基于 n-gram 相似度的开发集选择

郑达奇　涂兆鹏　姜文斌　米海涛　刘洋　刘群
中国科学院计算技术研究所　智能信息重点实验室　北京　100190
E-mail：{zhengdaqi, tuzhaopeng, jiangwenbin, htmi, yliu, liuqun}@ict.ac.cn

摘 要：在 Och 最小错误率训练和对数线性模型的机器翻译框架下，本文提出了一种选择开发集的方法，通过分析开发集句子与测试集在 n-gram 上的相似度，选出和测试集相似度较高的一部分句子组成新的开发集。相较于原开发集，在此新选出的开发集上进行调参，能提高测试集的 BLEU 值。

关键字：自然语言处理、机器翻译、开发集选择

Development Set Selection Method Based On N-Gram Similarity

Daqi Zheng, Zhaopeng Tu, Wenbin Jiang, Haitao Mi, Yang Liu, Qun Liu
Key Laboratory of Intelligent Information Processing, Institute of Computing Technology,
Chinese Academy of Sciences, Beijing 100190, China
E-mail：{zhengdaqi, tuzhaopeng, jiangwenbin, htmi, yliu, liuqun}@ict.ac.cn

Abstract: *Based on Och's MERT (Minimal Error Rate Training) and Log-Linear Model, we proposed a novel development set selection method. According to analyzing the similarity of n-gram between sentences in development set and test set, we choose the sentences of closer similarity as a new development set. Compared to the original development set, we can get better results on test set by tuning the weights on the new one.*

Keywords: *natural language processing, machine translation, development set selection*

1　引言

在最小错误率训练和对数线性模型的机器翻译基本框架下[Och et al, 2002, 2003]，我们假设，开发集和测试集是都是从整个句子集合这个总体抽出来的样本，这个总体服从某种分布。因此开发集和测试集是独立同分布的。于是在开发集上调参得到的结果可以直接用于解码测试集，并且应该能得到最好结果。但是在实际中，我们发现，在开发集上取得最好结果的参数，并不一定能在测试集上取得最好结果。我们猜想，这可能是因为开发集句子数量占句子总体的比例很小，所以把开发集调出来的最好参数当成测试集的最好参数，偏差发生的可能性较大。

由此我们想到，是不是能选出开发集中和测试集比较相似的一部分作为新开发集，这样使得新开发集在统计意义上靠近测试集，这样也许能得到更适合解码测试集的参数。

在上述想法指导下，本文提出了一种基于 n-gram 相似度的开发集选择方法。通过计算开发集中句子和测试集的在 n-gram 意义下的相似度，从开发集中选出和测试集相似度较高的一部分句子组成新的开发集。用在新开发集上训练得到的最优参数来解码测试集。从而提高了测试集上的解码结果。

本文余下部分安排如下：第 2 部分介绍其他人的相关工作，第 3 部分介绍我们实现的

基于 n-gram 相似度的开发集选择方法。第 4 部分给出了实验所用数据，以及结果和分析。最后给出了结论。

2 相关工作

相关工作按照领域可以分为机器翻译和机器学习两大类。因为之前没有人在机器翻译领域发表过开发集选择的相关工作，所以下面主要提到的是训练集选择的一些工作。

机器翻译领域之前的语料赋权的工作，又可以被分为 3 类：数据选择，数据赋权，和翻译模型自适应。前两者有可能提高对齐准确度，并防止领域无关的短语对被抽取出来。而翻译模型自适应则提高那些领域相关短语对的权重或者是引入那些源语言端没有出现的对应翻译[Spyros Matsoukas et al, 2009]。

微软研究院在 2006 年 NIST 评测中所使用的方法是：先做在平行语料上做 GIZA++对齐，找出测试集中所有基于字的 n-gram；在对齐语料中每次取一个对齐双语句对，如果有测试集中出现过得 n-gram，就提取此句对；如果测试集中每个 n-gram 都在对齐语料的不少于 k 个句对中出现，则停止。用提取出来的这部分句对做规则抽取。这种方法减小了训练数据规模，从而加快了整个流程。而且他们希望这样选出来的数据与测试集更相关。该方法提高了约 0.5 个 BLEU 值[Xiaodong He et al, 2006]。李志飞等在 Joshua 上使用了和微软工作基本相同的方法，只是确定了 n-gram 是从 1 到 10，k 取 20。他在报告没有提及用该方法提升的 BLEU 值[Zhifei Li et al. 2009]。

黄瑾等人提出了一种基于信息检索模型的训练数据选择与优化方法，通过选择现有训练数据资源中与待翻译文本相似的句子组成训练子集，可在不增加计算资源的情况下获得与使用全部数据相当甚至更优的机器翻译结果。通过将选择出的数据子集加入原始训练数据中优化训练数据的分布可进一步提高机器翻译的质量[黄瑾等，2008]。

Yasuda 等采用的方法舍弃了平行语料中那些虽然翻译正确但是对提高测试领域翻译效果没有帮助的句对[Yasuda et al. 2008]。Mandal 等使用主动学习选择那些适合人来翻译的数据[Mandal et al. 2008]。Hildebrand 等用了一些信息检索方面的方法来选择类似的平行语料[Hildebrand et al. 2005]。

吕雅娟等使用的方法是在用 GIZA++做对齐时提高相关部分语料的权重，这也可以通过重复这部分语料来达到相似的效果。她们还用插值的方式使得原来在全部语料上训练出来的模型向领域相关的方向偏移[Lu et al. 2007]。

Koehn 等为了解决特定领域相关语料总是较领域无关语料小很多的问题，在领域相关和领域无关上分别训练翻译模型，再用 Och 的最小错误率训练在一个指定的开发集上调参把翻译出来的短语分数融合在一起[Koehn et al. 2007][Och, 2003]。

Matsoukas 等用判别式方法在一个指定的开发集上调参，从而给平行语料中的句子赋不同权重。和前面一些方法不同点就在，不舍弃那些无关的语料，只是赋以较低的权重[Matsoukas et al. 2009]。

在介绍机器学习领域的相关工作之前，值得说明的是：该领域中提及的训练集，更像机器翻译领域的开发集而不是训练集。因为机器翻译中的训练集是用来获取翻译能力，开发集则用来调整模型参数的，机器学习中训练集则同时具有两者的功能。

下列机器学习领域中的方法和上面提到的方法在思路上很相似，都是通过选择训练样本中认为有效的一部分来做训练集。这样最显著的好处是可以减小训练过程中所需的样本数，从而缩短训练时间。

MacKay 等认为，如果在学习过程中选择那些特征突出的数据点来做训练，将会更有效，因为这些点含有更多的信息。但这种做法的前提是假设空间是正确的。[MacKay et al, 1992]

Zhang 等认为在遗传算法运行过程中选择训练集中合适的部分数据，而不是反复使用

全部的训练数据，可以显著缩短进化所需时间而不减低泛化精度。[BT Zhang, DY Cho, 1998]

Hasenjäger 等认为虽然数据选择问题在成功的机器学习过程中不如其它一些因素重要，但是，一般情况下随机的选择数据并不是一种有效的做法，因为没有利用到已有的信息。在主动学习中，如果学习机可以根据已知的信息来选择那些其认为含有最多信息量的训练数据，然后在所选择的数据上训练，虽然选择数据的计算代价很高，但是可以有效的减少必须的训练样本。[M Hasenjäger et al. 2000]

本文中所用的开发集选择方法借鉴了机器学习领域数据选择的思想，通过选择开发集而不是训练集，能够比上述机器翻译领域中的方法更直接处理翻译模型领域倾向问题。而且有效的避免了在制定的开发集上调参可能导致的模型过拟合问题，尤其当句子特征到权重的映射非常复杂并基于大量参数时，或者是开发集和测试集差异较大时。

3 基于 n-gram 相似度的开发集选择方法

3.1 基本思想

在对数线性模型和最小错误率训练的机器翻译框架下[Och et al, 2002, 2003]，用开发集 dev 调节模型参数λ，并将得到的 dev 上的最优参数λ_{dev}=mert(dev)（在 dev 上取得最好结果 BLEU 值最高的一组参数）用于解码测试集 tst。其基本假设是所有源语言句子 f 的集合 S 这个总体服从某种分布 D(f, λ)，dev 和 tst 都是从 S 的样本，并且都服从该分布 D。或者说：dev 和 tst 是独立同分布的。这个做法的本质类似解决参数估计中的点估计问题中的矩估计法，而且只用到了类似一阶矩估计法（暂时没有二阶矩计算方法），只不过这里的估计量是λ=mert(X)，用 dev 得到的估计值是λ_{dev}。此方法用λ_{dev}来估计 S 上的最优参数λ_S，而且认为λ_S就是 tst 的最优参数λ_{tst}。因为现在一般是单句解码，没有用到上下文信息，实际上，可以进一步得到：对所有的句子，可能性最大的解码的最优参数λ_{best}应该都是相同的。

但是在实际中，我们发现，在 dev 上取得最好结果的参数λ_{dev}，并不一定能在 tst 上取得最好结果。我们猜想，这可能是因为开发集句子数量占句子总体的比例很小，并不能准确描述该语言的分布规律，所以用 dev 上的估计值来估计 S 的参数偏差发生的可能性较大。

因为已有的开发集相对测试集来说较大，所以在机器翻译评测中，有一种方法是将已有的开发集 dev 按年代划分成几个子开发集 dev_1~dev_n和一个临时测试集 dev_tst，然后选择能在 dev_tst 上取得最好结果的参数λ_{dev_tst}用于正式的测试集解码，认为用该组参数解码的模型具有较强的泛化能力。也就是说，认为对应的 dev_x比其他子开发集更接近总体句子分布。另一种方法是用交叉测试来检验参数的一般化程度，具体做法大同小异。上述两种做法虽然假设开发集和总体的分布有偏差，但是仍然假设测试集和总体有相同的分布。所以它们只是考量λ_{dev}对λ_S的逼近程度。

如果假设测试集和总体的分布也有偏差，上述做法便有些不合适了。我们猜想，能否构造一个和测试集分布偏差较小的开发集，即：使得λ_{dev}和λ_{tst}的偏差尽可能缩小。简而言之，就是开发集和测试集都可能偏离总体分布，不如使开发集直接逼近测试集分布。

3.3 具体实现

在本节中，我们提出了一种很粗糙的度量句子和句子集合相似度的方法，并将其用于开发集选择。

图 1 描述了加入了开发集选择的整个翻译流程，首先用测试集做标准进行开发集选择，待用新开发集进行最小错误率训练调参得到最优参数后，用该组参数解码测试集得到翻译结果。

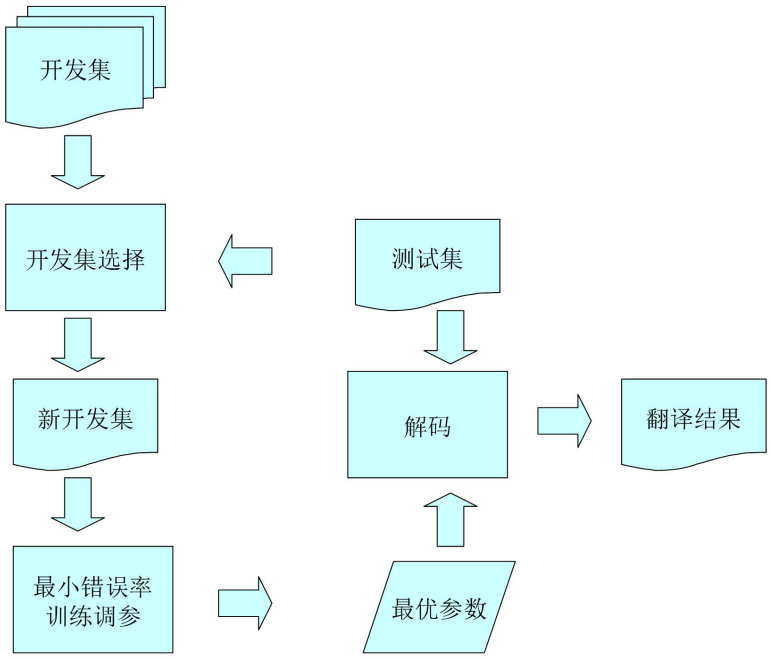

图 1. 开发集选择翻译流程图

3.3.1 测试集预处理

提取测试集上的 n-gram，得到一个集合 W，本文中提取的是 1 到 10 gram。给 W 中的每个 n-gram 赋上相应的分数 score(n-gram)，本文中令

$$score(n-gram) = length(n-gram) * count(n-gram)$$

$count(n-gram)$是$n-gram$在整个测试集中出现的次数

$length(n-gram)$是以$token$计数的$n-gram$的长度

这是一个经验公式。

3.3.2 开发集句子相似度评分

对于开发集中的每个句子 sent，提取 sent 中的 n-gram，也得到一个集合 W'，和 2.1 中对应，这里也是取 1 到 10 gram。W 和 W'的交集为 W*。

sent 的分数为：

$$sent-score(sent) = \frac{\sum_{n-gram \in W*} \{score(n-gram) * count(n-gram)\}}{length(sent)}$$

$score(n-gram)$的定义见上节

$count(n-gram)$此处是$n-gram$在$sent$中出现的次数

$length(sent)$此处是以$token$计数的sent的长度

这也是一个经验公式。

因为 1-gram 在评分中所占权重已经很低，我们便没有做使用停用词过滤的对比试验。

3.3.3 选择新开发集

选择开发集中相似度分数较高的前 x 句组成新开发集 dev_selected。我们希望，这样选

出来的 dev_selected 能比整个 dev 更接近于 tst 的分布，在 dev_selected 上训练出来的参数也能在 tst 上取得好结果。

4 实验及结果分析

我们使用的解码器是我们重实现的层次短语模型系统 Hiero[Chiang, 2005]。我们通过 MERT 来调节参数的权重，并以 BLEU4[Papineni et al., 2001]作为测试标准。

为了验证开发集选择的有效性，我们在 NIST 大规模数据和 IWSLT 小规模数据上分别做了实验：

4.1 NIST 实验

我们使用 LDC 双语对齐语料作为实验的训练集，它包含 5.6M 句对。实验使用的 5 元语言模型，是通过 SRI 工具从新华、GigaWord 以及双语语料中的英语端训练而来的。

在实验中，我们以 NIST05 作为测试集，我们从 NIST02~NIST08(NIST05 除外)中选择开发集。选择那些 sent-score 超过某个指定阈值的句子组成新开发集。

我们主要做了以下五个实验：

- baseline: 以 NIST02 作开发集；
- max-count1: 阈值为 1 时的开发集选择结果作为开发集；
- max-count2: 阈值为 2 时的开发集选择结果作为开发集；
- max-count3: 阈值为 3 时的开发集选择结果作为开发集；
- self-mert: 以 NIST05 本身作开发集，以求得其最好的结果。

表 1 显示了各集合中包含的句子数：

表 1 集合所含句子数

集合	句子数
baseline	878
max-count1	1788
max-count2	1025
max-count3	709
self-mert	1082

实验结果如下：

表 2 开发集和测试集上的 BLEU 值

	开发集结果	测试集结果
baseline	0.3449	0.3432
max-count1	0.3508	0.3385
max-count2	0.3607	0.3484
max-count3	0.3612	0.3398
self-mert	0.3515	0.3515

从表 2 中可以看到，用 NIST05 本身开发，MERT 调参最高只能达到 0.3515，只比 baseline(0.3432)高出 0.8 个点。如果开发集选择的实验结果能落在 baseline 和最高之间，就可以说明是有效果的。而 max-count2 实验结果比 baseline 高 0.5 个点，只比 self-mert 低 0.3 个点。可以认为是一个比较理想的结果。

同时，我们可以看到，max-count1 实验结果比 baseline 反而低 0.5 个点。我们认为，这是因为 max-count1 开发集选择阈值过低，使得新开发集中包含 1788 个句子，远大于 max-count2 开发集的句子数(1025)，多出来的都是与测试集相似度低的句子，这样就降低了新开发集整体和测试集的相似度，使得在新开发集上训练出来的参数更加偏离测试集的最优参数，导致测试集上结果偏低。

max-count3 上的结果也降低原因，应该是提高了阈值，使得选出来的句子数更少，从而更容易产生偏移，和测试集相差也可能越大。

这和直觉上的估计也十分相似。要在两方面原因中找一个对应峰值的平衡点，而 max-count2 应该就是出现在平衡点附近。

4.2 IWSLT 实验

我们使用 CT 双语对齐语料作为实验的训练集，它包含 134K 句对。实验使用的 5 元语言模型，是通过 SRI 工具从双语语料中的英语端训练而来的。

在实验中，我们以 IWSLT08 作为测试集。由于 IWSLT07 中只含 6 个参考译文，而 IWSLT03~IESLT05 中包含 16 个参考译文，所以我们只从 IWSLT03~IWSLT05 中选择开发集。

我们主要做了以下四个实验：

- baseline: 以 IWSLT07 作开发集；
- self-mert: 以 IWSLT08 本身作开发集，以求得其最好的结果。
- max-count(x): 阈值为 x 时的开发集选择结果作为开发集。

表 3 显示了各集合中包含的句子数：

表 3 集合所含句子数

集合	句子数	集合	句子数
IWSLT07	489	max-count5	113
IESLT08	507	max-count6	99
max-count1	413	max-count7	85
max-count2	249	max-count8	74
max-count3	191	max-count9	62
max-count4	143	max-count10	55

实验结果如下：

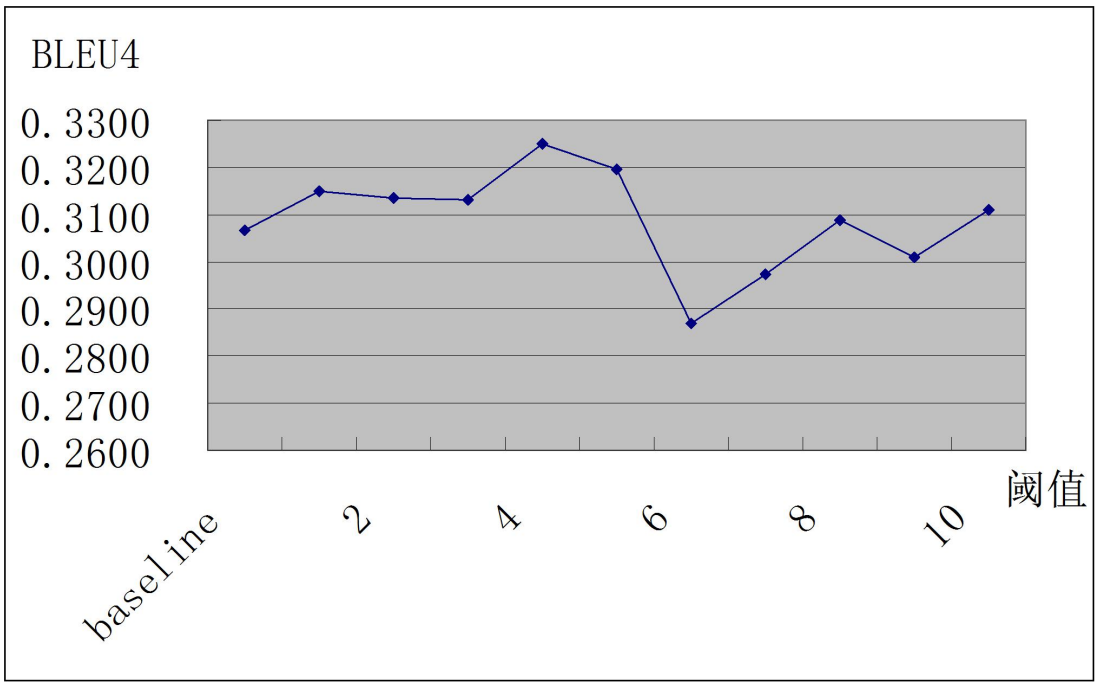

图 2 阈值不同的开发集选择在测试集上的 BLEU 值

表 4 开发集和测试集上的 BLEU 值

	开发集结果	测试集结果
baseline	0.2445	0.3065
self-mert	0.3342	0.3342
max-count1	0.4456	0.3149
max-count2	0.4631	0.3134
max-count3	0.4755	0.3130
max-count4	0.4800	0.3250
max-count5	0.4862	0.3197
max-count6	0.4891	0.2868
max-count7	0.4879	0.2975
max-count8	0.5083	0.3089
max-count9	0.5516	0.3011
max-count10	0.5590	0.3108

从图 2 中可以看到，开发集选择的实验结果在阈值为 4 的附近取到峰值(0.3250)。处于 baseline 和最高值(0.3065, 0.3342)这个区间靠近最高值的 1/3 处。和 NIST 上实验的结果结合

起来看，说明如果能准确找到峰值对应的阈值，那么这种开发集选择的方法还是比较有效的。至于结果曲线局部出现如此剧烈的变动，我们猜想，可能是由于相似度计算的方法不够准确，导致开发集选择的时候混入了一些垃圾数据干扰的原因。

5 结论

虽然度量相似度的方法非常粗糙，但是仍然能比 baseline 有提高，说明此方法可行。后续的工作可以有两个方向：

第一是提出更准确的计算相似度的方法，比如借鉴信息检索中这方面的成果，更准确的相似度也许会带来更大的 BLEU 值的提高。另一个更偏统计的思路是使得新开发集在概率分布的角度上逼近测试集，只是算法实现的复杂程度更高。不过鉴于在 NIST05 上用 MERT 也不能把结果调上去，可能反映出瓶颈更主要是在 MERT 调参上。

第二是对测试集中的每一个句子都做这样的开发集选择，考虑到这样做需要的计算资源太大，可以尝试在源语言句子上抽取特征，通过单句调参，建立特征到参数的映射关系。这样可以预调参数，解码的时候不需要重新训练。把开发集选择从另一个方向做到极致。

6 致谢

本文的研究工作得到以下项目的资助：国家自然科学基金重点项目（项目批准号 60736014）、863 重点项目课题（课题编号 2006AA010108）、国家自然科学基金项目（批准号 60873167）。感谢匿名评审的宝贵意见。感谢我女朋友的大力支持。

参考文献

[1] Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, PA, July.

[2] Franz Josef Och 2003. Minimum error rate training in statistical machine translation. In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, pages 160–167.

[3] Spyros Matsoukas and Antti-Veikko I. Rosti and Bing Zhang. 2009. Discriminative Corpus Weight Estimation for Machine Translation. Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pages 708–717. Singapore, 6-7 August 2009.

[4] Xiaodong He, Arul Menezes, Chris Quirk, Jianfeng Gao, Patrick Nguyen, Anthony Aue and Simon Corston-Oliver. 2006. Microsoft Research Chinese-English large track. In Proceedings of NIST 2006.

[5] Zhifei Li, Chris Callison-Burch, Chris Dyer, Juri Ganitkevitch, Sanjeev Khudanpur, Lane Schwartz, Wren Thornton, Jonathan Weese and Omar Zaidan. 2009. Joshua: An Open Source Toolkit for Parsing-based Machine Translation. In Proceedings of WMT 2009.

[6] 黄瑾，吕雅娟，刘群. 基于信息检索的统计翻译训练数据选择与优化. 中文信息学报，第 22 卷，第 2 期，第 40-46 页，2008 年 3 月

[7] Keiji Yasuda, Ruiqiang Zhang, Hirofumi Yamamoto, and Eiichiro Sumita. 2008. Method of selecting training data to build a compact and efficient translation model. In Proceedings of the Third International Joint Conference on Natural Language Processing, volume II, pages 655–660.

[8] Arindam Mandal, Dimitra Vergyri, Wen Wang, Jing Zheng, Andreas Stolcke, Gokhan Tur, Dilek Hakkani-Tüur, and Necip Fazil Ayan. 2008. Efficient data selection for machine translation. In Proceedings of the Second

IEEE/ACL Spoken Language Technology Workshop, pages 261–264.

[9] Almut Silja Hildebrand, Matthias Eck, Stephan Vogel, and Alex Waibel. 2005. Adaptation of the translation model for statistical machine translation based on information retrieval. In Proceedings of the 10th Annual Conference of European Association for Machine Translation, pages 133–142.

[10] Yajuan Lu, Jin Huang, and Qun Liu. 2007. Improving statistical machine translation performance by training data selection and optimization. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 343–350.

[11] Philipp Koehn and Josh Schroeder. 2007. Experiments in domain adaptation for statistical machine translation. In Proceedings of the Second Workshop on Statistical Machine Translation, pages 224–227.

[12] MacKay, D. J. C. 1992. Information-based objective functions for active data selection. Neural Computation, 4(4), 590–604.

[13] Zhang, B.-T. and Cho, D.-Y. 1998. Genetic programming with active data selection. In Newton, C., editor, Proceedings of the Second Asia-Pacific Conference on Simulated Evolution and Learning SEAL'98, volume 1.

[14] M Hasenjäger. 2000. Active Data Selection in Supervised and Unsupervised Learning. PhD thesis, University of Bielefeld.

[15] Chiang, David. 2005. A hierarchical phrase-based model for statistical machine translation. In Proceedings of the 43rd Annual Meeting of the ACL, pages 263–270, Ann Arbor, MI.

[16]K. Papineni, S. Roukos, and T. Ward. 2001. Bleu: a method for automatic evaluation of machine translation. IBM Research Report, RC22176