A General Framework to Weight Heterogeneous Parallel Data for Model Adaptation in Statistical Machine Translation

Kashif Shah, Loïc Barrault, Holger Schwenk
LIUM, University of Le Mans
Le Mans, France.
FirstName.LastName@lium.univ-lemans.fr

Abstract

The standard procedure to train the translation model of a phrase-based SMT system is to concatenate all available parallel data, to perform word alignment, to extract phrase pairs and to calculate translation probabilities by simple relative frequency. However, parallel data is quite inhomogeneous in many practical applications with respect to several factors like data source, alignment quality, appropriateness to the task, etc. We propose a general framework to take into account these factors during the calculation of the phrase-table, e.g. by better distributing the probability mass of the individual phrase pairs. No additional feature functions are needed. We report results on two well-known tasks: the IWSLT'11 and WMT'11 evaluations, in both conditions translating from English to French. We give detailed results for different functions to weight the bitexts. Our best systems improve a strong baseline by up to one BLEU point without any impact on the computational complexity during training or decoding.

1 Introduction

Two type of resources are needed to build statistical machine translation (SMT) systems: monolingual texts to build a language model (LM) and bilingual texts – also called bitexts – to train the translation model (TM). While huge amounts of monolingual data are available in large variety of domains, parallel data is a sparse resource in many domains. The parallel data often comes from international organizations, *e.g.* Europarl, UN, or data crawled from the Internet or even bitexts automatically extracted from comparable corpora. Usually, parallel data really relevant to the translation task is only available in limited amount. The performance of an SMT system for a particular translation task heavily depends upon the appropriateness and usefulness of the data used to build the models. It is a common practice to concatenate all available parallel data, to perform word alignment, to extract phrase pairs and to calculate translation probabilities by simple relative frequency. The parallel data is incorrectly assumed to be uniform with respect to several aspects. It seem obvious that not all available bitexts are relevant to the translation task, usually called in-domain versus out-of domain data. Even within one corpus, eventually considered to be in-domain, there is no reason to assume that all the parallel sentences are equally appropriate. The genre of the data may be also different, for instance, scientific text is translated with the models trained mainly on news data. Similarly, the quality of the data may differ when considering human translations versus automatically crawled data from the web. Moreover, in certain domains it is worth to consider the temporal distance of the data with respect to the task - also called *recency effect*.

Considering all these factors, model adaptation is a topic of increasing interest and various techniques are proposed in literature. One way to adapt the translation model is to use mixture models (Civera and Juan, 2007; Zhao et al., 2004a; Foster and Kuhn, 2007; Koehn and Schroeder, 2007), or to perform self-enhancement (Ueffing, 2006; Ueffing, 2007; Chen et al., 2008), or more generally unsupervised-training (Schwenk, 2008; Nicola Bertoldi, 2009; Lambert et al., 2011; Bojar and Tamchyna, 2011). Most recently weighting the data is getting much attention from the research community. Various *goodness scores* extracted at different levels during the model training are considered to weight the data. The data with a higher goodness scores is given higher weights to have positive impact on translation quality.

Matsoukas et al. (2009) proposed a technique in which they weighted each sentence in the training bitexts to optimize a discriminative function on a given tuning set. Sentence level features were extracted to estimate the weights that are relevant to the given task. The feature vectors were mapped to scalar weights $(0, 1)$ which are then used to estimate probability with weighted counts.

Foster et al. (2010) proposed an extended approach by an instant weighting scheme which learns weights on individual phrase pairs instead of sentences and incorporated the instance-weighting model into a linear combination. Phillips and Brown (2011) trained the models with a second-order Taylor approximation of weighted translation instances and discount models on the basis of this approximation. Zhao et al. (2004b) rescore phrase translation pairs for statistical machine translation using tf.idf to encode the weights in phrase translation pairs. The translation probability is then modeled by similarity functions defined in a vector space. Huang and Xiang (2010) proposed a rescoring algorithm in which phrase pair features are combined with linear regression model and neural network to predict the quality score of the phrase translation pair. These phrase scores are used to boost good phrase translations and bad translations are discarded. Shah et al. (2010) proposed a technique to weight heterogeneous data by weighted resampling of the alignments. In an extended work, the same authors proposed to consider meta-weights for each part of the training data (Shah et al., 2011).

The work proposed in this paper is an extension and generalization of several ideas proposed in previous works such as weighted counts with goodness scores. However our proposed framework gives the flexibility to inject the goodness scores in a unified formulation calculated at various levels. It is based on the following principles:

- the use of a set of "quality measures" at different levels: weights for each corpus (or data source) and for each individual sentence in the bitexts.
- no additional feature functions to express the quality or appropriateness of certain phrase pairs, but we modify only the existing phrase probabilities. By these means, we don't have to deal with the additional complexity of decoding and optimizing the weights of many feature functions.
- resampling the bitexts or alignments is computationally expensive for large corpora since the resampled data is ideally much bigger than the original one. Instead, we integrate the various weighting schemes directly in the calculation of the translation probabilities.
- our approach has only a small number of parameter to optimize.

We unified several ideas into one efficient and simple framework that can be easily used. We clearly distinguished between weighting at different levels i.e corpus level and sentence level, contrary to the approach discussed in (Matsoukas et al., 2009). Moreover, they used a neural network to map the sentence level feature scores to a single weight. Our approach does not need any such mapping of features, goodness scores are directly used to produce weighted counts. The proposed framework doesn't depend on how these weights or goodness scores are calculated - one can use any measure which predicts the relevance of the training data to a given domain. Further, the proposed technique has the ability to take into account the goodness scores extracted at any level i.e corpus level, sentence level, word-to-word alignment level or even phrase level.

The rest of the paper is organized as follows. In the next section we present in detail the architecture of our approach. Experimental results for IWSLT'11 and WMT'11 task are summarized and discussed in section 3. The paper concludes with a discussion and perspective on this work.

2 Architecture of our approach

In the following we will first summarize how the phrase-table is calculated in the popular Moses SMT toolkit. Each research team has its own (undocumented) heuristics, but we assume that the basic procedure is very similar for most phrase-based systems. To formalize our ideas, let us assume that we translate a sentence from the source language $s = s_i \dots s_I$ to the target language $t = t_i \dots t_J$. A phrase $\tilde{s}$ or $\tilde{t}$ is a sequence of one or more words in the source or target language language respectively. The phrase table is a large collection of phrase pairs $(\tilde{s}, \tilde{t})$. Note that each source phrase $\tilde{s}_i$ generally has several translations $\tilde{t}_{ij}$. For each phrase pair $(\tilde{s}, \tilde{t})$, we usually have several probabilities used to weight it. Moses uses four probabilities: the forward

phrase-translation probability $P(\tilde{t}|\tilde{s})$, the backward phrase-translation probability $P(\tilde{s}|\tilde{t})$, and two lexical probabilities, again in the forward and backward direction. These probabilities are used in the standard log-linear model as feature functions $f_i(s,t)$:

$$t^* = \arg\max_t \sum_i \lambda_i \log f_i(s,t) \quad (1)$$

Moses uses in total fourteen feature functions: the above mentioned four scores for the phrases, a phrase and word penalty, six scores for the lexicalized distortion model, a language model score and a distance based reordering model.

The phrase-table itself is created by the following procedure

1. collect parallel training data
2. eventually discard sentence pairs that are too long or which have a large length difference
3. run Giza++ on this data in both directions (source-to-target and target-to-source)
4. use some heuristics to symmetrize the alignments in both directions, e.g. the so-called *grow-diagonal-...* (Koehn et al., 2003) and extract a list of phrases
5. calculate the lexical probabilities
6. calculate the phrase probabilities $P(\tilde{t}|\tilde{s})$ and $P(\tilde{s}|\tilde{t})$.
7. create the phrase table by merging the forward and backward probabilities

In our approach we only modify the way how the phrase translations probabilities $P(\tilde{t}|\tilde{s})$ and $P(\tilde{s}|\tilde{t})$ are calculated. The goal is to increase the probability of phrase pairs which we believe to be more important for the considered task, to be more reliable, etc; and consequently, to down weight those which should be used less often. It is important to point out that our phrase table has exactly the same number of entries as the original one and that we do not add more feature functions. Currently, we do not modify the lexical scores of each phrase pair, but we will investigate this in the future. In summary, we only modify step 6 in the above procedure. For this we modified the tool `memscore` (Hardmeier, 2010).

In practice, we also need to adapt step 4 since we need to keep track for each phrase pair from which corpus it was extracted and what are the scores of the corresponding sentence.

2.1 Standard phrase probabilities

The standard procedure to calculate the phrase probabilities is simple relative frequency:

$$P(\tilde{t}_{ij}|\tilde{s}_i) = \frac{Count(\tilde{s}_i, \tilde{t}_{ij})}{\sum_k Count(\tilde{s}_i, \tilde{t}_{ik})} \quad (2)$$

The `memscore` tool also implements various smoothing methods such as Witten-Bell, Kneser-Ney discounting etc. but to the best of our knowledge, their eventual benefit was not extensively studied and these smoothing techniques are not widely used. In any case, the calculation of the phrase probabilities does not consider from which corpus the phrase was extracted, or more generally, any kind of weight that was attached to the originating sentence.

This can obviously lead to wrong probability distributions. As a simple example we can consider a phrase pair $ps_i, \tilde{t}_{ij}$ which appears a couple of times in the in-domain corpus, and which provides the correct translation for the task, and another phrase pair $ps_i, \tilde{t}_{ik}$ which appears many times in a (larger) out of-domain corpus. This wrong translation will wrongly get a higher probability when relative frequency estimates are used (or any of the standard smoothing techniques).

A similar argument holds at the sentence or even phrase level, for instance even a generally in-domain corpus can contain few sentences which are out-of topic, badly aligned, etc.

2.2 Weighted phrase probabilities

We have modified the `memscore` tool in order to take into account a weight attached to each corpus and let us assume that we have the following information on our parallel training data:

- the parallel data can be organized into C different parts. In most of the cases, we will use the source of the data to partition it, e.g. Europarl, United Nations, web-crawled, but one could also use some kind of clustering algorithm. We associate the weight w_c, to each corpus c=1 $\ldots C$. We will discuss later how to obtain those weights.

- a set of S "goodness scores" $q_s(s_i, t_i), s = 1 \ldots S$ for each parallel sentence pair $(s_i, t_i), i = 1 \ldots L$ where L is the number of parallel sentences. Again, we will delay for now how to produce those sentence scores. We keep track of these sentence scores when extracting phrases. All the phrases extracted from the same sentence obtain the same phrase-level goodness scores $h_s(s_j, t_j), j = 1 \ldots P$ where $P \gg S$ is the number of extracted phrases.

Using these notations we will calculate the phrase probability as follows. Let us first consider only the weights of the individual corpora. This is achieved by extending equation 2 as follows:

$$P(\tilde{t}_{ij}|\tilde{s}_i) = \frac{\sum_{c=1}^{C} w_c Count_c(\tilde{s}_i, \tilde{t}_{ij})}{\sum_{c=1}^{C} w_c \sum_k Count_c(\tilde{s}_i, \tilde{t}_{ik})} \quad (3)$$

The equation 3 is identical as given in (Matsoukas et al., 2009), where w_c represents the features-mapped to a weight calculated for each sentence by neural network. However in our case it represents the direct weight for each corpus. If all corpus weights are identical, equation 3 simplifies to the original formulation in equation 2. Considering in addition the goodness scores at the sentence level, we will get:

$$P(\tilde{t}_{ij}|\tilde{s}_i) = \frac{\sum_{c=1}^{C} \left\{ w_c Count_c(\tilde{s}_i, \tilde{t}_{ij}) \cdot \prod_{s=1}^{S} h_{c,s}^{\gamma_s}(\tilde{s}_i, \tilde{t}_{ij}) \right\}}{\sum_{c=1}^{C} w_c \sum_k \left\{ Count_c(\tilde{s}_i, \tilde{t}_{ik}) \cdot \prod_{s=1}^{S} h_{c,s}^{\gamma_s}(\tilde{s}_i, \tilde{t}_{ik}) \right\}} \quad (4)$$

where γ_s is an additional parameter to weight the different sentence goodness scores among each other. We implemented phrase probability calculation according to equation 4 in the the `memscore` tool of Moses.

2.3 Calculation of the corpus weights and sentence goodness scores

Our theoretical framework and implementation is generic and does not depend on the exact calculation of the corpus weights or the sentence goodness scores. Any value that expresses the appropriateness of the corpus and sentence with respect to the task can be used. In the following we outline some possibilities which were used in our experiments.

Weighting parallel corpora was already investigated previously in the literature. For instance Shah et al. (2010) used a resampling technique to weight parallel corpora. They have proposed two methods to obtain the corpus weights: via LM interpolation and numerical optimization to maximize the BLEU score on some development data. The second approach showed slightly better performance, but it is computationally quite expensive (a new phrase table must be build for each optimization loop). Therefore, we decided to use corpus weights obtained by LM interpolation in our experiments. The idea is to build a LM on the source (or target) side of the bitexts, independently for each corpus. There is a well known EM procedure to linearly interpolate these individual LMs to minimize the perplexity on some development data. The resulting corpus coefficients can be directly used to weight the parallel corpora.

Perplexity can also be used to weight each individual sentence. This was used to select a relevant subset of LM data (Axelrod et al., 2011) or bitexts (Moore and Lewis, 2010). In our case, we build a LM on the source side of the in-domain corpus and use this model to calculate the perplexity of each sentence in all the other corpora. Since lower perplexity represents "better" sentences, we set $q(s_i, t_i)$ to the inverse of the perplexity. It is important to note that our approach is a generalization of data selection approaches: instead of doing a hard decision which data to keep to discard, we keep all the sentences and attach a weight to each one (this weight could be zero in an extreme case).

It was also observed that parallel sentences which are closer to the test set period are more important than older ones (Hardt and Elming, 2010; Levenberg et al., 2010; Shah et al., 2011), in particular when translating texts in the news domain. Following (Shah et al., 2011), we use an exponential decay function or this goodness function:

$$q(s_i, t_i) = e^{-\alpha \cdot \Delta t} i \quad (5)$$

where α is the decay factor and Δt is the discretized time distance (0 for most recent part, 1 for the next one, etc.).

Finally, it was argued that the alignment score produced by Giza++ could be used as a measure whether the phrases extracted from the corresponding sentence pair should be up- or down-

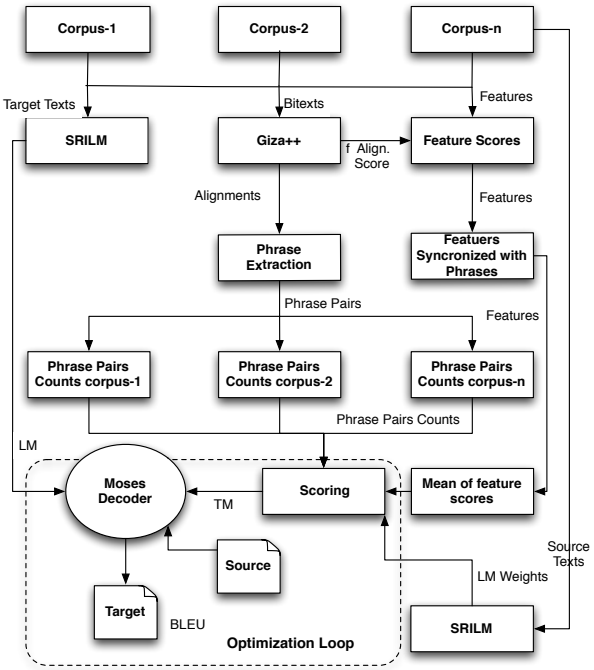

Figure 1: Overall architecture of our approach.

weighted. In order to ease comparison, we used the same equation as Shah et al. (2010):

$$q(s_i, t_i) = \log(\beta \cdot \frac{(\sqrt[n_{trg}]{a_{src\_trg}} + \sqrt[n_{src}]{a_{trg\_src}})}{2}) \quad (6)$$

where a is the alignment score, n the size of a sentence and β a smoothing coefficient to optimize.

2.4 Overall architecture

The overall architecture of our approach is given in figure 1. Suppose we have number of parallel corpora coming from various sources. First of all, sentence level goodness scores are calculated and synchronized with the parallel sentences. These sentences are concatenated to perform word-to-word alignment in both directions using GIZA++. This is done on the concatenated sentences since GIZA++ may perform badly if some of the individual bitexts are rather small. Alignment scores corresponding to each sentence pair are added to the goodness scores file. Then, phrases are extracted and the goodness score $q(s_i, t_i)$ is synchronized with the phrases. In the case that one phrase occurs in multiple sentences (this actually happens quite often), we use the arithmetic mean of the goodness scores in our experiments. The maximum or some other interpolation functions could be also used. Finally, the phrase-translation probabilities are calculated according to equation 4 in forward and backward direction.

The parameters of our approach α , β and γ along with w_c are numerically optimized [1]. In this optimization loop we keep the weights of the feature functions constant, i.e. λ_i in equation 1 (we use the ones of the standard system without weighted phrase translation probabilities). Eventually, these weights optimized using the standard MERT procedure once we have fixed the parameters of our approach.

Corpus	En tokens	Fr tokens
TED	2.0	2.2
News-Commentary	2.8	3.3
Europarl v6	50.6	56.2
ccb2	232.5	272.6
TOTAL	287.9	334.3

Table 1: Size of parallel corpora (in millions) to build baseline systems for WMT and IWSLT Tasks.

3 Experimental evaluation

We have built several phrase-based systems using the Moses toolkit (Koehn et al., 2007). The scoring framework is implemented by extending the memory based scoring tool called `memscore` (Hardmeier, 2010) available in the Moses toolkit. In our system, fourteen feature functions are used. These feature functions include phrase and lexical translation probabilities in both directions, seven features for the lexicalized distortion model, a word and phrase penalty, and the target language model. The MERT tool (Och, 2003) is used to tune the coefficients of these feature functions. The experiments are performed on two well-known evaluation tasks *i.e.* the 2011 WMT and IWSLT English/French evaluations. The corpora and their sizes used to build the systems for both these tasks are given in table 1.

3.1 Experiments on the WMT task

For the WMT task we used the official development sets of the 2011 WMT translation tasks, *i.e* news-test09 as development corpus and news-test10 as test corpus. We built English-French systems by using the time-stamped *Europarl* and news-commentary (*nc*) corpora. The LM is created by interpolating several language models

[1] with CONDOR (Berghen and Bersini, 2005)

WMT Task	Corpus weights	Alignment scores	Temporal distance	perplexity	BLEU (on test)
Baseline					28.16
System 1	yes				28.41
System 2		yes			28.21
System 3			yes		28.35
System 4				yes	28.56
System 5	yes	yes			28.55
System 6	yes		yes		28.60
System 7	yes			yes	28.61
System 8			yes	yes	28.79
System 9	yes	yes	yes		28.65
System 10	yes	yes		yes	28.67
System 11	yes		yes	yes	**28.89**
System 12	yes		yes	yes	**29.11 (optimized)**

Table 2: BLEU scores obtained with systems trained with different goodness scores on WMT Task.

trained separately on the target side of the bitexts and all available target language monolingual data (about 1.5G words). These individual language models are interpolated and merged into one huge model. The coefficient of the individual models are optimized using the usual EM procedure to minimize perplexity on the development data. Initial corpus weights for the bitexts were obtained by building another interpolated LM on the target side of the bitexts only.

We explored the following goodness scores to weight the relevance of the bitexts and the individual sentences: corpus weights, alignment scores, recency of the data with respect to the test set period and the sentence perplexity in the target language with respect to an in-domain language model. The news-commentary (*nc*) corpus was used for that purpose. The time information provided with *Europarl* data is used to estimate recency feature. This information was not available for *nc*, so we considered the sentences in chronologically ordered with respect to temporal distance. The alignment scores provided by GIZA++ were normalized using equation 6.

The results of the baseline system and various combinations of the different goodness scores are summarized in table 2. In order to get an idea which goodness score give best results, we have first performed experiments using default values for the parameters of the feature functions. For this purpose, we have used the values reported to be optimal in (Shah et al., 2011).

The baseline system achieves a BLEU score of 28.16 on the test set. Each goodness score alone brought small improvements in the BLEU score (systems 1–4 in Table 2), the best being sentence perplexity (+0.4 BLEU). An interesting property of our approach is that the individual gains seem to add up when we use several goodness scores, for instance combining recency and sentence perplexity gives and improvement by 0.63 BLEU (system 8) while the individual improvements are only +0.19 and 0.40 respectively. Combining corpus weights and sentence perplexity is less useful, as expected, since sentence perplexity implicitly weights the corpora. This is in fact an improved corpus weighting with a finer granularity. Our best system was obtained when combining corpus weights, recency and sentence perplexity weighting (system 11). For this system only, we numerically optimized the weights w_c, α and γ on the development set (see figure 1). The default and new weights are:

$$
\begin{aligned}
w_{eparl} &= 0.47714 \rightarrow 0.32823 \\
w_{nc} &= 0.52285 \rightarrow 0.67121 \\
\alpha &= 0.01300 \rightarrow 0.02102 \\
\beta &= 0.14530 \rightarrow 0.12901 \\
\gamma_{as} &= 0.1 \rightarrow 0.01289 \\
\gamma_{td} &= 0.1 \rightarrow 0.19201 \\
\gamma_{ppl} &= 0.1 \rightarrow 0.15451
\end{aligned}
$$

where γ_x is the coefficient among alignment score (*as*), temporal distance(*td*) and perplexity(*ppl*). By these means, we get an overall improvement of roughly +1 BLEU score (28.16 $\rightarrow$ 29.11) on test set. It is important to stress that this system is trained on exactly the same data than the baseline system and that the phrase table contains the same phrase-pairs. Our approach only improves the forward and backward probability estimates $P(\tilde{t}|\tilde{s})$ and $P(\tilde{s}|\tilde{t})$.

3.2 Experiments on the IWLST task

We performed the same type of experiments for the IWSLT task. The parallel training data was the in-domain *TED* corpus, the news-commentary corpus (*nc*) and a subset of the French–English 10^9 Gigaword (internally called *ccb2*). The results for this task are summarize in Table 3. the official Dev and test sets of the 2011 IWSLT talk task are used. Initial experiments have shown that large parts of the *ccb2* corpus are not relevant for this task (looking at the sentence perplexities). Therefore, we decided to only use a subset of this corpus, namely all the sentences with a perplexity lower than 70. This process preserve only 3% of the *ccb2* data. The baseline system trained on this data achieves a BLEU score of 26.34 on the test data. Using all the data in *ccb2* worsens significantly the results: the BLEU scores is 25.73. In principle, it is not necessary to select subsets of the parallel training data with our approach to weight sentences by perplexity, but this speeds up processing since we do not need to consider many sentences pairs with a very low weight. We perform a kind of pruning: all those sentences get a zero weight and are discarded right away. We used the LM build on the in-domain *TED* to calculate the sentence perplexities and the LM interpolation weights are used as corpus weights. The recency score was not used for this task since the test set of the TED corpus has no time information.

We observed the same behavior as for the WMT task: each individual goodness score improves the BLEU score on the test set (systems 1–3), weighting by sentence perplexity being the best one (+0.43 BLEU). The best system is obtained when combining all three goodness scores, leading a BLEU score of 26.91 (system 8). Again, the numerical optimization of the weights of the feature functions achieves an additional small improvement, giving an overall improvement of 0.73 BLEU. The weight of the goodness scores are shown in table 4.

4 Comparative Analysis

It is interesting to compare the impact of the different goodness scores considered. An interesting fact is that the trend is similar for both tasks.

By comparing the results obtained with the various systems, we can observe that the **corpus weights**, used alone or in combination with other features, are always beneficial (by pairwise comparison of *e.g.* systems 2 and 5, systems 3 and 6 or systems 4 and 7 from WMT task). The average gain provided by such weighting is around 0.2. Those weights correspond to the LM interpolation coefficient optimized to minimize the perplexity on the development set. They are useful to weight a whole corpus and to ensure that the in-domain corpus will globally receive higher weight than the other corpora.

Sentence level perplexity is also always useful (compare *e.g.* systems 1 and 7 or systems 6 and 11 from WMT task). While one could think that this goodness scores is redundant with corpus weight, it does bring additional information about the relevance of the sentence. This can be explained by the fact that a globally out-of-domain corpus can contain a fraction of useful sentences while, on the contrary, an in-domain corpus may contain some less useful ones. This is part of the heterogeneous aspect of any corpus. The average gain of using sentence perplexity is almost 0.3 for the WMT task and 0.37 for IWSLT task.

Concerning the **alignment score**, the results obtained are more mitigated (see *e.g.* the comparison between systems 2 and 5 on WMT and systems 2 and 4 from IWSLT task). The average gain is very low, and it is the only goodness score which sometimes decrease the BLEU score. The **temporal distance** has the expected behavior. When comparing systems 1 and 6, 3 and 8 or 4 and 11 from WMT Task, we can observe that an improvement of more than 0.2 is obtained.

5 Conclusion and future work

We have proposed a general framework to improve the phrase translation probabilities in a phrase-based SMT system. For this, we use a set of "goodness scores" at the corpus or sentence level. These scores are used to calculate forward and backward phrase translations probabilities which

IWSLT Task	Corpus weights	Alignment scores	perplexity	BLEU (on test)
Baseline				26.34
System 1	yes			26.61
System 2		yes		26.41
System 3			yes	26.77
System 4	yes	yes		26.51
System 5	yes		yes	**26.86**
System 6	yes		yes	**26.99 (optimized)**
System 7		yes	yes	26.81
System 8	yes	yes	yes	**26.91**
System 9	yes	yes	yes	**27.07 (optimized)**

Table 3: BLEU scores obtained with systems trained with different goodness scores on IWSLT Task.

IWSLT Task	c_{ted}	c_{nc}	$c_{ccb2.px70}$	β	γ_{ppl}	γ_{as}
Default values	0.74032	0.17378	0.08591	0.1	0.1	0.1
Optimized	0.69192	0.16982	0.13831	0.19251	0.18151	0.03118

Table 4: Weights on IWSLT Task (ppl=perlexity, as=alignment score).

are better adapted to the task. Our framework and implementation is generic and does not depend on the exact calculation of the corpus weights or the sentences goodness scores. Any value that expresses the appropriateness of the corpus and sentence with respect to the task can be used. The adapted system has exactly the same time and space complexity as the baseline system since we do not modify the number of entries in the phrase-table or add additional features. Also, the training time is only slightly increased.

We evaluated this approach on two well-known tasks: the 2011 WMT and IWSLT English/French evaluations. We have investigated several goodness scores: weights for corpora coming from different sources and weights at the sentence level based on the quality of the GIZA++ alignments, the recency with respect to the test set period and task appropriateness measured by the perplexity with respect to an in-domain language model. Using each one of these goodness scores, improved the BLEU score with respect to a strong baseline. However, best results were obtained by using all the goodness scores. This yielded an overall improvement of almost 1 point BLEU for the WMT task and more than 0.7 BLEU on the IWSLT task.

Future work will concentrate on other goodness scores. It would be interesting to compare the results with proposed goodness scores by integrating them directly into log-linear model as feature functions.

Acknowledgments

This work has been partially funded by the European Commission under the project Euromatrix-Plus, the French government under the project Cosmat and by an Overseas scholarship of Higher Education Pakistan. We would like to thank the unknown reviewers for their valuable comments.

References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Frank Vanden Berghen and Hugues Bersini. 2005. CONDOR, a new parallel, constrained extension of Powell's UOBYQA algorithm: Experimental results and comparison with the DFO algorithm. *Journal of Computational and Applied Mathematics*, 181:157–175, September.

Ondrej Bojar and Aleš Tamchyna. 2011. Improving translation model by monolingual data.

In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 330–336, Edinburgh, Scotland, July. Association for Computational Linguistics.

Boxing Chen, Min Zhang, Aiti Aw, and Haizhou Li. 2008. Exploiting n-best hypotheses for SMT self- enhancement. In *Association for Computational Linguistics*, pages 157–160.

Jorge Civera and Alfons Juan. 2007. Domain adaptation in statistical machine translation with mixture modelling. In *Second Workshop on SMT*, pages 177–180.

George Foster and Roland Kuhn. 2007. Mixture-model adaptation for SMT. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 128–135. 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*, EMNLP '10, pages 451–459, Stroudsburg, PA, USA. Association for Computational Linguistics.

Christian Hardmeier. 2010. Fast and extensible phrase scoring for statistical machine translation. *The Prague Bulletin of Mathematical Linguistics*, pages 87–96.

Daniel Hardt and Jakob Elming. 2010. Incremental re-training for post-editing smt. In *The Ninth Conference of the Association for Machine Translation in the Americas 2010*.

Fei Huang and Bing Xiang. 2010. Feature-rich discriminative phrase rescoring for smt. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 492–500, Stroudsburg, PA, USA. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on ' Human Language Technology - Volume 1*, NAACL '03, pages 48–54, Stroudsburg, PA, 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 *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Patrik Lambert, Holger Schwenk, Christophe Servan, and Sadaf Abdul-Rauf. 2011. Investigations on translation model adaptation using monolingual data. In *Sixth Workshop on SMT*.

Abby Levenberg, Chris Callison-Burch, and Miles Osborne. 2010. Stream-based translation models for statistical machine translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 394–402, Stroudsburg, PA, USA. Association for Computational Linguistics.

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*, pages 708–717.

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.

Marcello Federico Nicola Bertoldi. 2009. Domain adaptation for statistical machine transla-

tion. In *Forth Workshop on SMT*, pages 182–189.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 160–167, Stroudsburg, PA, USA. Association for Computational Linguistics.

Aaron B. Phillips and Ralf D. Brown. 2011. Training machine translation with a second-order taylor approximation of weighted translation instances. In *Machine Translation Summit XIII*.

Holger Schwenk. 2008. Investigations on large-scale lightly-supervised training for statistical machine translation. In *IWSLT*, pages 182–189.

Kashif Shah, Loïc Barrault, and Holger Schwenk. 2010. Translation model adaptation by resampling. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, WMT '10, pages 392–399, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kashif Shah, Loïc Barrault, and Holger Schwenk. 2011. Parametric weighting of parallel data for statistical machine translation. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1323–1331, Chiang Mai, Thailand, November. Asian Federation of Natural Language Processing.

Nicola Ueffing. 2006. Using monolingual source language data to improve MT performance. In *IWSLT*, pages 174–181.

Nicola Ueffing. 2007. Transductive learning for statistical machine translation. In *Association for Computational Linguistics*, pages 25–32.

Bing Zhao, Matthias Ech, and Stephen Vogal. 2004a. Language model adaptation for statistical machine translation with structured query models. In *Proceedings of the 20th international conference on Computational Linguistics*. Association for Computational Linguistics.

Bing Zhao, Stephan Vogel, Matthias Eck, and Alex Waibel. 2004b. Phrase pair rescoring with term weighting for statistical machine translatio. In *EMNLP*, pages 206–213.