Simulating Discriminative Training for Linear Mixture Adaptation in Statistical Machine Translation

George Foster, Boxing Chen, and Roland Kuhn
National Research Council Canada
first.last@nrc-cnrc.gc.ca

Abstract

Linear mixture models are a simple and effective technique for performing domain adaptation of translation models in statistical MT. In this paper, we identify and correct two weaknesses of this method. First, we show that standard maximum-likelihood weights are biased toward large corpora, and that a straightforward preprocessing step that down-samples phrase tables can be used to counter this bias. Second, we show that features inspired by prototypical linear mixtures can be used to loosely simulate discriminative training for mixture models, with results that are almost certainly superior to true discriminative training. Taken together, these enhancements yield BLEU gains of approximately 1.5 over existing linear mixture techniques for translation model adaptation.

1 Introduction

SMT systems, like other statistical NLP systems, experience difficulties when the domain in which they must operate is different from the one on which they were trained. Domain adaptation techniques intended to address this problem have attracted significant research attention in recent years. One simple and popular approach is mixture adaptation, in which models trained on various sub-corpora are weighted according to their relevance for the current domain. Mixture adaptation is most effective when there are many heterogeneous sub-corpora, at least some of which are not too different from the test domain.[1] It typically requires only a small in-domain sample—a development or test set—in order to obtain weights, and has been shown to work well with all major SMT model types: language (Foster and Kuhn, 2007), translation (Koehn and Schroeder, 2007; Foster et al., 2010; Sennrich, 2012b), and reordering (Chen et al., 2013).

There are two main types of mixture model: linear mixtures, which combine weighted probabilities from component models additively; and log-linear mixtures, which combine multiplicatively. Linear mixtures frequently perform better (Foster et al., 2010), especially when there are a relatively large number of sub-corpora, and when the models derived from them are not smooth. This is likely due to the "veto power" that any component model exercises within a log-linear combination: it can suppress hypotheses by assigning them low probabilities. To avoid the complete suppression of in-domain hypotheses by weaker models, the only option is to effectively turn these models off by downweighting them severely, thereby discarding whatever useful information they might possess.

Although linear mixtures are attractive for adaptation, they have the potential disadvantage that it is difficult to tune mixture weights directly for an SMT error metric such as BLEU. This is because, in order to allow for decoder factoring, models must be mixed at the local level, ie over ngrams or phrase/rule pairs. Thus the linear mixtures oc-

[1] The assumption that train and test domains are fairly similar is shared by most current work on SMT domain adaptation, which focusses on modifying scores. Haddow and Koehn (2012) show that coverage problems dominate scoring problems when train and test are more distant.

Sima'an, K., Forcada, M.L., Grasmick, D., Depraetere, H., Way, A. (eds.)
Proceedings of the XIV Machine Translation Summit (Nice, September 2–6, 2013), p. 183–190.

cur *inside* the normal log probabilities that are assigned to these entities, making mixture weights inaccessible to standard tuning algorithms. Notice that log-linear mixtures do not suffer from this problem, since component probabilities can be incorporated as features directly into the standard SMT model.

The usual solution to the problem of setting linear mixture weights is to sidestep it by optimizing some other criterion, typically dev-set likelihood (Sennrich, 2012b), instead of BLEU. This achieves good empirical results, but leaves open the question of whether tuning linear weights directly for BLEU would work better, as one might naturally expect. This question—which to our knowledge has not yet been answered in the literature—is the one we address in this paper.

Modifying standard tuning algorithms to accommodate local linear mixtures presents a challenge that varies with the algorithm. In MERT (Och, 2003) for instance, one could replace exact line maximization within Powell's algorithm with a general-purpose line optimizer capable of handling local linear mixtures at the same time as the usual log-linear weights. For expected BLEU methods (Rosti et al., 2011), handling local weights would require computing gradient information for them. For MIRA (Chiang et al., 2008), the required modifications are somewhat less obvious.

Rather than tackling the problem directly by attempting to modify our tuning algorithm, we opted for a simpler indirect approach in which features are used to *simulate* a discriminatively-trained linear mixture, relying on the ability of MIRA to handle large feature sets. Our aim in doing so is not to achieve a close mathematical approximation, but rather to use features to capture the kinds of information we expect to be inherent in linear mixtures. We apply this approach to translation model (TM) adaptation and show that it outperforms standard linear mixtures trained for dev-set likelihood. We also give strong evidence that it would outperform linear mixtures trained directly to maximize BLEU. As an additional contribution, we identify a problem with standard likelihood estimation for mixtures that use unsmoothed component models from unbalanced corpora, and provide a simple solution that significantly improves performance.

2 Linear Mixture Adaptation for TMs

We now describe the adaptation methods used in this paper. We begin with standard maximum-likelihood mixtures, then describe the enhancement for dealing with unbalanced corpora, and finally present our method for simulating discriminatively-trained mixtures. All techniques are applied to TM adaptation; for brevity we describe them in terms of target-conditioned estimates only.

2.1 Maximum Likelihood Mixtures

Given a set of training sub-corpora, let $p_i(s|t)$ be the conditional probability for phrase pair s, t in a phrase table extracted from the ith sub-corpus. The resulting mixture model is:

$$p(s|t) = \sum_i w_i \, p_i(s|t). \tag{1}$$

To set the weights w, we extract a bag of phrase pairs from an in-domain development set using standard techniques. This yields a joint distribution $\tilde{p}(s,t)$, which we use to define a maximum likelihood objective:

$$\hat{w} = \operatorname*{argmax}_w \sum_{s,t} \tilde{p}(s,t) \log \sum_i w_i \, p_i(s|t).$$

This maximization can be performed in a straightforward and efficient way using the EM algorithm.

2.2 Correcting for Large-Corpus Bias

A naive implementation of the method in the previous section would extract a separate phrase table from each sub-corpus and assign $p_i(s|t) = 0$ to all pairs s, t not in the ith table.[2] This creates a bias in favour of large sub-corpora, which will tend to contain a greater number of phrase pairs from the development set than smaller sub-corpora. To see why, consider the expected count credited to the ith model for phrase pair s, t during the E-step of the EM algorithm:

$$\frac{w_i \, p_i(s|t)}{\sum_j w_j \, p_j(s|t)}.$$

Clearly, if the ith model is the only one that assigns non-0 probability to s, t, then its count increment for this pair will be 1, regardless of how small

[2] In cases where t also does not exist in the ith table, this makes the resulting distribution $p_i(*|t)$, and hence the overall mixture, non-normalized. This is not a concern per se, but rather a symptom of the problem we identify in this section.

a (non-0) probability it actually assigns. Since weights are set to normalized expected counts in the M-step, this implies that sub-corpora with higher dev-set coverage will tend to get assigned higher weights.

One might argue that weights tuned for BLEU could be expected to exhibit the same bias, in which case it would obviously not be harmful. However, note that the above analysis applies also to large *out-of-domain* sub-corpora. Even though models from such corpora may assign very small $p_i(s|t)$ to phrase pairs from the dev set, they will still get a large EM bonus for pairs that are covered by few or no other models. However, domain mismatch means that there will likely be other pairs s, t' for which $p_i(s|t') \gg p_i(s|t)$. At translation time, such pairs will be detrimental to BLEU score if w_i is high.

One way to solve this problem would be to jointly smooth models across sub-corpora, for instance using a hierarchical MAP approach (Aminzadeh et al., 2012). We experimented with simple add-epsilon smoothing, but found that a more effective approach was to sample sub-corpora in order to remove size discrepancies. More precisely, before running EM, we randomly select r_i phrase pairs from the ith phrase table according to $\tilde{p}(s,t)$, their joint probability in the dev set distribution,[3] and consider $p_i(s|t)$ to be 0 for all other pairs. We set $r_i = s_i \, n_{min}/n_i$, where s_i is the number of pairs in table i for which $\tilde{p}(s,t) > 0$, n_i is the total number of pairs in that table, and $n_{min} = \min_i \, n_i$. This approach gave slightly better results than subsampling corpora prior to phrase extraction, and was easier to implement with our existing mixture-model infrastructure. After obtaining weights with EM, the full sub-corpus phrase tables are mixed as usual.

2.3 Simulating Discriminative Training

Discriminative training applied to a mixture model (1) will yield a particular set of weights w and a resulting feature value:

$$g_w(s,t) = \log \sum_{i=1}^{m} w_i \, p_i(s|t).$$

Our aim is to find a set of phrase-pair features $f_j[p_1(s|t), \ldots, p_m(s|t)]$ that can be suitably weighted to approximate $g_w(s,t)$ for any arbitrary value of w. Formally, for any w, there should exist a set of log-linear weights λ such that:

$$g_w(s,t) \approx \sum_j \lambda_j \log f_j[p_1(s|t), \ldots, p_m(s|t)]. \quad (2)$$

Notice that this property does not guarantee that optimizing λ will yield a combination $\sum_j \lambda_j \log f_j[\cdots]$ that can be interpreted (even approximately) as $g_w(s,t)$ for some set of linear weights w. In other words, the space defined by weighted combinations of features $\log f_j[\cdots]$ might strictly contain the space of linear mixtures $g_w(s,t)$. This does not pose a problem, however, because our aim is not to carry out an exact simulation of discriminatively-trained linear mixtures but rather a loose simulation that captures the essential properties of these mixtures for adaptation. Furthermore, if an assignment to λ does result in a combination that is outside the space of linear mixtures, it will be because that assignment is better, in the sense of yielding a higher training BLEU score, than any linear mixture.

A set of functions that trivially satisfies (2) is one that includes a feature f_w for *every* set of weights w:

$$f_w[p_1(s|t), \ldots, p_m(s|t)] = \sum_{i=1}^{m} w_i \, p_i(s|t),$$

where equality in (2) is achieved by setting $\lambda_w = 1$ and $\lambda_{w'} = 0, \; \forall w' \neq w$. Although this solution is clearly impractical, it motivates our choice of features, which are intended to capture or approximate various prototypical weightings. Our hypothesis is that a precise setting of linear weights is less important than being close to an appropriate prototype.

We now turn to a description of our features, which are as follows:

- EM. Under the hypothesis that maximum-likelihood weights may sometimes be identical to maximum-BLEU weights, we define one feature to be the weighted combination $\sum_{i=1}^{m} \hat{w}_i \, p_i(s|t)$ returned by EM.[4]
- Avg. This feature captures the case when all component models are equally valuable: $\sum_{i=1}^{m} p_i(s|t)/m$

[3] Note that this discards pairs that weren't extracted from the dev set, for which $\tilde{p}(s,t) = 0$.

[4] We tried sharpened and flattened versions of these weights, $\hat{w}^{\alpha}$, but did not see gains by tuning α.

- Avgnz. To approximate mixtures that downweight models which frequently assign 0 probabilities, and weight other models uniformly, this feature takes the average over non-zero probabilities: $\sum_{i=1}^{m} p_i(s|t)/m_{nz}$, where $m_{nz} = \sum_{i=1}^{m} [p_i(s|t) \neq 0]$

- Support. This is a version of *Avgnz* in which non-zero probabilities are mapped to 1: m_{nz}/m.

- Max. Mixtures which contain fairly uniform weights can be roughly approximated by the component model that assigns highest probability to the current phrase pair (ie, the Viterbi approximation): $\max_{i=1}^{m} p_i(s|t)$.

- Onevsall(w). This feature captures mixtures in which the ith model is weighted by w and all other models are weighted uniformly: $w\, p_i(s|t) + \frac{1-w}{m-1} \sum_{j \neq i} p_j(s|t)$. We used one Onevsall feature for each component model, all parameterized by the same weight $w \in [0, 1]$.

In addition to the above dense features, we also experimented with sparse boolean features intended to capture the rank of a given model's probability assignment for the current phrase pair:

- SparseRank(c). Among all models for which $p(s|t) > c$, fire if the rank of $p_i(s|t)$ is in $[b, e]$. We defined one instance of this feature for each sub-model i and each rank range in $[1, 1]$, $[2, 3]$, $[4, 8]$, and $[9, \infty)$.

3 Experiments

We carried out experiments with a phrase-based SMT system on two different language pairs, using a heterogeneous mix of training sub-corpora.

3.1 Data

Data for all our experiments was made available as part of NIST Open MT 2012.[5] Our first setting uses data from the Chinese to English constrained track, comprising approximately 10M sentence pairs. We manually grouped 14 sub-corpora on the basis of genres and origins. Table 1 summarizes the statistics and genres of all training and test material. Our development set was taken from the NIST 2005 evaluation set, augmented with some web-genre material reserved from other NIST corpora.

corpus	# segs	# en tok	%	genre
fbis	250K	10.5M	3.7	nw
financial	90K	2.5M	0.9	fin
gale_bc	79K	1.3M	0.5	bc
gale_bn	75K	1.8M	0.6	bn ng
gale_nw	25K	696K	0.2	nw
gale_wl	24K	596K	0.2	wl
hkh	1.3M	39.5M	14.0	hans
hkl	400K	9.3M	3.3	law
hkn	702K	16.6M	5.9	nw
isi	558K	18.0M	6.4	nw
lex&ne	1.3M	2.0M	0.7	lex
others_nw	146K	5.2M	1.8	nw
sinorama	282K	10.0M	3.5	nw
un	5.0M	164M	58.2	un
TOTAL	10.1M	283M	100.0	(all)
devtest				
dev	1,506	161K		nw wl
NIST06	1,664	189K		nw bn ng
NIST08	1,357	164K		nw wl

Table 1: NIST Chinese-English data. In the *genre* column: nw=newswire, fin=financial, bc=broadcast conversation, bn=broadcast news, ng=newsgroup, wl=weblog, hans=Hansard, law=legal, lex=lexica, un=United Nations proceedings.

The second setting uses NIST 2012 Arabic to English data, but excluding the UN data. There are about 1.5M English running words in these training data. We manually grouped the training data into 7 groups according to genre and origin. Table 2 summarizes the statistics and genres of all the training corpora and the development and test sets. We use the evaluation sets from NIST 2006, 2008, and 2009 as our development set and two test sets, respectively.

3.2 System

Experiments were carried out with a phrase-based system similar to Moses (Koehn et al., 2007). The corpus was word-aligned using IBM2, HMM, and IBM4 models, and the phrase table was the union of phrase pairs extracted from these separate alignments, with a length limit of 7. Con-

[5] http://www.nist.gov/itl/iad/mig/openmt12.cfm

corpus	# segs	# en toks	%	genre
gale_bc	57K	1.6M	3.3	bc
gale_bn	45K	1.2M	2.5	bn
gale_ng	21K	491K	1.0	ng
gale_nw	17K	659K	1.4	nw
gale_wl	24K	590K	1.2	wl
isi	1.1M	34.7M	72.6	nw
other_nw	224K	8.7M	18.2	nw
TOTAL	1.5M	47.8M	100.0	(all)
devtest				
NIST06	1,664	202K		nw wl
NIST08	1,360	205K		nw wl
NIST09	1,313	187K		nw wl

Table 2: NIST Arabic-English data. The *genre* labels are the same as for Chinese.

method	Arabic	Chinese
baseline	46.79	31.72
mixtm-ml	47.39	32.73
mixtm-uni	48.02	33.29
mixtm-samp	**48.13**	**34.01**

Table 3: Mixture TM Adaptation.

ditional phrase pair estimates in both directions were obtained using Kneser-Ney smoothing (Chen et al., 2011), and were weighted (each directional estimate separately) for adaptation. We also used lexical estimates in both directions which were not weighted (no sub-corpus-specific values were available). Additional features included a hierarchical lexical reordering model (Galley and Manning, 2008) (6 features), standard distortion and word penalties (2 features), a 4-gram LM trained on the target side of the parallel data, and a 6-gram English *Gigaword* LM (14 features total). The decoder used a distortion limit of 7, and at most 30 translations for each source phrase. The system was tuned with batch lattice MIRA (Cherry and Foster, 2012).

3.3 Results

Our first experiment compares a non-adapted baseline, trained on all available corpora, with standard maximum likelihood (ML) mixture TM adaptation as described in section 2.1 and the sampled variant described in section 2.2. The results are presented in table 3, in the form of BLEU scores averaged over both test sets for each language pair. ML mixture adaptation (mixtm-ml) yields significant gains of 0.6 and 0.9 for Arabic and Chinese over the unadapted model. However, in our setting, this underperforms assigning uniform weights to all component models (mixtm-uni). Equal-size sampling of the phrase tables (mixtm-samp) performs best, improving over the plain ML method by 0.7 and 1.3 BLEU.

corpus	PT size	weights	
	(# pairs)	mixtm-ml	mixtm-samp
financial	4.4M	0.033	0.142
gale_bc	3.6M	0.022	0.091
gale_bn	4.6M	0.059	0.126
gale_nw	1.8M	0.082	0.279
gale_wl	1.4M	0.017	0.176
hkh	62.2M	0.120	0.008
hkl	8.2M	0.002	0.024
hkn	26.6M	0.035	0.021
isi	26.1M	0.045	0.011
ne_lex	1.9M	0.003	0.030
others_nw	7.8M	0.050	0.042
fbis	19.6M	0.142	0.026
sinorama	17.0M	0.053	0.020
unv2	360.1M	0.341	0.005

Table 4: Comparison of weights assigned to Chinese sub-corpora by standard and sampled ML mixture models.

Table 4 shows the weights assigned to the Chinese component models by these two strategies, and gives some insight into their behaviour. The bias of mixtm-ml in favour of large corpora is very evident, particularly in its assignment of by far the largest weight to the highly out-of-domain UN corpus. This trend is reversed by mixtm-samp, which assigns a suitably low weight to UN, and the highest weight to the small but very relevant gale_nw corpus.

We now evaluate the features intended to allow for simulating discriminative training described in section 2.3. Table 5 contains results (average BLEU scores, as before) for each feature on its own. The w and c parameters for the onevsall and sparserank features were optimized for devset BLEU separately for Arabic and Chinese, with resulting values $w = 0.9$ and 0.5; and $c = 0.05$ and 0.005 respectively. All but one of the features (avgnz) outperform the baseline, probably because all incorporate some mechanism for im-

method or feature	Arabic	Chinese
baseline	46.79	31.72
EM = mixtm-samp	48.13	34.01
avg	48.02	33.29
avgnz	46.46	31.66
support	48.27	32.88
max	47.13	32.72
onevsall(w)	**48.60**	**34.22**
sparserank(c)	48.15	33.38

Table 5: Performance of features for simulating discriminative training.

method or feature	Arabic	Chinese
baseline	46.79	31.72
mixtm-samp	48.13	34.01
onevsall(w)	48.60	34.22
all features	48.48	34.20
onevsall(09) + avgnz + support + max	**48.84**	——
onevsall(05) + avg + max + EM	——	**34.37**

Table 6: Feature combinations.

method	Arabic		Chinese	
	dev	test	dev	test
mixtm-samp	48.10	48.13	28.65	34.01
simplex	48.62	48.28	28.76	34.06
onevsall(w)	48.81	48.60	29.13	34.22
best	49.05	48.84	29.28	34.37

Table 7: Comparison of direct optimization for linear weights with feature simulation.

plicitly countering the harmful effects of large out-of-domain corpora such as the UN. Avgnz does not have this property because it will emphasize non-zero probabilities without connecting them to a particular component. Although some features appear to be competitive with EM, the only one that clearly outperforms it on both language pairs is onevsall. This is not particularly surprising, since ovevsall generates multiple values (one per sub-corpus) unlike all other features except sparserank (4 potential boolean features per sub-corpus). However, it is interesting to note that onevsall is essentially a *log-linear* mixture in which component models are heavily smoothed by linear interpolation.

Table 6 shows the combination of all features from table 5. Disappointingly, this does somewhat worse than onevsall on its own. To determine the cause, we compared the dev-set BLEU scores for onevsall with those for the full model. On Arabic these are 48.81 versus 49.00; and on Chinese they are 29.13 versus 29.46. Hence we conclude that MIRA is mildly overfitting on the full feature sets. This is not quite the whole story, however. To see if we could improve on onevsall, we used a greedy feature selection strategy, hillclimbing on dev-set score, and stopping when we reached a local maximum. The (language-pair-specific) results are shown in table 6. These improve over the full models and give small gains over onevsall. The corresponding dev-set scores are 49.05 and 29.28, which indicates that, at least for Arabic, MIRA was both overfitting and failing to properly optimize the full model.

Our final experiment is an attempt to determine how our loose simulation of discriminative training for linear mixtures compares with true discriminative training. Lacking a sophisticated implementation of the latter, we used a brute force downhill simplex search (Press et al., 2002). We first ran MIRA with mixtm-samp linear weights, then performed 200 simplex iterations to optimize linear weights with log-linear weights held constant, then ran MIRA to re-optimize log-linear weights. The results are shown in table 7. The simplex search was mildly successful in its attempt to improve dev-set BLEU scores, gaining 0.52 and 0.11 over the mixtm-samp baseline for Arabic and Chinese respectively. Since, unlike MIRA, it makes no provision for generalization beyond the dev set, it is not fair to compare its test scores to those of mixtm-samp; indeed, we note that its test performance is poor relative to dev. What is significant is that the feature-based methods—onevsall and best—do much better on the *dev set* than simplex. We conclude that it is very unlikely that a more sophisticated optimizer working with a better objective than raw BLEU would be able to find a set of linear weights that outperform the test results of our features.[6]

[6] As noted above, however, it almost certainly does not accomplish this feat while remaining within the space of linear combinations.

4 Related Work

Domain adaptation for SMT is currently a very active topic, encompassing a wide variety of approaches. TM linear mixture models of the kind we study here were first proposed by Foster et al (2010), and have been refined by (Sennrich, 2012b), who applied them to multiple sub-corpora and investigated alternative techniques such as count weighting to capture the amount of sub-corpus-internal evidence for each phrase pair. Sennrich (2012a) reports interesting results using clustering to automate the identification of coherent sub-corpora. (Aminzadeh et al., 2012) examine the interaction between adaptation and MAP smoothing, and compare different combining techniques. Other ways of combining various component models have also been proposed, including "fill-up" with a predefined preference order (Bisazza et al., 2011), using multiple decoder paths (Koehn and Schroeder, 2007), and ensemble combination of complete models in the decoder (Razmara et al., 2012). Mixture approaches have also been used for LM adaptation (Foster and Kuhn, 2007). Finally, data selection approaches (Axelrod et al., 2011) can be seen as an extreme form of mixture modeling in which weights are either 0 or 1.

5 Conclusion

Our main aim in this paper has been to investigate whether SMT features inspired by various prototypical linear mixtures can be used to simulate discriminative training of linear mixture models for TM adaptation. Because of the fact that linear mixtures occur within the log-probabilities assigned to phrase pairs, they are not accessible to standard tuning algorithms without non-trivial modification, and hence are usually trained non-discriminatively for maximum likelihood.

Our simulation is loose in the sense that its features are not strictly limited to the space available to linear mixtures; they can and do "cheat" by leaving that space in order to improve performance. We find that we can improve significantly on linear mixtures trained using maximum likelihood, by 0.7 BLEU on an Arabic to English task, and by 0.4 BLEU on a Chinese to English task. Most of the gain is achieved by a single set of features (onevsall), parameterized by a combining weight, in which there is one instance per component model which assigns that model the combining weight and interpolates it with a uniform combination of all other models. The onevsall features achieve higher BLEU score on a development set than does a linear mixture optimized directly using downhill simplex. We consider this to be strong evidence in favour of a conclusion that our approach would outperform true discriminative training for linear mixtures.

We have also proposed an enhancement to maximum likelihood training of linear mixtures that involves sampling input phrase tables in order to balance their size. This counters a strong bias in favour of large sub-corpora with unsmoothed component models that is especially harmful when these corpora are highly out-of-domain. Although our results above improve on the performance of linear mixtures, we believe that EM-trained linear mixtures still offer a simple and effective way to perform domain adaptation (we also note that one of our best sub-sets of adaptation features includes a maximum-likelihood combination).

References

Aminzadeh, A Ryan, Jennifer Drexler, Timothy Anderson, and Wade Shen. 2012. Improved phrase translation modeling using MAP adaptation. In *Text, Speech and Dialogue*. Springer.

Axelrod, Amittai, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *EMNLP 2011*.

Bisazza, A., N. Ruiz, and M. Federico. 2011. Fill-up versus interpolation methods for phrase-based SMT adaptation. In *IWSLT 2011*.

Chen, Boxing, Roland Kuhn, George Foster, and Howard Johnson. 2011. Unpacking and transforming feature functions: New ways to smooth phrase tables. In *MT Summit 2011*.

Chen, Boxing, George Foster, and Roland Kuhn. 2013. Adaptation of reordering models for statistical machine translation. In *NAACL 2013*.

Cherry, Colin and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *NAACL 2012*.

Chiang, David, Yuval Marton, and Philip Resnik. 2008. Online large-margin training of syntactic and structural translation features. In *EMNLP 2008*.

Foster, George and Roland Kuhn. 2007. Mixture-model adaptation for SMT. In *WMT 2007*.

Foster, George, Cyril Goutte, and Roland Kuhn. 2010. Discriminative instance weighting for domain adaptation in statistical machine translation. In *EMNLP 2010*.

Galley, Michel and C. D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *EMNLP 2008*, pages 848–856, Hawaii, October.

Haddow, Barry and Philipp Koehn. 2012. Analysing the effect of out-of-domain data on smt systems. In *WMT 2012*.

Koehn, Philipp 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, Prague, Czech Republic, June. Association for Computational Linguistics.

Koehn, P., H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL 2007, Demonstration Session*.

Och, Franz Josef. 2003. Minimum error rate training for statistical machine translation. In *Proceedings of the 41th Annual Meeting of the Association for Computational Linguistics (ACL)*, Sapporo, July. ACL.

Press, William H., Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. 2002. *Numerical Recipes in C++*. Cambridge University Press, Cambridge, UK.

Razmara, Majid, George Foster, Baskaran Sankaran, and Anoop Sarkar. 2012. Mixing multiple translation models in statistical machine translation. In *ACL 2012*.

Rosti, Antti-Veikko, Bing Zhang, Spyros Matsoukas, and Richard Schwartz. 2011. Expected BLEU training for graphs: BBN system description for WMT11 system combination task. In *WMT 2011*.

Sennrich, Rico. 2012a. Mixture-modeling with unsupervised clusters for domain adaptation in statistical machine translation. In *EAMT 2012*.

Sennrich, Rico. 2012b. Perplexity minimization for translation model domain adaptation in statistical machine translation. In *EACL 2012*.