Perplexity Minimization for Translation Model Domain Adaptation in Statistical Machine Translation

Rico Sennrich
Institute of Computational Linguistics
University of Zurich
Binzmühlestr. 14
CH-8050 Zürich
sennrich@cl.uzh.ch

Abstract

We investigate the problem of domain adaptation for parallel data in Statistical Machine Translation (SMT). While techniques for domain adaptation of monolingual data can be borrowed for parallel data, we explore conceptual differences between translation model and language model domain adaptation and their effect on performance, such as the fact that translation models typically consist of several features that have different characteristics and can be optimized separately. We also explore adapting multiple (4–10) data sets with no *a priori* distinction between in-domain and out-of-domain data except for an in-domain development set.

1 Introduction

The increasing availability of parallel corpora from various sources, welcome as it may be, leads to new challenges when building a statistical machine translation system for a specific domain. The task of determining which parallel texts should be included for training, and which ones hurt translation performance, is tedious when performed through trial-and-error. Alternatively, methods for a weighted combination exist, but there is conflicting evidence as to which approach works best, and the issue of determining weights is not adequately resolved.

The picture looks better in language modelling, where model interpolation through perplexity minimization has become a widespread method of domain adaptation. We investigate the applicability of this method for translation models, and discuss possible applications.

We move the focus away from a binary combination of in-domain and out-of-domain data. If we can scale up the number of models whose contributions we weight, this reduces the need for *a priori* knowledge about the fitness[1] of each potential training text, and opens new research opportunities, for instance experiments with clustered training data.

2 Domain Adaptation for Translation Models

To motivate efforts in domain adaptation, let us review why additional training data can improve, but also decrease translation quality.

Adding more training data to a translation system is easy to motivate through the data sparseness problem. Koehn and Knight (2001) show that translation quality correlates strongly with how often a word occurs in the training corpus. Rare words or phrases pose a problem in several stages of MT modelling, from word alignment to the computation of translation probabilities through Maximum Likelihood Estimation. Unknown words are typically copied verbatim to the target text, which may be a good strategy for named entities, but is often wrong otherwise. In general, more data allows for a better word alignment, a better estimation of translation probabilities, and for the consideration of more context (in phrase-based or syntactic SMT).

A second effect of additional data is not necessarily positive. Translations are inherently ambiguous, and a strong source of ambiguity is the

[1] We borrow this term from early evolutionary biology to emphasize that the question in domain adaptation is not how "good" or "bad" the data is, but how well-adapted it is to the task at hand.

Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, pages 539–549,
Avignon, France, April 23 - 27 2012.

domain of a text. The German word "Wort" (engl. *word*) is typically translated as *floor* in Europarl, a corpus of Parliamentary Proceedings (Koehn, 2005), owing to the high frequency of phrases such as *you have the floor*, which is translated into German as *Sie haben das Wort*. This translation is highly idiomatic and unlikely to occur in other contexts. Still, adding Europarl as out-of-domain training data shifts the probability distribution of $p(t|\text{"Wort"})$ in favour of $p(\text{"floor"}|\text{"Wort"})$, and may thus lead to improper translations.

We will refer to the two problems as the data sparseness problem and the ambiguity problem. Adding out-of-domain data typically mitigates the data sparseness problem, but exacerbates the ambiguity problem. The net gain (or loss) of adding more data changes from case to case. Because there are (to our knowledge) no tools that predict this net effect, it is a matter of empirical investigation (or, in less suave terms, trial-and-error), to determine which corpora to use.[2]

From this understanding of the reasons for and against out-of-domain data, we formulate the following hypotheses:

1. A weighted combination can control the contribution of the out-of-domain corpus on the probability distribution, and thus limit the ambiguity problem.

2. A weighted combination eliminates the need for data selection, offering a robust baseline for domain-specific machine translation.

We will discuss three mixture modelling techniques for translation models. Our aim is to adapt all four features of the standard Moses SMT translation model: the phrase translation probabilities $p(\bar{t}|\bar{s})$ and $p(\bar{s}|\bar{t})$, and the lexical weights $lex(\bar{t}|\bar{s})$ and $lex(\bar{s}|\bar{t})$.[3]

2.1 Linear Interpolation

A well-established approach in language modelling is the linear interpolation of several models, i.e. computing the weighted average of the individual model probabilities. It is defined as follows:

$$p(x|y;\lambda) = \sum_{i=1}^{n} \lambda_i p_i(x|y) \qquad (1)$$

with λ_i being the interpolation weight of each model i, and with $(\sum_i \lambda_i) = 1$.

For SMT, linear interpolation of translation models has been used in numerous systems. The approaches diverge in how they set the interpolation weights. Some authors use uniform weights (Cohn and Lapata, 2007), others empirically test different interpolation coefficients (Finch and Sumita, 2008; Yasuda et al., 2008; Nakov and Ng, 2009; Axelrod et al., 2011), others apply monolingual metrics to set the weights for TM interpolation (Foster and Kuhn, 2007; Koehn et al., 2010).

There are reasons against all these approaches. Uniform weights are easy to implement, but give little control. Empirically, it has been shown that they often do not perform optimally (Finch and Sumita, 2008; Yasuda et al., 2008). An optimization of BLEU scores on a development set is promising, but slow and impractical. There is no easy way to integrate linear interpolation into log-linear SMT frameworks and perform optimization through MERT. Monolingual optimization objectives such as language model perplexity have the advantage of being well-known and readily available, but their relation to the ambiguity problem is indirect at best.

Linear interpolation is seemingly well-defined in equation 1. Still, there are a few implementation details worth pointing out. If we directly interpolate each feature in the translation model, and define the feature values of non-occurring phrase pairs as 0, this disregards the meaning of each feature. If we estimate $p(x|y)$ via MLE as in equation 2, and $c(y) = 0$, then $p(x|y)$ is strictly speaking undefined. Alternatively to a naive algorithm, which treats unknown phrase pairs as having a probability of 0, which results in a deficient probability distribution, we propose and implement the following algorithm. For each value pair (x, y) for which we compute $p(x|y)$, we replace λ_i with 0 for all models i with $p(y) = 0$, then renormalize the weight vector λ to 1. We do this for $p(\bar{t}|\bar{s})$ and $lex(\bar{t}|\bar{s})$, but not for $p(\bar{s}|\bar{t})$ and $lex(\bar{s}|\bar{t})$, the reasoning being the con-

[2] A frustrating side-effect is that these findings rarely generalize. For instance, we were unable to reproduce the finding by Ceauşu et al. (2011) that patent translation systems are highly domain-sensitive and suffer from the inclusion of parallel training data from other patent subdomains.

[3] We can ignore the fifth feature, the phrase penalty, which is a constant.

sequences for perplexity minimization (see section 2.4). Namely, we do not want to penalize a small in-domain model for having a high out-of-vocabulary rate on the source side, but we do want to penalize models that know the source phrase, but not its correct translation. A second modification pertains to the lexical weights $lex(\overline{s}|\overline{t})$ and $lex(\overline{t}|\overline{s})$, which form no true probability distribution, but are derived from the individual word translation probabilities of a phrase pair (see (Koehn et al., 2003)). We propose to not interpolate the features directly, but the word translation probabilities which are the basis of the lexical weight computation. The reason for this is that word pairs are less sparse than phrase pairs, so that we can even compute lexical weights for phrase pairs which are unknown in a model.[4]

2.2 Weighted Counts

Weighting of different corpora can also be implemented through a modified Maximum Likelihood Estimation. The traditional equation for MLE is:

$$p(x|y) = \frac{c(x,y)}{c(y)} = \frac{c(x,y)}{\sum_{x'} c(x',y)} \quad (2)$$

where c denotes the count of an observation, and p the model probability. If we generalize the formula to compute a probability from n corpora, and assign a weight λ_i to each, we get[5]:

$$p(x|y;\lambda) = \frac{\sum_{i=1}^{n} \lambda_i c_i(x,y)}{\sum_{i=1}^{n} \sum_{x'} \lambda_i c_i(x',y)} \quad (3)$$

The main difference to linear interpolation is that this equation takes into account how well-evidenced a phrase pair is. This includes the distinction between lack of evidence and negative evidence, which is missing in a naive implementation of linear interpolation.

Translation models trained with weighted counts have been discussed before, and have been shown to outperform uniform ones in some settings. However, researchers who demonstrated this fact did so with arbitrary weights (e.g. (Koehn, 2002)), or by empirically testing different weights (e.g. (Nakov and Ng, 2009)). We do not know of any research on automatically determining weights for this method, or which is not limited to two corpora.

[4]For instance if the word pairs (the,der) and (man,Mann) are known, but the phrase pair (the man, der Mann) is not.

[5]Unlike equation 1, equation 3 does not require that $(\sum_i \lambda_i) = 1$.

2.3 Alternative Paths

A third method is using multiple translation models as alternative decoding paths (Birch et al., 2007), an idea which Koehn and Schroeder (2007) first used for domain adaptation. This approach has the attractive theoretical property that adding new models is guaranteed to lead to equal or better performance, given the right weights. At best, a model is beneficial with appropriate weights. At worst, we can set the feature weights so that the decoding paths of one model are never picked for the final translation. In practice, each translation model adds 5 features and thus 5 more dimensions to the weight space, which leads to longer search, search errors, and/or overfitting. The expectation is that, at least with MERT, using alternative decoding paths does not scale well to a high number of models.

A suboptimal choice of weights is not the only weakness of alternative paths, however. Let us assume that all models have the same weights. Note that, if a phrase pair occurs in several models, combining models through alternative paths means that the decoder selects the path with the highest probability, whereas with linear interpolation, the probability of the phrase pair would be the (weighted) average of all models. Selecting the highest-scoring phrase pair favours statistical outliers and hence is the less robust decision, prone to data noise and data sparseness.

2.4 Perplexity Minimization

In language modelling, perplexity is frequently used as a quality measure for language models (Chen and Goodman, 1998). Among other applications, language model perplexity has been used for domain adaptation (Foster and Kuhn, 2007). For translation models, perplexity is most closely associated with EM word alignment (Brown et al., 1993) and has been used to evaluate different alignment algorithms (Al-Onaizan et al., 1999).

We investigate translation model perplexity minimization as a method to set model weights in mixture modelling. For the purpose of optimization, the cross-entropy $H(p)$, the perplexity $2^{H(p)}$, and other derived measures are equivalent. The cross-entropy $H(p)$ is defined as:[6]

[6]See (Chen and Goodman, 1998) for a short discussion of the equation. In short, a lower cross-entropy indicates that the model is better able to predict the development set.

$$H(p) = -\sum_{x,y} \tilde{p}(x,y) \log_2 p(x|y) \quad (4)$$

The phrase pairs (x, y) whose probability we measure, and their empirical probability $\tilde{p}$ need to be extracted from a development set, whereas p is the model probability. To obtain the phrase pairs, we process the development set with the same word alignment and phrase extraction tools that we use for training, i.e. GIZA++ and heuristics for phrase extraction (Och and Ney, 2003). The objective function is the minimization of the cross-entropy, with the weight vector λ as argument:

$$\hat{\lambda} = \arg\min_{\lambda} -\sum_{x,y} \tilde{p}(x,y) \log_2 p(x|y;\lambda) \quad (5)$$

We can fill in equations 1 or 3 for $p(x|y;\lambda)$. The optimization itself is convex and can be done with off-the-shelf software.[7] We use L-BFGS with numerically approximated gradients (Byrd et al., 1995).

Perplexity minimization has the advantage that it is well-defined for both weighted counts and linear interpolation, and can be quickly computed. Other than in language modelling, where $p(x|y)$ is the probability of a word given a n-gram history, conditional probabilities in translation models express the probability of a target phrase given a source phrase (or vice versa), which connects the perplexity to the ambiguity problem. The higher the probability of "correct" phrase pairs, the lower the perplexity, and the more likely the model is to successfully resolve the ambiguity. The question is in how far perplexity minimization coincides with empirically good mixture weights.[8] This depends, among others, on the other model components in the SMT framework, for instance the language model. We will not evaluate perplexity minimization against empirically optimized mixture weights, but apply it in situations where the latter is infeasible, e.g. because of the number of models.

[7] A quick demonstration of convexity: equation 1 is affine; equation 3 linear-fractional. Both are convex in the domain $\mathbb{R}_{>0}$. Consequently, equation 4 is also convex because it is the weighted sum of convex functions.

[8] There are tasks for which perplexity is known to be unreliable, e.g. for comparing models with different vocabularies. However, such confounding factors do not affect the optimization algorithm, which works with a fixed set of phrase pairs, and merely varies λ.

Our main technical contributions are as follows: Additionally to perplexity optimization for linear interpolation, which was first applied by Foster et al. (2010), we propose perplexity optimization for weighted counts (equation 3), and a modified implementation of linear interpolation. Also, we independently perform perplexity minimization for all four features of the standard SMT translation model: the phrase translation probabilities $p(\bar{t}|\bar{s})$ and $p(\bar{s}|\bar{t})$, and the lexical weights $lex(\bar{t}|\bar{s})$ and $lex(\bar{s}|\bar{t})$.

3 Other Domain Adaptation Techniques

So far, we discussed mixture modelling for translation models, which is only a subset of domain adaptation techniques in SMT.

Mixture-modelling for language models is well established (Foster and Kuhn, 2007). Language model adaptation serves the same purpose as translation model adaptation, i.e. skewing the probability distribution in favour of in-domain translations. This means that LM adaptation may have similar effects as TM adaptation, and that the two are to some extent redundant. Foster and Kuhn (2007) find that "both TM and LM adaptation are effective", but that "combined LM and TM adaptation is not better than LM adaptation on its own".

A second strand of research in domain adaptation is data selection, i.e. choosing a subset of the training data that is considered more relevant for the task at hand. This has been done for language models using techniques from information retrieval (Zhao et al., 2004), or perplexity (Lin et al., 1997; Moore and Lewis, 2010). Data selection has also been proposed for translation models (Axelrod et al., 2011). Note that for translation models, data selection offers an unattractive trade-off between the data sparseness and the ambiguity problem, and that the optimal amount of data to select is hard to determine.

Our discussion of mixture-modelling is relatively coarse-grained, with 2-10 models being combined. Matsoukas et al. (2009) propose an approach where each sentence is weighted according to a classifier, and Foster et al. (2010) extend this approach by weighting individual phrase pairs. These more fine-grained methods need not be seen as alternatives to coarse-grained ones. Foster et al. (2010) combine the two, applying linear interpolation to combine the instance-

weighted out-of-domain model with an in-domain model.

4 Evaluation

Apart from measuring the performance of the approaches introduced in section 2, we want to investigate the following open research questions.

1. Does an implementation of linear interpolation that is more closely tailored to translation modelling outperform a naive implementation?

2. How do the approaches perform outside a binary setting, i.e. when we do not work with one in-domain and one out-of-domain model, but with a higher number of models?

3. Can we apply perplexity minimization to other translation model features such as the lexical weights, and if yes, does a separate optimization of each translation model feature improve performance?

4.1 Data and Methods

In terms of tools and techniques used, we mostly adhere to the work flow described for the WMT 2011 baseline system[9]. The main tools are Moses (Koehn et al., 2007), SRILM (Stolcke, 2002), and GIZA++ (Och and Ney, 2003), with settings as described in the WMT 2011 guide. We report two translation measures: BLEU (Papineni et al., 2002) and METEOR 1.3 (Denkowski and Lavie, 2011). All results are lowercased and tokenized, measured with five independent runs of MERT (Och and Ney, 2003) and MultEval (Clark et al., 2011) for resampling and significance testing.

We compare three baselines and four translation model mixture techniques. The three baselines are a purely in-domain model, a purely out-of-domain model, and a model trained on the concatenation of the two, which corresponds to equation 3 with uniform weights. Additionally, we evaluate perplexity optimization with weighted counts and the two implementations of linear interpolation contrasted in section 2.1. The two linear interpolations that are contrasted are a naive one, i.e. a direct, unnormalized interpolation of all translation model features, and a modified one that normalizes λ for each phrase pair $(\bar{s}, \bar{t})$ for $p(\bar{t}|\bar{s})$ and recomputes the lexical weights based on interpolated word translation probabilites. The fourth weighted combination is using alternative decoding paths with weights set through MERT. The four weighted combinations are evaluated twice: once applied to the original four or ten parallel data sets, once in a binary setting in which all out-of-domain data sets are first concatenated.

Since we want to concentrate on translation model domain adaptation, we keep other model components, namely word alignment and the lexical reordering model, constant throughout the experiments. We contrast two language models. An unadapted, out-of-domain language model trained on data sets provided for the WMT 2011 translation task, and an adapted language model which is the linear interpolation of all data sets, optimized for minimal perplexity on the in-domain development set.

While unadapted language models are becoming more rare in domain adaptation research, they allow us to contrast different TM mixtures without the effect on performance being (partially) hidden by language model adaptation with the same effect.

The first data set is a DE–FR translation scenario in the domain of mountaineering. The in-domain corpus is a collection of Alpine Club pub-

Data set	sentences	words (fr)
Alpine (in-domain)	220k	4 700k
Europarl	1 500k	44 000k
JRC Acquis	1 100k	24 000k
OpenSubtitles v2	2 300k	18 000k
Total train	5 200k	91 000k
Dev	1424	33 000
Test	991	21 000

Table 1: Parallel data sets for German – French translation task.

Data set	sentences	words
Alpine (in-domain)	650k	13 000k
News-commentary	150k	4 000k
Europarl	2 000k	60 000k
News	25 000k	610 000k
Total	28 000k	690 000k

Table 2: Monolingual French data sets for German – French translation task.

[9] http://www.statmt.org/wmt11/baseline.html

lications (Volk et al., 2010). As parallel out-of-domain dataset, we use Europarl, a collection of parliamentary proceedings (Koehn, 2005), JRC-Acquis, a collection of legislative texts (Steinberger et al., 2006), and OpenSubtitles v2, a parallel corpus extracted from film subtitles[10] (Tiedemann, 2009). For language modelling, we use in-domain data and data from the 2011 Workshop on Statistical Machine Translation. The respective sizes of the data sets are listed in tables 1 and 2.

As the second data set, we use the Haitian Creole – English data from the WMT 2011 featured translation task. It consists of emergency SMS sent in the wake of the 2010 Haiti earthquake. Originally, Microsoft Research and CMU operated under severe time constraints to build a translation system for this language pair. This limits the ability to empirically verify how much each data set contributes to translation quality, and increases the importance of automated and quick domain adaptation methods.

Note that both data sets have a relatively high ratio of in-domain to out-of-domain parallel training data (1:20 for DE–EN and 1:5 for HT–EN) Previous research has been performed with ratios of 1:100 (Foster et al., 2010) or 1:400 (Axelrod et al., 2011). Since domain adaptation becomes more important when the ratio of IN to OUT is low, and since such low ratios are also realistic[11], we also include results for which the amount of in-domain parallel data has been restricted to 10% of the available data set.

We used the same development set for language/translation model adaptation and setting the global model weights with MERT. While it is theoretically possible that MERT will give too high weights to models that are optimized on the same development set, we found no empirical evidence for this in experiments with separate development sets.

4.2 Results

The results are shown in tables 5 and 6. In the DE–FR translation task, results vary between 13.5 and 18.9 BLEU points; in the HT–EN task, between 24.3 and 33.8. Unsurprisingly, an adapted LM performs better than an out-of-domain one, and using all available in-domain parallel data is better than using only part of it. The same is not true for out-of-domain data, which highlights the problem discussed in the introduction. For the DE–FR task, adding 86 million words of out-of-domain parallel data to the 5 million in-domain data set does not lead to consistent performance gains. We observe a decrease of 0.3 BLEU points with an out-of-domain LM, and an increase of 0.4 BLEU points with an adapted LM. The out-of-domain training data has a larger positive effect if less in-domain data is available, with a gain of 1.4 BLEU points. The results in the HT–EN translation task (table 6) paint a similar picture. An interesting side note is that even tiny amounts of in-domain parallel data can have strong effects on performance. A training set of 1600 emergency SMS (38 000 tokens) yields a comparable performance to an out-of-domain data set of 1.5 million tokens.

As to the domain adaptation experiments, weights optimized through perplexity minimization are significantly better in the majority of cases, and never significantly worse, than uniform

Data set	units	words (en)
SMS (in-domain)	16 500	380 000
Medical	1 600	10 000
Newswire	13 500	330 000
Glossary	35 700	90 000
Wikipedia	8 500	110 000
Wikipedia NE	10 500	34 000
Bible	30 000	920 000
Haitisurf dict	3 700	4000
Krengle dict	1 600	2 600
Krengle	650	4 200
Total train	120 000	1 900 000
Dev	900	22 000
Test	1274	25 000

Table 3: Parallel data sets for Haiti Creole – English translation task.

Data set	sentences	words
SMS (in-domain)	16k	380k
News	113 000k	2 650 000k

Table 4: Monolingual English data sets for Haiti Creole – English translation task.

[10] http://www.opensubtitles.org

[11] We predict that the availability of parallel data will steadily increase, most data being out-of-domain for any given task.

System	out-of-domain LM full IN TM		adapted LM full IN TM		adapted LM small IN TM	
	BLEU	METEOR	BLEU	METEOR	BLEU	METEOR
in-domain	16.8	35.9	17.9	37.0	15.7	33.5
out-of-domain	13.5	31.3	14.8	32.3	14.8	32.3
counts (concatenation)	16.5	35.7	18.3	37.3	17.1	35.4
binary in/out						
weighted counts	17.4	36.6	18.7	37.9	17.6	36.2
linear interpolation (naive)	17.4	36.7	18.8	37.9	17.6	36.1
linear interpolation (modified)	17.2	36.5	18.9	38.0	17.6	36.2
alternative paths	17.2	36.5	18.6	37.8	17.4	36.0
4 models						
weighted counts	17.3	36.6	18.8	37.8	17.4	36.0
linear interpolation (naive)	17.1	36.5	18.5	37.7	17.3	35.9
linear interpolation (modified)	17.2	36.5	18.7	37.9	17.3	36.0
alternative paths	17.0	36.2	18.3	37.4	16.3	35.1

Table 5: Domain adaptation results DE–FR. Domain: Alpine texts. Full IN TM: Using the full in-domain parallel corpus; small IN TM: using 10% of available in-domain parallel data.

weights.[12] However, the difference is smaller for the experiments with an adapted language model than for those with an out-of-domain one, which confirms that the benefit of language model adaptation and translation model adaptation are not fully cumulative. Performance-wise, there seems to be no clear winner between weighted counts and the two alternative implementations of linear interpolation. We can still argue for weighted counts on theoretical grounds. A weighted MLE (equation 3) returns a true probability distribution, whereas a naive implementation of linear interpolation results in a deficient model. Consequently, probabilities are typically lower in the naively interpolated model, which results in higher (worse) perplexities. While the deficiency did not affect MERT or decoding negatively, it might become problematic in other applications, for instance if we want to use an interpolated model as a component in a second perplexity-based combination of models.[13]

When moving from a binary setting with one in-domain and one out-of-domain translation model (trained on all available out-of-domain data) to 4–10 translation models, we observe a serious performance degradation for alternative paths, while performance of the perplexity optimization methods does not change significantly. This is positive for perplexity optimization because it demonstrates that it requires less *a priori* information, and opens up new research possibilities, i.e. experiments with different clusterings of parallel data. The performance degradation for alternative paths is partially due to optimization problems in MERT, but also due to a higher susceptibility to statistical outliers, as discussed in section 2.3.[14]

A pessimistic interpretation of the results would point out that performance gains compared to the best baseline system are modest or even inexistent in some settings. However, we want to stress two important points. First, we often do not know *a priori* whether adding an out-of-domain data set boosts or weakens translation performance. An automatic weighting of data sets reduces the need for trial-and-error experimentation and is worthwhile even if a performance increase is not guaranteed. Second, the potential impact of a weighted combination depends on the translation scenario and the available data sets. Generally, we expect non-uniform weighting to have a bigger impact when the models that are combined are more dissimilar (in terms of fitness for the task), and if the ratio of in-domain to out-of-domain data is low. Conversely, there are situa-

[12] This also applies to linear interpolation with uniform weights, which is not shown in the tables.

[13] Specifically, a deficient model would be dispreferred by the perplexity minimization algorithm.

[14] We empirically verified this weakness in a synthetic experiment with a randomly split training corpus and identical weights for each path.

System	out-of-domain LM full IN TM		adapted LM full IN TM		adapted LM small IN TM	
	BLEU	METEOR	BLEU	METEOR	BLEU	METEOR
in-domain	30.4	30.7	33.4	31.7	29.7	28.6
out-of-domain	24.3	28.0	28.9	30.2	28.9	30.2
counts (concatenation)	30.3	31.2	33.6	32.4	31.3	31.3
binary in/out						
weighted counts	31.0	31.6	33.8	32.4	31.5	31.3
linear interpolation (naive)	30.8	31.4	33.7	32.4	31.9	31.3
linear interpolation (modified)	30.8	31.5	33.7	32.4	31.7	31.2
alternative paths	30.8	31.3	33.2	32.4	29.8	30.7
10 models						
weighted counts	31.0	31.5	33.5	32.3	31.8	31.5
linear interpolation (naive)	30.9	31.4	33.8	32.4	31.9	31.3
linear interpolation (modified)	31.0	31.6	33.8	32.5	32.1	31.5
alternative paths	25.9	29.2	24.3	29.1	29.8	30.9

Table 6: Domain adaptation results HT–EN. Domain: emergency SMS. Full IN TM: Using the full in-domain parallel corpus; small IN TM: using 10% of available in-domain parallel data.

tions where we actually expect a simple concatenation to be optimal, e.g. when the data sets have very similar probability distributions.

4.2.1 Individually Optimizing Each TM Feature

It is hard to empirically show how translation model perplexity optimization compares to using monolingual perplexity measures for the purpose of weighting translation models, as e.g. done by (Foster and Kuhn, 2007; Koehn et al., 2010). One problem is that there are many different possible configurations for the latter. We can use source side or target side language models, operate with different vocabularies, smoothing techniques, and n-gram orders.

One of the theoretical considerations that favour measuring perplexity on the translation model rather than using monolingual measures is that we can optimize each translation model feature separately. In the default Moses translation model, the four features are $p(\overline{s}|\overline{t})$, $lex(\overline{s}|\overline{t})$, $p(\overline{t}|\overline{s})$ and $lex(\overline{t}|\overline{s})$.

We empirically test different optimization schemes as follows. We optimize perplexity on each feature independently, obtaining 4 weight vectors. We then compute one model with one weight vector per feature (namely the feature that the vector was optimized on), and four models that use one of the weight vectors for all features. A further model uses a weight vector that is the

weights	perplexity 1	2	3	4	BLEU
weighted counts					
uniform	5.12	7.68	4.84	13.67	30.3
separate	**4.68**	**6.62**	**4.24**	**8.57**	**31.0**
1	**4.68**	6.84	4.50	10.86	30.3
2	4.78	**6.62**	4.48	10.54	30.3
3	4.86	7.31	**4.24**	9.15	30.8
4	5.33	7.87	4.52	**8.57**	30.9
average	4.72	6.71	4.38	9.95	30.4
linear interpolation (modified)					
uniform	19.89	82.78	4.80	10.78	30.6
separate	**5.45**	**8.56**	**4.28**	**8.85**	**31.0**
1	**5.45**	8.79	4.40	8.89	30.8
2	5.71	**8.56**	4.54	8.91	30.9
3	6.46	11.88	**4.28**	9.07	31.0
4	6.12	10.86	4.47	**8.85**	30.9
average	5.73	9.72	4.34	8.89	30.9
LM	6.01	9.83	4.56	8.96	30.8

Table 7: Contrast between a separate optimization of each feature and applying the weight vector optimized on one feature to the whole model. HT–EN with out-of-domain LM.

average of the other four. For linear interpolation, we also include a model whose weights have been optimized through language model perplexity optimization, with a 3-gram language model (modified Knesey-Ney smoothing) trained on the target side of each parallel data set.

Table 7 shows the results. In terms of BLEU score, a separate optimization of each feature is a winner in our experiment in that no other scheme is better, with 8 of the 11 alternative weighting schemes (excluding uniform weights) being significantly worse than a separate optimization. The differences in BLEU score are small, however, since the alternative weighting schemes are generally felicitious in that they yield both a lower perplexity and better BLEU scores than uniform weighting. While our general expectation is that lower perplexities correlate with higher translation performance, this relation is complicated by several facts. Since the interpolated models are deficient (i.e. their probabilities do not sum to 1), perplexities for weighted counts and our implementation of linear interpolation cannot be compard. Also, note that not all features are equally important for decoding. Their weights in the log-linear model are set through MERT and vary between optimization runs.

5 Conclusion

This paper contributes to SMT domain adaptation research in several ways. We expand on work by (Foster et al., 2010) in establishing translation model perplexity minimization as a robust baseline for a weighted combination of translation models.[15] We demonstrate perplexity optimization for weighted counts, which are a natural extension of unadapted MLE training, but are of little prominence in domain adaptation research. We also show that we can separately optimize the four variable features in the Moses translation model through perplexity optimization.

We break with prior domain adaptation research in that we do not rely on a binary clustering of in-domain and out-of-domain training data. We demonstrate that perplexity minimization scales well to a higher number of translation models. This is not only useful for domain adaptation, but for various tasks that profit from mixture modelling. We envision that a weighted combination could be useful to deal with noisy datasets, or applied after a clustering of training data.

[15]The source code is available in the Moses repository `http://github.com/moses-smt/mosesdecoder`

Acknowledgements

This research was funded by the Swiss National Science Foundation under grant 105215_126999.

References

Yaser Al-Onaizan, Jan Curin, Michael Jahr, Kevin Knight, John Lafferty, Dan Melamed, Franz-Josef Och, David Purdy, Noah A. Smith, and David Yarowsky. 1999. Statistical machine translation. Technical report, Final Report, JHU Summer Workshop.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the EMNLP 2011 Workshop on Statistical Machine Translation*.

Alexandra Birch, Miles Osborne, and Philipp Koehn. 2007. CCG supertags in factored statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 9–16, Prague, Czech Republic, June. Association for Computational Linguistics.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. 1995. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.*, 16:1190–1208, September.

Alexandru Ceauşu, John Tinsley, Jian Zhang, and Andy Way. 2011. Experiments on domain adaptation for patent machine translation in the PLuTO project. In *Proceedings of the 15th conference of the European Association for Machine Translation*, Leuven, Belgium.

Stanley F. Chen and Joshua Goodman. 1998. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13:359–393.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 176–181, Portland, Oregon, USA, June. Association for Computational Linguistics.

Trevor Cohn and Mirella Lapata. 2007. Machine Translation by Triangulation: Making Effective Use of Multi-Parallel Corpora. In *Proceedings of the*

45th Annual Meeting of the Association of Computational Linguistics, pages 728–735, Prague, Czech Republic, June. Association for Computational Linguistics.

Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic Metric for Reliable Optimization and Evaluation of Machine Translation Systems. In *Proceedings of the EMNLP 2011 Workshop on Statistical Machine Translation*.

Andrew Finch and Eiichiro Sumita. 2008. Dynamic model interpolation for statistical machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, StatMT '08, pages 208–215, Stroudsburg, PA, USA. Association for Computational Linguistics.

George Foster and Roland Kuhn. 2007. Mixture-model adaptation for smt. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, pages 128–135, Stroudsburg, PA, USA. Association for Computational Linguistics.

George Foster, Cyril Goutte, and Roland Kuhn. 2010. Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 451–459, Stroudsburg, PA, USA. Association for Computational Linguistics.

Philipp Koehn and Kevin Knight. 2001. Knowledge sources for word-level translation models. In Lillian Lee and Donna Harman, editors, *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, pages 27–35.

Philipp Koehn and Josh Schroeder. 2007. Experiments in domain adaptation for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, pages 224–227, Stroudsburg, PA, USA. Association for Computational Linguistics.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54, Morristown, NJ, USA. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June. Association for Computational Linguistics.

Philipp Koehn, Barry Haddow, Philip Williams, and Hieu Hoang. 2010. More linguistic annotation for statistical machine translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 115–120, Uppsala, Sweden, July. Association for Computational Linguistics.

Philipp Koehn. 2002. Europarl: A Multilingual Corpus for Evaluation of Machine Translation.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Machine Translation Summit X*, pages 79–86, Phuket, Thailand.

Sung-Chien Lin, Chi-Lung Tsai, Lee-Feng Chien, Keh-Jiann Chen, and Lin-Shan Lee. 1997. Chinese language model adaptation based on document classification and multiple domain-specific language models. In George Kokkinakis, Nikos Fakotakis, and Evangelos Dermatas, editors, *EUROSPEECH*. ISCA.

Spyros Matsoukas, Antti-Veikko I. Rosti, and Bing Zhang. 2009. Discriminative corpus weight estimation for machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, pages 708–717, Stroudsburg, PA, USA. Association for Computational Linguistics.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, ACLShort '10, pages 220–224, Stroudsburg, PA, USA. Association for Computational Linguistics.

Preslav Nakov and Hwee Tou Ng. 2009. Improved statistical machine translation for resource-poor languages using related resource-rich languages. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3*, EMNLP '09, pages 1358–1367, Stroudsburg, PA, USA. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Morristown, NJ, USA. Association for Computational Linguistics.

Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaz Erjavec, Dan Tufis, and Daniel Varga. 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'2006)*.

A. Stolcke. 2002. SRILM – An Extensible Language Modeling Toolkit. In *Seventh International Conference on Spoken Language Processing*, pages 901–904, Denver, CO, USA.

Jörg Tiedemann. 2009. News from opus - a collection of multilingual parallel corpora with tools and interfaces. In N. Nicolov, K. Bontcheva, G. Angelova, and R. Mitkov, editors, *Recent Advances in Natural Language Processing*, volume V, pages 237–248. John Benjamins, Amsterdam/Philadelphia, Borovets, Bulgaria.

Martin Volk, Noah Bubenhofer, Adrian Althaus, Maya Bangerter, Lenz Furrer, and Beni Ruef. 2010. Challenges in building a multilingual alpine heritage corpus. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

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 3rd International Joint Conference on Natural Language Processing (IJCNLP)*.

Bing Zhao, Matthias Eck, and Stephan Vogel. 2004. Language model adaptation for statistical machine translation with structured query models. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA. Association for Computational Linguistics.