}$, a learning speed k,
each instance has two views: $x_i=[x_i^{(1)},x_i^{(2)}]$;
1. Initially let the training sample be $L_1=L_2=\{(x_i,y_i)\}_{i=1}^{l}$
2. Repeat until unlabeled data is used up:
3. Train a *view*^{1} classifier $c^{(1)}$ from L_1, and a *view*^{2} classifier $c^{(2)}$ from L_2.
4. Classify the remaining unlabeled data with $c^{(1)}$ and $c^{(2)}$ separately.
5. Add $c^{(1)}$'s top k most-confident predictions $(x,c^{(1)})$ to L_2.
Add $c^{(2)}$'s top k most-confident predictions $(x,c^{(2)})$ to L_1.
Remove these from the unlabeled data.

Fig.2 Framework of co-training algorithm

图 2 协同训练算法的框架

3 面向统计机器翻译的协同权重训练方法

本节将协同训练的方法应用到统计机器翻译的特征权重训练中来.首先介绍常用的统计机器翻译的特征权重训练方法.公式(1)给出了给定一个源语言句子 f 时任意可能的候选译文 e 的翻译概率.从所有可能的候选译文中,翻译系统选择最高概率的译文作为最后的输出结果 $\hat{e}$:

$$\hat{e}=\arg\max_e p(e\mid f)=\arg\max_e \frac{\exp(\sum_{i=1}^{N}\lambda_i h_i(e,f))}{\sum_{e'}\exp(\sum_{i=1}^{N}\lambda_i h_i(e',f))}.$$

公式中的分母部分并不影响最终的 $\hat{e}$ 的选择,从而将公式转化为

$$\hat{e}=\arg\max_e \exp(\sum_{i=1}^{N}\lambda_i h_i(e,f))=\arg\max_e(\sum_{i=1}^{N}\lambda_i h_i(e,f)) \quad (2)$$

对式(2)中参数 λ 的训练,通常有感知机[21]、最小错误率训练[15]、MIRA(margin infused relaxed algorithm)[22]等,其中,最小错误率训练是最常用的训练方法,我们将在下一节详细介绍.

3.1 统计机器翻译的最小错误率训练方法

最小错误率训练是由 Och 在 2003 年提出的能够直接以评测指标(通常为 BLEU)为训练目标的参数训练方法[9],该方法的提出,显著地提高了统计机器翻译系统的性能.最小错误率训练方法在参数的搜索空间中寻找一个能够使得错误率最小的解(在统计机器翻译中,最小错误通常指的是最大 BLEU.).给定源语言句子 f 和一个可能的候选译文 e,以及该源语言句子 f 对应的参考译文 r,我们用函数 $E(r,e)$表示错误个数(在本文中使用的衡量错误率的方法为 BLEU).

对于 S 个源语言句子 f_1^s 和它们的候选译文 e_1^s 以及参考译文 r_1^s,错误个数的计算公式为

$$E(r_1^s,e_1^s)=\sum_{s=1}^{S}E(r_s,e_s).$$

在统计机器翻译中,我们使用开发集来训练公式(2)中的特征权重 λ.在开发集中,对于一个源语言句子 f_s,我们的解码器会得到一个 N-best 的候选译文集合 C_s.我们的目标就是优化参数 λ,使得这 S 个句子的候选译文集合中错误最少的译文被选择出来,其优化解 $\hat{\lambda}$ 通过如下公式获得:

$$\hat{\lambda} = \arg\max_{\lambda}\left\{\sum_{s=1}^{S} E(r_s, \hat{e}(f_s, \lambda))\right\},$$

其中,

$$\hat{e}(f_s, \lambda) = \arg\max_{e \in C_s}\left(\sum_{i=1}^{N} \lambda_i h_i(e, f_s)\right).$$

最小错误率训练通过调整对数线性模型的权重,使得系统在开发集上总体 BLEU 得分最高.首先,利用初始参数对开发集的每个句子进行解码,得到该句子的 *N*-Best 候选译文,然后依次调整特征权重.特征权重调整后,每个句子的 *N*-Best 候选译文的得分(特征和特征权重的点积)也相应地发生变化;根据得分重新排列候选译文,并得到新的最好候选译文,从而由这些最好候选译文构成的总体 BLEU 值也相应变化.特征权重的调整采取沿坐标轴下降的算法,即先固定其他维的特征权重,调某一维的特征权重;当在该维上最优后,再依次调整其他维度.由于解码得到的仅仅是 *N*-Best 候选译文而不是全部的候选译文,所以为了能够得到稳定的优化解,通常需要进行多次最小错误率训练,其过程如图 3 所示.我们维护了一个候选译文列表 C_1^s,其初始值为第 1 遍解码的 *N*-Best 列表(第 2 行).通过使用每次最小错误率训练后得到的新的λ来对开发集重新解码(第 6 行),并尽可能地丰富这个候选译文列表(第 4 行);用扩充了的候选译文列表重新进行最小错误率训练(第 5 行),直至得不到新的候选译文为止(第 3 行).

Input: Development data set { f_1^s, r_1^s };
1. Initially let λ=*random*(·) and C_1^s={·}
2. Decode f_1^s using λ to get *N*-Best list: N_1^s
3. While ($C_1^s \neq C_1^s \cup N_1^s$)
4. $C_1^s \neq C_1^s \cup N_1^s$
5. MERT using translation candidates C_1^s to get a new λ
6. Decode f_1^s using the new λ to get *N*-Best list: N_1^s

Fig.3 MERT for statistical machine translation
图 3 统计机器翻译的最小错误率训练

3.2 协同训练框架下的最小错误率训练

本节将协同训练方法应用到统计机器翻译的特征权重训练上.目前我们使用的 BTG 解码器和 Hiero 解码器均是研究领域内被证明比较优秀的解码器,从而能够保证每个解码器的输出结果具有一定的可靠性,也就满足了协同训练方法的第 1 个基本假设.BTG 解码器和 Hiero 解码器是两种类型的解码模型(在本文中,我们使用了两个异构的解码器来进行实验,但是协同训练的框架并不要求解码器必须异构,只要能够保证两个解码器具有较大的差异即可),BTG 使用了最大熵调序模型,而 Hiero 的调序模型是通过层次短语实现的,并没有显式的调序模型.这两种解码器存在着较大的差异,符合协同训练方法的第 2 个基本假设.在满足协同训练方法基本假设的情况下,我们将两个不同的解码器看作类别为可数个的(翻译结果是可数的)两个不同的分类算法,开发集对应于标注数据,测试集对应于未标注数据.

协同训练框架下的最小错误率训练算法如图 4 所示:给定原始开发集{f_1^s, r_1^s}和测试集{$f_{1+t}^{s+t}, r_{1+t}^{s+t}$}(其中,测试集的参考译文部分不参与训练过程,只用来评价最终训练结果的好坏),以及参与协同训练的两个解码器 BTG 和 Hiero;我们首先初始化 BTG 和 Hiero 的开发集 L_{BTG} 和 L_{Hiero} 为原始开发集{f_1^s, r_1^s},并令每次需要更新的数据集 *U* 为测试集(第 1 行);然后针对每个解码器 BTG/Hiero 使用标准的最小错误率训练方法和开发集 L_{BTG}/L_{Hiero} 获得特征权重λ_{BTG}/λ_{Hiero}(第 3 行和第 4 行),用相应的特征权重λ_{BTG}/λ_{Hiero}对测试集 *U* 进行解码以获取测试集的 *N*-Best 候选译文 $Nbest_{BTG}$/$Nbest_{Hiero}$(第 5 行和第 6 行),并从测试集的 *N*-Best 候选译文中选择最好的译文作为测试集的准参考译文 $\hat{U}_{BTG}$ /$\hat{U}_{Hiero}$ (第 7 行和第 8 行),并将其作为额外的开发集,连同原始的开发集一起构成下一轮

协同训练的开发集($L_{BTG}=\{\{f_1^s,r_1^s\}\cup\hat{U}_{Hiero}\}$;$L_{Hiero}=\{\{f_1^s,r_1^s\}\cup\hat{U}_{BTG}\}$)(注意,这里的额外开发集是交换使用的,从而达到了协同训练的目的)(第 9 行);最后,使用更新了的开发集重新获取新的特征权重(第 3 行和第 4 行);该过程循环执行(第 2 行).

Input: Development data: $\{f_1^s,r_1^s\}$, and test data: $\{f_{1+t}^{s+t}\}$, two decoder (BTG and Hiero);
1. Initially, let $L_{BTG}=L_{Hiero}=\{f_1^s,r_1^s\}$, and $U=\{f_{1+t}^{s+t}\}$
2. Repeat:
3. MERT using L_{BTG} to get the feature weight λ_{BTG} for BTG
4. MERT using L_{Hiero} to get the feature weight λ_{Hiero} for Hiero
5. Decoding U with feature weight λ_{BTG} and BTG to get N-best List: $Nbest_{BTG}$
6. Decoding U with feature weight λ_{Hiero} and Hiero to get N-best List: $Nbest_{Hiero}$
7. Select best translations for U from $Nbest_{BTG}$: $\hat{U}_{BTG}$
8. Select best translations for U from $Nbest_{Hiero}$: $\hat{U}_{Hiero}$
9. Let $L_{BTG}=\{\{f_1^s,r_1^s\}\cup\hat{U}_{Hiero}\}$ and $L_{Hiero}=\{\{f_1^s,r_1^s\}\cup\hat{U}_{BTG}\}$

Fig.4 Co-Training for statistical machine translation

图 4 面向统计机器翻译的协同参数训练方法

与标准的协同训练方法不同,我们并不是选择一部分测试集的样本添加到开发集中,而是将所有的测试集连同最好的翻译译文添加到开发集中(选择部分合适的测试数据添加到开发集中参与协同训练,也许能得到更好的性能提高.但是鉴于篇幅,本文不作进一步的讨论,该问题会在将来的工作中做深入的分析),并每次更新测试集的译文.协同训练也不是像标准的协同训练算法那样当测试集为空时停止,而是使用了一个固定的循环次数,在本文的实验中,我们进行了 10 次循环.在这儿,我们所说的最好译文指的是 *N*-Best 列表中最好的前 4 个翻译译文.除了使用最好的前 4 个候选译文作为准参考译文的方法外,在下一节,我们将介绍最小贝叶斯风险系统融合来选择准参考译文的协同训练方法.

3.3 最小贝叶斯风险的准参考译文选择

最小贝叶斯风险的准参考译文选择使用的是句子级的最小贝叶斯风险系统融合[23]:当我们得到 BTG 和 Hiero 系统的 *N*-Best 候选译文后,我们使用贝叶斯风险重排序的方法将两个 *N*-Best 翻译列表中的候选译文重新打分排序,然后选择得分最高的候选译文作为准参考译文.

给定一个源语言句子,最小贝叶斯风险系统融合使用如下公式对各个系统的 *N*-Best 翻译列表中的翻译 e 进行打分:

$$mbr(e)=\sum_{e'}P(e\mid f)L(e,e'),$$

其中,e'是搜索空间中的任意候选译文,通常情况下,我们很难得到搜索空间中所有的候选译文,所以我们使用参与融合的各个系统 *N*-Best 列表代替;$P(e|f)$是源语言句子 f 翻译为目标语言句子 e 的翻译概率,使用 MBR 重排序时,通常使用翻译系统对候选译文的总体打分来近似.当 $P(e|f)$不可获得或不可比较时,可以假设其服从均匀分布(由于 BTG 和 Hiero 打分不可直接比较,所以在本文的实验中,$P(e|f)$使用的是均匀分布);$L(e,e')$是损失函数(这里使用句子级的 BLEU),该损失函数起到了增益函数的作用:其值在 0~1 之间,越大的值表示越高的相似度.最小贝叶斯风险的系统融合从各个系统的 *N*-Best 候选译文中按照相似度进行重新排序,那些跟其他翻译相似的候选译文能够得到更高的打分(具有更小的贝叶斯风险).

如图 5 所示:不同于图 4 的方法从各个解码器的 *N*-Best 翻译列表中选择最好的译文,使用了最小贝叶斯风险的系统融合方法是将各个系统的 *N*-Best 翻译列表进行融合和重排序来得到新的 *N*-Best 翻译列表 $Nbest_{Comb}$(第 7 行),并从新的 *N*-Best 翻译列表 $Nbest_{Comb}$ 中选择最好的译文 $\hat{U}_{Comb}$ 作为准参考译文(第 8 行),最后将准参考译文 $\hat{U}_{Comb}$ 和测试集作为额外的开发集连同原始的开发集 $\{f_1^s,r_1^s\}$ 一同构成各个解码器的开发集 L_{BTG} 和 L_{Hiero}(不同于第 3.2 节协同训练中两个解码器的开发集使用不同的准参考译文,在最小贝叶斯风险的准参考译文选

择后,两个解码器的开发集是完全一样的)(第 9 行),并进行下一轮的最小错误率训练(第 3 行).需要指出的是,$\hat{U}_{Comb}$ 里的准参考译文既有来自 BTG 解码器的译文,也有来自 Hiero 解码器的译文.

Input: Development data: $\{f_1^s, r_1^s\}$, and test data: $\{f_{1+t}^{s+t}\}$, two decoder (BTG and Hiero);

1. Initially, let $L_{BTG} = L_{Hiero} = \{f_1^s, r_1^s\}$, and $U = \{f_{1+t}^{s+t}\}$
2. Repeat:
3. MERT using L_{BTG} to get the feature weight λ_{BTG} for BTG
4. MERT using L_{Hiero} to get the feature weight λ_{Hiero} for Hiero
5. Decoding U with feature weight λ_{BTG} and BTG to get N-best List: $Nbest_{BTG}$
6. Decoding U with feature weight λ_{Hiero} and Hiero to get N-best List: $Nbest_{Hiero}$
7. MBR combination of $Nbest_{BTG}$ and $Nbest_{Hiero}$ to get $Nbest_{Comb}$
8. Select best translations for U from $Nbest_{Comb}$: $\hat{U}_{Comb}$
9. Let $L_{BTG} = L_{Hiero} = \{\{f_1^s, r_1^s\} \cup \hat{U}_{Comb}\}$

Fig.5 Co-Training for statistical machine translation with MBR combination

图 5 最小贝叶斯风险系统融合的协同参数训练方法

4 实验结果和分析

4.1 协同训练的实验结果及分析

我们使用的训练集、开发集和测试集与第 1 节描述的一样.如前所述,NIST03 和 NIST05 数据集均为新闻语料,而 NIST06Web 和 NIST08Web 则为网络语料.我们使用 4 份语料进行了两组实验,这两组实验的实验配置见表 4:第 1 组实验设置使用的是 NIST03(新闻语料)作为开发集,并使用 NIST06Web(网络语料)参与协同训练,最终在 NIST08Web(网络语料)上测试协同训练的效果;第 2 组实验设置使用的是 NIST06Web(网络语料)作为开发集,并使用 NIST03(新闻语料)参与协同训练,并最终在 NIST05(新闻语料)上测试协同训练的效果.

Table 4 Two settings for co-training

表 4 协同训练的两组设置

	开发集	协同训练的测试集	最终测试集
实验设置 1	NIST03	NIST06Web	NIST08Web
实验设置 2	NIST06Web	NIST03	NIST05

我们分别使用了协同权重训练方法(如图 4 所示)和最小贝叶斯风险系统融合的协同权重训练方法(如图 5 所示),其结果见表 5 和表 6.其中,Baseline 指的是表 3 中的结果,Co-Train 指的是使用面向统计机器翻译的协同权重训练(如图 4 所示)的结果,MBR-Co-Train 指的是使用最小贝叶斯风险系统融合的协同权重训练(如图 5 所示)的结果.为了更好地对比最小贝叶斯风险系统融合对协同训练的影响,我们引入了两个对比结果:

- MBR-ReRank-Co-Train.在这组对比实验中,我们使用最小贝叶斯风险的重排序方法将 BTG 和 Hiero 的翻译候选进行重排序,并将最好的前 4 个翻译结果作为对方的准参考译文;
- Top2Combin-Co-Train.我们将 BTG 解码器和 Hiero 解码器输出的结果中各自选取两个翻译候选凑成 4 个准参考翻译译文,并将此准参考翻译译文参与协同权重训练.

Table 5 Bleu scores for setting 1

表 5 实验设置 1 的结果

开发集 (NIST03)	协同训练的测试集(NIST06Web)			最终测试集(NIST08Web)	
	BTG	Hiero	MBR	BTG	Hiero
Baseline	25.96	26.35	26.67	18.47	19.82
Co-Train	26.64 (+**0.68**)	26.76 (+0.41)	26.81 (+0.14)	19.27 (+**0.80**)	20.06 (+0.24)
MBR-ReRank-Co-Train	26.72 (+**0.76**)	26.92 (+0.57)	26.75 (+0.08)	19.43 (+**0.96**)	20.18 (+0.31)
Top2Combin-Co-Train	26.57 (+**0.61**)	26.93 (+**0.58**)	27.09 (+**0.42**)	19.04 (+**0.57**)	20.21 (+0.39)
MBR-Co-Train	27.06 (+**1.10**)	27.47 (+**1.12**)	27.61 (+**0.94**)	19.32 (+**0.85**)	20.56 (+**0.74**)

Table 6 Bleu scores for setting 2

表 6 实验设置 2 的结果

开发集(NIST06Web)	协同训练的测试集(NIST03)			最终测试集(NIST05)	
	BTG	Hiero	MBR	BTG	Hiero
Baseline	34.30	35.11	35.17	33.23	33.90
Co-Train	35.10 (+**0.80**)	35.02 (–0.09)	35.46 (+0.29)	33.99 (+**0.76**)	34.47 (+0.57)
MBR-ReRank-Co-Train	35.07 (+**0.77**)	34.94 (–0.17)	35.28 (+0.11)	33.79 (+**0.56**)	33.91 (+0.01)
Top2Combin-Co-Train	34.91 (+**0.61**)	35.59 (+0.48)	35.58 (+0.41)	34.00 (+**0.57**)	34.49 (+0.59)
MBR-Co-Train	35.12 (+**0.82**)	35.77 (+**0.66**)	35.92 (+**0.75**)	33.94 (+**0.71**)	34.57 (+**0.67**)

从表 5 中实验设置 1 的结果可以看到,协同权重训练方法虽然可以显著[24]提高 BTG 解码器的性能(在 NIST06Web 上提高了 0.68,在 NIST08Web 上提高了 0.80),但并不能显著提高 Hiero 解码器的性能(在 NIST06Web 上提高了 0.41,在 NIST08Web 上提高了 0.24).我们分别将 Baseline 和 Co-Train 的两个解码器的翻译结果进行了 MBR 系统融合,系统融合的结果显示在 MBR 栏里.由于 Hiero 解码器的性能提升有限,所以在 NIST06Web 数据集上 MBR 系统融合的性能提升也非常的小(只有 0.14).从表 6 中实验设置 2 的实验结果我们也可以得出类似的结论.分析其原因,我们认为,准参考译文的质量是影响协同训练效果的关键因素,Hiero 解码器在各个数据集上均取得了比 BTG 解码器更好的性能,所以当使用 Hiero 解码器的译文作为 BTG 解码器的参考译文时,可以显著提高 BTG 解码器的翻译质量.然而,当使用两者相比性能略差的 BTG 解码器的译文来作为 Hiero 解码器的准参考译文时,对 Hiero 解码器的性能提升非常有限(最好的结果是在表 6 的 NIST05 数据集上有 0.57 个百分点的性能提升),甚至翻译效果会略微变差(在 NIST03 数据集上下降了 0.09 个百分点). MBR-ReRank-Co-Train 的效果相比 Co-Train 的效果略微好一些,但没有显著的差别.同 Co-Train 一样,在 MBR-ReRank-Co-Train 中,使用 Hiero 解码器的译文作为 BTG 解码器的参考译文时,可以显著提高 BTG 解码器的翻译质量,反之则不然.Top2Combin-Co-Train 在 BTG 的性能上虽不能像 Co-Train 那样好,然而各个数据上的表现却比较平均.我们发现,最小贝叶斯风险系统融合的协同权重训练方法不但可以显著[24]提高 BTG 解码器的性能(最好结果是在表 5 的 NIST06Web 数据集上有 1.1 个百分点的提升,最差是在表 6 的 NIST05 数据集上有 0.71 个百分点的提升),而且可以显著提高 Hiero 解码器的性能(最好和最差分别是表 5 的 NIST06Web 上的 1.12 个百分点和表 6 的 NIST03 上的 0.66 个百分点).

最小贝叶斯风险系统融合的协同权重训练相比原始的协同权重训练在各个数据集上均有更为显著的性能提高.最小贝叶斯风险系统融合的协同权重训练方法之所以能够提高 BTG 和 Hiero 解码器的性能,我们认为,主要原因是最小贝叶斯风险系统融合的方法可以从 BTG 和 Hiero 两个系统的输出结果中选择合适的翻译候选作为准参考译文,高质量的准参考译文能够在每轮的参数训练中起到积极的作用.另外,我们注意到,参与协同训练的测试集(NIST06Web 和 NIST03)上的性能提升相比最终测试集(NIST08Web 和 NIST05)上的性能提升要大一些,这也是合乎逻辑的,因为尽管参与协同训练的测试集和最终测试集属同一领域,但是其数据仍然存在着或大或小的差异.而由于前者参与了协同训练,故而所得到的特征权重或多或少要倾向于前者.

为了能更详细地观察协同训练的训练过程,我们在图 6 中显示了实验设置 1 实验中 10 遍协同训练(图 4 算法)的 BLEU 值的变化.其中,BTG(06Web)/Hiero(06Web)指的是 BTG/Hiero 解码器在 NIST06Web 年数据集上的 BLEU 值变化,BTG(08Web)/Hiero(08Web)指的是 BTG/Hiero 解码器在 NIST08Web 年数据集上的 BLEU 值变化.从图 6 可以看到,当我们没有采用最小贝叶斯风险系统融合来选择准参考译文时,协同训练方法的训练过程不稳定,原因可能如前所述,跟选用了质量较差的准参考译文有关.图 7 是实验设置 2 实验中 10 遍协同训练(图 4 算法)的 BLEU 值的变化,变化曲线同样不稳定.

我们同样在图 8 和图 9 中显示了实验设置 1 和实验设置 2 的最小贝叶斯风险系统融合的协同训练方法(图 5)的 BLEU 值变化.由于最小贝叶斯风险系统融合方法可以从参与协同训练的解码器结果中选择合适的译文作为准参考译文,从而能够避免质量较差或者不适合的译文(与解码器搜索空间中最好译文候选差别比较大)作为准参考译文时,最小错误率训练效果不稳定的现象.从结果中可以看到,最小贝叶斯风险系统融合的引入不但能够提高协同训练方法的效果,而且可以使得整个训练曲线比较平稳(尽管也有部分抖动).从图 8 和图 9 中看到,

最小贝叶斯风险系统融合方法本身并没有带来太大的性能提升,其性能与 Hiero 系统的输出结果相差不大.采用最小贝叶斯风险系统融合的协同训练方法在第 5 次协同训练之后开始收敛,其后性能提升有限.

我们在图 10 中分别显示了 10 次迭代过程中最小贝叶斯系统融合的结果中来自 BTG 和 Hiero 解码器的句子的分布情况(其中,左 Y 轴表示最小贝叶斯风险的系统融合的结果的句子中来自 BTG 解码器的句子所占的比例,右 Y 轴表示来自 Hiero 解码器所占的比例).随着迭代的进行,BTG 解码器所占的比重越来越大(尽管并不是平稳的增长),来自 Hiero 解码器的句子的比例越来越小(在两个实验设置上结果均如此).究其原因,我们认为,由于来自 Hiero 的准参考译文能够提升 BTG 解码器的翻译质量,而翻译的比较好的结果又更容易被最小贝叶斯风险的系统融合方法选中作为系统融合的结果,故而 BTG 解码器的句子比会呈现上升趋势.

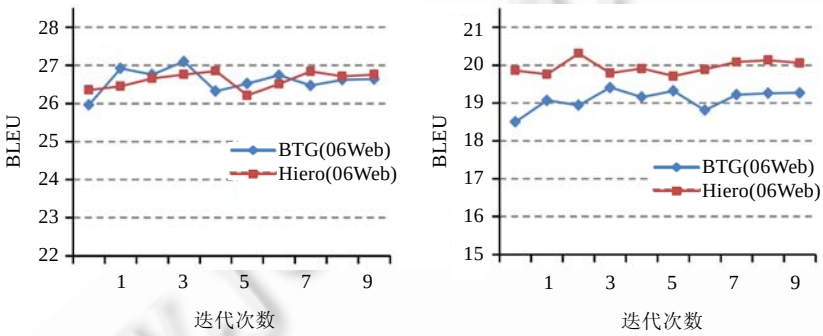


Fig.6 BLEU scores of co-training for setting 1

图 6 实验设置 1 上协同训练的 BLEU 值变化

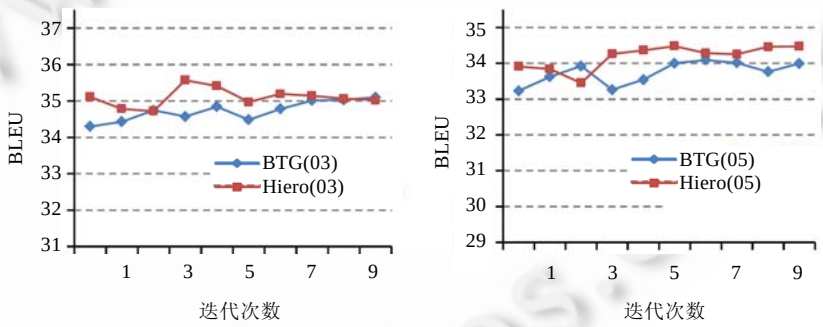


Fig.7 BLEU scores of co-training for setting 2

图 7 实验设置 2 上协同训练的 BLEU 值变化

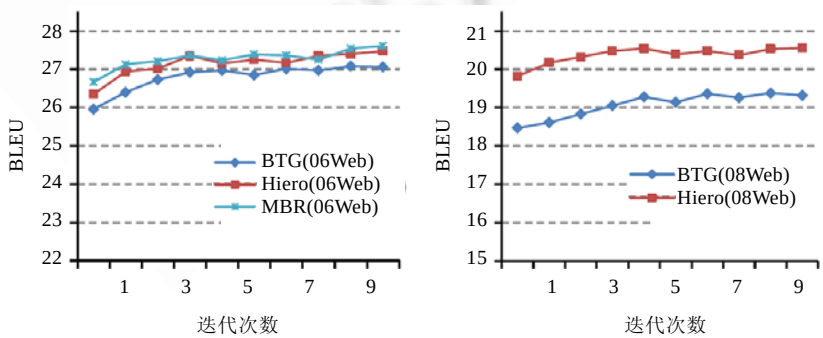


Fig.8 BLEU scores of co-training with MBR combination for setting 1

图 8 实验设置 1 上最小贝叶斯风险系统融合的协同权重训练的 BLEU 值变化

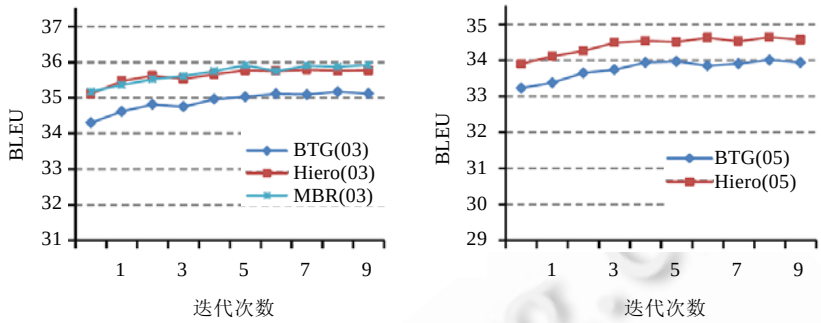


Fig.9 BLEU scores of co-training with MBR combination for setting 2

图 9 实验设置 2 上最小贝叶斯风险系统融合的协同权重训练的 BLEU 值变化

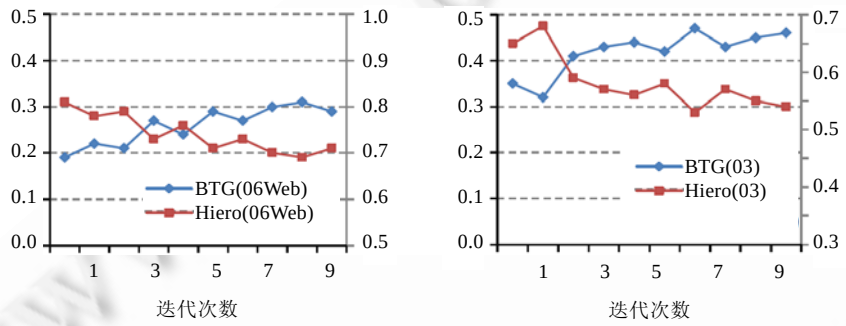


Fig.10 Ratios of sentences from BTG and Hiero in the MBR combination results

图 10 每次迭代后最小贝叶斯风险系统融合的结果中所含 BTG 和 Hiero 的翻译结果的句子比

4.2 协同训练同自学习方法的比较

使用协同训练来引入测试数据集参与特征权重的训练从而解决特征权重的领域适应问题之所以能够取得较好的结果,我们认为原因有两个:

1) 测试数据集的引入可以增加目标领域内样本的数量,从而使得特征权重的训练向目标领域倾斜;
2) 两个不同解码器的使用可以从不同的翻译结果中选择更为合适的译文作为准参考译文,从而可以改善添加到开发集中的目标领域内样本的质量.

为了增加目标领域内样本的数量,存在着另外一个与协同训练非常相似的训练方法,即自学习.在该节,我们将进行实验对协同训练和自学习进行比较.

与协同训练不同,自学习不需要多个解码器生成多组翻译结果.自学习的训练过程简单描述为:首先使用初始的开发数据集训练翻译模型,然后使用训练好的翻译模型对参与自学习的测试数据集进行解码,并从解码得到的翻译候选译文中选择合适的译文作为准参考译文添加到开发数据集当中,参与下一轮的特征权重的训练.为得到合适的准参考译文,我们同样采用了 MBR 重排序的方法(即第 3.3 节描述的 MBR 系统融合方法,因为只有一个系统,所以称为 MBR 重排序).

我们在设置 2 上分别使用 BTG 和 Hiero 进行了自学习训练,实验结果见表 7.其中,MBR-Self-Train 指的是自学习训练方法的结果,其最终 BLEU 得分指的是使用 MBR 重排序的结果.我们并将 MBR 重排序之后的结果重新进行了 MBR 系统融合,对应 MBR-Self-Train 一行中 MBR 一栏的结果.从实验结果中可以看到,因为自学习同样增加了开发数据集中领域内样本的数量,所以自学习的训练方法可以在目标领域内改善机器翻译结果的质量.然而,由于自学习仅仅只能从单一系统的翻译结果中选择准参考译文,故其性能相比协同训练仍有一定的差距.

Table 7 Comparison of self-training and co-training for setting 2

表 7 实验设置 2 上协同训练跟自学习的比较

开发集(NIST06Web)	协同训练的测试集(NIST03)			最终测试集(NIST05)	
	BTG	Hiero	MBR	BTG	Hiero
Baseline	34.30	35.11	35.17	33.23	33.90
MBR-Co-Train	35.12 (+0.82)	35.77 (+0.66)	35.92 (+0.75)	33.94 (+0.71)	34.57 (+0.67)
MBR-Self-Train	34.72 (+0.42)	35.72 (+0.61)	35.66 (+0.49)	33.75 (+0.52)	34.18 (+0.28)

4.3 参与协同训练的测试集的大小对协同训练的影响

为了分析参与协同训练的测试集的大小对协同训练过程的影响,我们分别从参与协同训练的测试集中随机的选取 $n/10(n=1,\ldots,10)$的数据作为额外的开发集,并用 NIST05 作为最终的测试集.其在 NIST05 上的翻译结果如图 11 所示,其中,Poly.(Hiero(05))是 Hiero(05)的 BLEU 值变化的多项式(阶数为 4)拟合趋势线,而 Poly.(BTG(05))是 BTG(05)的 BLEU 值变化的多项式拟合趋势线.

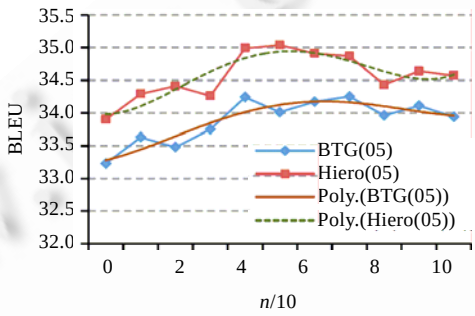


Fig.11 Effect of different test data size during co-training for setting 2

图 11 实验设置 2 上参与协同训练的测试集的大小对协同训练的影响

从图 11 中我们发现,最好的性能基本上在 4/10~7/10 之间,与原始开发数据集的规模相当.最好的 BLEU 比完全添加整个测试集在 BTG 和 Hiero 上分别高 0.47 和 0.30.从两条多项式拟合趋势线中发现,两条趋势线均呈现先升后降的趋势,也就是说,并不是参与协同训练的目标领域内的测试样本越多越好.这是因为,准参考翻译并不是完全正确的参考翻译,也就是说,准参考译文仍然带有错误.如果太多这样的参考译文被添加到开发数据集当中,就会影响特征权重的训练过程.

5 结 论

本文提出了使用协同训练来解决统计机器翻译中特征权重的领域自适应问题,即通过使用不同解码器的译文作为准参考译文,将测试集添加到开发集当中,使其参与特征权重的训练,从而使特征权重向测试集的领域倾斜,达到领域自适应的目的.本文进一步引入了最小贝叶斯风险系统融合的策略来选择测试集的准参考译文,该方法可以将参与协同训练的解码器的候选译文进行句子级的系统融合,选择合适的译文候选作为准参考译文,从而提高了准参考译文的质量,并进而提高协同训练的效果.实验结果表明,协同训练方法可以用来解决统计机器翻译训练过程中的特征权重领域自适应问题,并显著提高目标领域内机器翻译结果的质量.与此同时,基于最小贝叶斯风险系统融合的引入,不但可以进一步提高目标领域内机器翻译的质量,还可以使得训练过程更加稳定.

总结,本文的主要贡献点有:

1) 首次使用了协同训练的方法来解决统计机器翻译中特征权重的领域自适应问题,并进行了详细的实验和分析;
2) 引入了最小贝叶斯风险的系统融合方法来选择合适的译文候选,使得训练过程更加平稳.

通过实验我们发现,准参考译文的选择是影响协同训练性能的一个重要因素.在未来的研究中,我们将考虑使用更好的译文候选作为准参考译文,比如使用更加复杂和有效的词汇级系统融合方法.本文并没有对测试集中的句子进行详细分析,而是将其全部放入开发集中进行协同训练,这也许并不是最合适的方法.在未来的研究中,我们将采取策略选择一部分更适合的测试数据参与协同训练,以期望得到更好的训练效果.

References:

[1] Zhao TJ, *et al*. The Principle of Machine Translation. Harbin: Harbin Institute of Technology Press, 2001 (in Chinese).

[2] Brown P, Pietra S, Peitra V, Mercer R. The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics, 1993,19(2):263−311.

[3] Och FJ, Ney H. Discriminative training and maximum entropy models for statistical machine translation. In: Proc. of the Association for Computational Linguistics (ACL). 2002. 295−302. [doi: 10.3115/1073083.1073133]

[4] Koehn P, Schroeder J. Experiments in domain adaptation for statistical machine translation. In: Proc. of the 2nd Workshop on Statistical Machine Translation. 2007. 224−227.

[5] Lü YJ, Huang J, Liu Q. Improving statistical machine translation performance by training data selection and optimization. In: Proc. of the Empirical Methods in Natural Language Processing (EMNLP). 2007. 343−350.

[6] Sanchis-Trilles G, Cettolo M. Online language model adaptation via *n*-gram mixtures for statistical machine translation. In: Proc. of the European Association for Machine Translation (EAMT). 2010.

[7] Wu H, Wang H, Zong C. Domain adaptation for statistical machine translation with domain dictionary and monolingual corpora. In: Proc. of the Int'l Conf. on Computational Linguistics (COLING). 2008. 993−1000.

[8] Cao J, Lü YJ, Su JS, Liu Q. SMT domain adaptation based on monolingual context information. Journal of Chinese Information Processing, 2010,24(6):50−56 (in Chinese with English abstract).

[9] Ueffing N, Haffari G, Sarkar A. Transductive learning for statistical machine translation. In: Proc. of the 2nd Workshop on Statistical Machine Translation. 2007. 25−32.

[10] Li M, Zhao Y, Zhang D, Zhou M. Adaptive development data selection for log-linear model in statistical machine translation. In: Proc. of the Int'l Conf. on Computational Linguistics (COLING). 2010. 662−670.

[11] Wu D. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. Computational Linguistics, 1997,23(3): 377−403.

[12] Xiong D, Liu Q, Lin S. Maximum entropy based phrase reordering model for statistical machine translation. In: Proc. of the Association for Computational Linguistics (ACL). 2006. 521−528. [doi: 10.3115/1220175.1220241]

[13] Chiang W. Hierarchical phrase-based translation. Computational Linguistics, 2007,33(2):201−228. [doi: 10.1162/coli.2007.33.2.201]

[14] Och FJ, Ney H. A systematic comparison of various statistical alignment models. Computational Linguistics, 2003,29(1):19−51. [doi: 10.1162/089120103321337421]

[15] Och FJ. Minimum error rate training in statistical machine translation. In: Proc. of the Association for Computational Linguistics (ACL). 2003. 160−167. [doi: 10.3115/1075096.1075117]

[16] Blum A, Mitchell T. Combining labeled and unlabeled data with co-training. In: Proc. of the Annual Conf. on Learning Theory (COLT). 1998. 92−100. [doi: 10.1145/279943.279962]

[17] Pierce D, Cardie C. Limitations of co-training for natural language learning from large data sets. In: Proc. of the Empirical Methods in Natural Language Processing (EMNLP). 2001. 1−9.

[18] Sarkar A. Applying co-training methods to statistical parsing. In: Proc. of the North American Chapter of the Association for Computational Linguistics (NAACL). 2001. 1−8. [doi: 10.3115/1073336.1073359]

[19] Hwa R, Osborne M, Anoop S, Mark S. Corrected co-training for statistical parsers. In: Proc. of the ICML 2003 Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining. 2003.

[20] Wang W, Zhou Z. Analyzing co-training style algorithms. In: Proc. of the European Conf. on Machine Learning (ECML). 2007. 454−465. [doi: 10.1007/978-3-540-74958-5_42]

[21] Liang P, Alexandre B, Dan K, Ben T. An end-to-end discriminative approach to machine translation. In: Proc. of the Int'l Conf. on Computational Linguistics (COLING) and the Association for Computational Linguistics (ACL). Sydney, 2006. 761−768. [doi: 10.3115/1220175.1220271]

[22] Watanabe T, Suzuki J, Tsukada H, Isozaki H. Online large-margin training for statistical machine translation. In: Proc. of the Empirical Methods in Natural Language Processing (EMNLP) and the Special Interest Group on Natural Language Learning of the Association for Computational Linguistics (CoNLL). 2007. 764−773.

[23] Rosti AV, Ayan NF, Xiang B, Matsoukas S, Schwartz R, Dorr BJ. Combining outputs from multiple machine translation systems. In: Proc. of the North American Chapter of the Association for Computational Linguistics (NAACL) and Human Language Technologies (HLT). 2007. 228−235.

[24] Koehn P. Statistical significance tests for machine translation evaluation. In: Proc. of the Empirical Methods in Natural Language Processing (EMNLP). 2004. 388−395.

附中文参考文献:

[1] 赵铁军,等.机器翻译原理.哈尔滨:哈尔滨工业大学出版社,2001.

[8] 曹杰,吕亚娟,苏劲松,刘群.利用上下文信息的统计机器翻译领域自适应.中文信息学报,2010,24(6):50−56.

刘树杰(1982－),男,山东潍坊人,博士生,主要研究领域为统计机器翻译,词汇对齐,机器学习.

李沐(1972－),男,博士,研究员,主要研究领域为自然语言处理.

李志灏(1973－),男,博士,副研究员,主要研究领域为自然语言处理.

周明(1964－),男,研究员,教授,博士生导师,主要研究领域为自然语言处理,计算语言学,人工智能.