Translation Model Adaptation by Resampling

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

Abstract

The translation model of statistical machine translation systems is trained on parallel data coming from various sources and domains. These corpora are usually concatenated, word alignments are calculated and phrases are extracted. This means that the corpora are not weighted according to their importance to the domain of the translation task. This is in contrast to the training of the language model for which well known techniques are used to weight the various sources of texts. On a smaller granularity, the automatic calculated word alignments differ in quality. This is usually not considered when extracting phrases either.

In this paper we propose a method to automatically weight the different corpora and alignments. This is achieved with a resampling technique. We report experimental results for a small (IWSLT) and large (NIST) Arabic/English translation tasks. In both cases, significant improvements in the BLEU score were observed.

1 Introduction

Two types of resources are needed to train statistical machine translation (SMT) systems: parallel corpora to train the translation model and monolingual texts in the target language to build the language model. The performance of both models depends of course on the quality and quantity of the available resources.

Today, most SMT systems are generic, *i.e.* the same system is used to translate texts of all kinds. Therefore, it is the domain of the training resources that influences the translations that are selected among several choices. While monolingual texts are in general easily available in many domains, the freely available parallel texts mainly come from international organisations, like the European Union or the United Nations. These texts, written in particular jargon, are usually much larger than in-domain bitexts. As an example we can cite the development of an NIST Arabic/English phrase-based translation system. The current NIST test sets are composed of a news wire part and a second part of web-style texts. For both domains, there is only a small number of in-domain bitexts available, in comparison to almost 200 millions words of out-of-domain UN texts. The later corpus is therefore likely to dominate the estimation of the probability distributions of the translation model.

It is common practice to use a mixture language model with coefficients that are optimized on the development data, *i.e.* by these means on the domain of the translation task. Domain adaptation seems to be more tricky for the translation model and it seems that very little research has been done that seeks to apply similar ideas to the translation model. To the best of our knowledge, there is no commonly accepted method to weight the bitexts coming from different sources so that the translation model is best optimized to the domain of the task. Mixture models are possible when only two different bitexts are available, but are rarely used for more corpora (see discussion in the next section).

In this work we propose a new method to adapt the translation model of an SMT system. We only perform experiments with phrase-based systems, but the method is generic and could be easily applied to an hierarchical or syntax-based system. We first associate a weighting coefficient to each bitext. The main idea is to use resampling to produce a new collection of weighted alignment files, followed by the standard procedure to extract the phrases. In a second step, we also consider the

Proceedings of the Joint 5th Workshop on Statistical Machine Translation and MetricsMATR, pages 392–399,
Uppsala, Sweden, 15-16 July 2010.

alignment score of each parallel sentence pair, emphasizing by these means good alignments and down-weighting less reliable ones. All the parameters of our procedure are automatically tuned by optimizing the BLEU score on the development data.

The paper is organized as follows. The next section describes related work on weighting the corpora and model adaptation. Section 3 describes the architecture allowing to resample and to weight the bitexts. Experimental results are presented in section 4 and the paper concludes with a discussion.

2 Related Work

Adaptation of SMT systems is a topic of increasing interest since few years. In previous work, adaptation is done by using mixture models, by exploiting comparable corpora and by self-enhancement of translation models.

Mixture models were used to optimize the coefficients to the adaptation domain. (Civera and Juan, 2007) proposed a model that can be used to generate topic-dependent alignments by extension of the HMM alignment model and derivation of Viterbi alignments. (Zhao et al., 2004) constructed specific language models by using machine translation output as queries to extract similar sentences from large monolingual corpora. (Foster and Kuhn, 2007) applied a mixture model approach to adapt the system to a new domain by using weights that depend on text distances to mixture components. The training corpus was divided into different components, a model was trained on each part and then weighted appropriately for the given context. (Koehn and Schroeder, 2007) used two language models and two translation models: one in-domain and other out-of-domain to adapt the system. Two decoding paths were used to translate the text.

Comparable corpora are exploited to find additional parallel texts. Information retrieval techniques are used to identify candidate sentences (Hildebrand et al., 2005). (Snover et al., 2008) used cross-lingual information retrieval to find texts in the target language that are related to the domain of the source texts.

A self-enhancing approach was applied by (Ueffing, 2006) to filter the translations of the test set with the help of a confidence score and to use reliable alignments to train an additional phrase table. This additional table was used with the existing generic phrase table. (Ueffing, 2007) further refined this approach by using transductive semi-supervised methods for effective use of monolingual data from the source text. (Chen et al., 2008) performed domain adaptation simultaneously for the translation, language and reordering model by learning posterior knowledge from N-best hypothesis. A related approach was investigated in (Schwenk, 2008) and (Schwenk and Senellart, 2009) in which lightly supervised training was used. An SMT system was used to translate large collections of monolingual texts, which were then filtered and added to the training data.

(Matsoukas et al., 2009) propose to weight each sentence in the training bitext by optimizing a discriminative function on a given tuning set. Sentence level features were extracted to estimate the weights that are relevant to the given task. Then certain parts of the training bitexts were down-weighted to optimize an objective function on the development data. This can lead to parameter over-fitting if the function that maps sentence features to weights is complex.

The technique proposed in this paper is somehow related to the above approach of weighting the texts. Our method does not require an explicit specification of the in-domain and out-of-domain training data. The weights of the corpora are directly optimized on the development data using a numerical method, similar to the techniques used in the standard minimum error training of the weights of the feature functions in the log-linear criterion. All the alignments of the bitexts are resampled and given equal chance to be selected and therefore, influence the translation model in a different way. Our proposed technique does not require the calculation of extra sentence level features, however, it may use the alignments score associated with each aligned sentence pair as a confidence score.

3 Description of the algorithm

The architecture of the algorithm is summarized in figure 1. The starting point is an (arbitrary) number of parallel corpora. We first concatenate these bitexts and perform word alignments in both directions using GIZA++. This is done on the concatenated bitexts since GIZA++ may perform badly if some of the individual bitexts are rather small. Next, the alignments are separated in parts corre-

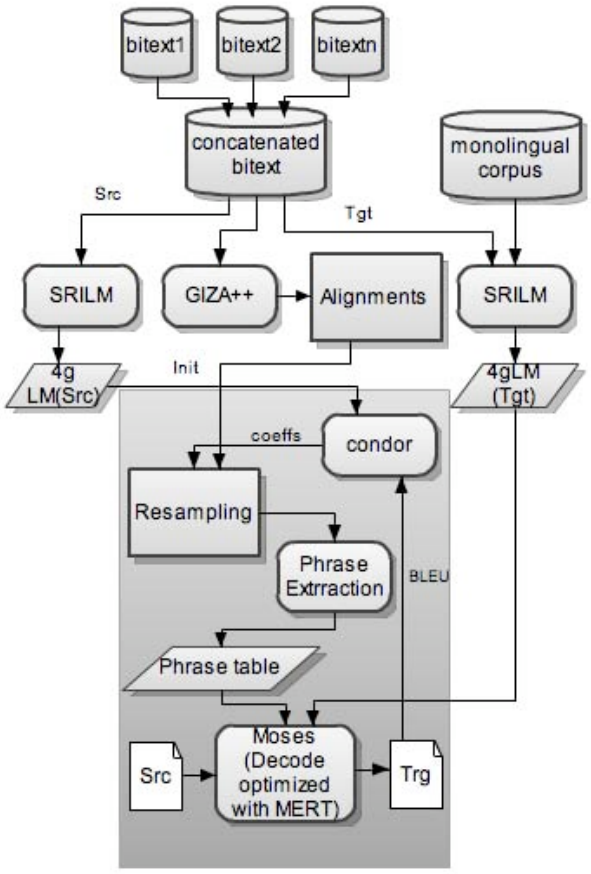

Figure 1: Architecture of SMT Weighting System

sponding to the individual bitexts and a weighting coefficient is associated to each one. We are not aware of a procedure to calculate these coefficients in an easy and fast way without building an actual SMT system. Note that there is an EM procedure to do this for language modeling.

In the next section, we will experimentally compare equal coefficients, coefficients set to the same values than those obtained when building an interpolated language model on the source language, and a new method to determine the coefficients by optimizing the BLEU score on the development data.

One could imagine to directly use these coefficients when calculating the various probabilities of the extracted phrases. In this work, we propose a different procedure that makes no assumptions on how the phrases are extracted and probabilities are calculated. The idea is to *resample alignments* from the alignment file corresponding to the individual bitexts according to their weighting coefficients. By these means, we create a new, potentially larger alignment file, which then in turn will be used by the standard phrase extraction procedure.

3.1 Resampling the alignments

In statistics, resampling is based upon repeated sampling within the same sample until a sample is obtained which better represents a given data set (Yu, 2003). Resampling is used for validating models on given data set by using random subsets. It overcomes the limitations to make assumptions about the distribution of the data. Usually resampling is done several times to better estimate and select the samples which better represents the target data set. The more often we resample, the closer we get to the true probability distribution.

In our case we performed resampling with replacement according to the following algorithm:

Algorithm 1 *Resampling*

1: **for** $i = 0$ to required size **do**
2: Select any alignment randomly
3: $Al_{score} \leftarrow$ normalized alignment score
4: $Threshold \leftarrow$ rand[0, 1]
5: **if** $Al_{score} > Threshold$ **then**
6: keep it
7: **end if**
8: **end for**

Let us call resampling factor, the number of times resampling should be done. An interesting question is to determine the optimal value of this resampling factor.

It actually depends upon the task or data we are experimenting on. We may start with one time resampling and could stop when results becomes stable. Figure 2 plots a typical curve of the BLEU score as a function of the number of times we resample. It can be observed that the curve is growing proportionally to the resampling factor until it becomes stable after a certain point.

3.2 Weighting Schemes

We concentrated on translation model adaptation when the bitexts are heterogeneous, *e.g.* in-domain and out-of-domain or of different sizes. In this case, weighting these bitexts seems interesting and can be used in order to select data which better represent the target domain. Secondly when sentences are aligned, some alignments are reliable and some are less. Using unreliable alignments can put negative effect on the translation quality. So we need to exclude or down-weight

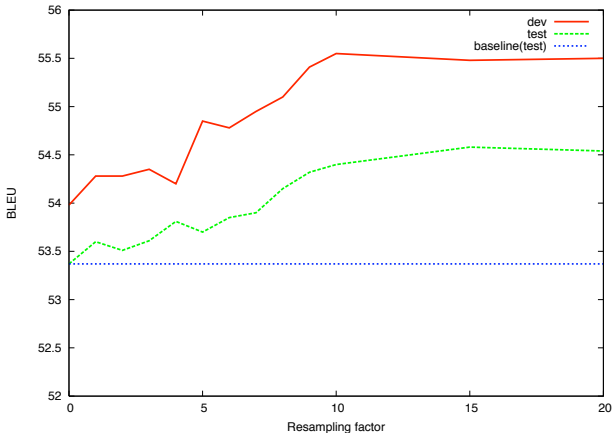

Figure 2: The curve shows that by increasing the resampling factor we get better and stable results on Dev and Test.

unreliable alignments and keep or up-weight the good ones. We conceptually divided the weighting in two parts that is (i) weighting the corpora and (ii) weighting the alignments

3.2.1 Weighting Corpora

We started to resample the bitexts with equal weights to see the effect of resampling. This gives equal importance to each bitext without taking into account the domain of the text to be translated. However, it should be better to give appropriate weights according to a given domain as shown in equation 1

$$\alpha_1 bitext_1 + \alpha_2 bitext_2 + .. + \alpha_n bitext_n \quad (1)$$

where the α_n are the coefficients to optimize. One important question is how to find out the appropriate coefficient for each corpus. We investigated a technique similar to the algorithm used to minimize the perplexity of an interpolated target LM. Alternatively, it is also possible to construct a interpolated language model on the source side of bitexts. This approach was implemented and these coefficients were used as the weights for each bitext. One can certainly ask the question whether the perplexity is a good criterion for weighting bitexts. Therefore, we worked on direct optimization of these coefficients by CONDOR (Berghen and Bersini, 2005). This freely available tool is a numerical optimizer based on Powell's UOBYQA algorithm (Powell, 1994). The aim of CONDOR is to minimize a objective function using the least number of function evaluations. Formally, it is used to find $x^* \in R^n$ with given constraints which satisfies

$$F(x^*) = \min_x F(x) \quad (2)$$

where n is the dimension of search space and x^* is the optimum of x. The following algorithm was used to weight the bitexts.

Algorithm 2 $WeightingCorpora$

1: Determine word to word alignment with GIZA++ on concatenated bitext.
2: **while** Not converged **do**
3: Run Condor initialized with LM weights.
4: Create new alignment file by resampling according to weights given by Condor.
5: Use the alignment file to extract phrases and build the translation table (phrase table)
6: Tune the system with MERT (this step can be skipped until weights are optimized to save time)
7: Calculate the BLEU score
8: **end while**

3.2.2 Weighting Alignments

Alignments produced by GIZA++ have alignment scores associated with each sentence pair in both direction, *i.e.* source to target and target to source. We used these alignment scores as confidence measurement for each sentence pair. Alignment scores depend upon the length of each sentence, therefore, they must be normalized regarding the size of the sentence. Alignment scores have a very large dynamic range and we have applied a logarithmic mapping in order to flatten the probability distribution :

$$\log(\lambda \cdot \frac{(\sqrt[n_{trg}]{a_{src\_trg}} + \sqrt[n_{src}]{a_{trg\_src}})}{2}) \quad (3)$$

where a is the alignment score, n the size of a sentence and λ a coefficient to optimize. This is also done by Condor.

Of course, some alignments will appear several times, but this will increase the probability of certain phrase-pairs which are supposed to be more related to the target domain. We have observed that the weights of an interpolated LM build on the source side of the bitext are good initial values for CONDOR. Moreover, weights optimized by Condor are in the same order than these "LM weights". Therefore, we do not perform MERT of the SMT systems build at each step of the optimization of the weights α_i and λ by CONDOR,

	IWSLT Task		NIST Task	
	Dev (Dev6)	Test (Dev7)	Dev (NIST06)	Test (NIST08)
Baseline	53.98	53.37	43.16	42.21
With equal weights	53.71	53.20	43.10	42.11
With LM weights	54.20	53.71	43.42	42.22
Condor weights	54.80	53.98	43.49	42.28

Table 1: BLEU scores when weighting corpora (one time resampling)

	IWSLT Task		NIST Task	
	Dev (Dev6)	Test (Dev7)	Dev (NIST06)	Test (NIST08)
Baseline	53.98	53.37	43.16	42.21
With equal weights	53.80	53.30	43.13	42.15
With LM weights	54.32	53.91	43.54	42.37
Condor weights	55.10	54.13	43.80	42.40

Table 2: BLEU scores when weighting corpora (optimum number of resampling)

	IWSLT Task			NIST Task		
	Dev (Dev6)	Test (Dev7)	TER(Test)	Dev (NIST06)	Test (NIST08)	TER(Test)
Baseline	53.98	53.37	32.75	43.16	42.21	51.69
With equal weights	53.85	53.33	32.80	43.28	42.21	51.72
With LM weights	54.80	54.10	31.50	43.42	42.41	51.50
Condor weights	55.48	54.58	31.31	43.95	42.54	51.35

Table 3: BLEU and TER scores when weighting corpora and alignments (optimum number of resampling)

but use the values obtained by running MERT on a system obtained by using the "LM weights" to weight the alignments. Once CONDOR has converged to optimal weights, we can then tune our system by MERT. This saves lot of time taken by the tuning process and it had no impact on the results.

4 Experimental evaluation

The baseline system is a standard phrase-based SMT system based on the Moses SMT toolkit (Koehn and et al., 2007). In our system we used fourteen features functions. These features functions include phrase and lexical translation probabilities in both directions, seven features for lexicalized distortion model, a word and phrase penalty, and a target language model. The MERT tool is used to tune the coefficients of these feature functions. We considered Arabic to English translation. Tokenization of the Arabic source texts is done by a tool provided by SYSTRAN which also performs a morphological decomposition. We considered two well known official evaluation tasks to evaluate our approach, namely NIST and IWSLT.

For IWSLT, we used the BTEC bitexts (194M words), Dev1, Dev2, Dev3 (60M words each) as training data, Dev6 as development set and Dev7 as test set. From previous experiments, we have evidence that the various development corpora are not equally important and weighting them correctly should improve the SMT system. We analyze the translation quality as measured by the BLEU score for the three methods: equal weights, LM weights and Condor weights and considering one time resampling. Further experiments were performed using the optimized number of resampling with and without weighting the alignments. We have realized that it is beneficial to always include the original alignments. Even if we resample many times there is a chance that some alignments might never be selected but we do not want to loose any information. By keeping original alignments, all alignments are given a chance to be se-

lected at least once. All these results are summarized in tables 1, 2 and 3.

One time resampling along with equal weights gave worse results than the baseline system while improvements in the BLEU score were observed with LM and Condor weights for the IWSLT task, as shown in table 1. Resampling many times always gave more stable results, as already shown in figure 2 and as theoretically expected. For this task, we resampled 15 times. The improvements in the BLEU score are shown in table 2. Furthermore, using the alignment scores resulted in additional improvements in the BLEU score. For the IWSLT task, we achieved and overall improvement of 1.5 BLEU points on the development set and 1.2 BLEU points on the test set as shown in table 3

To validate our approach we further experimented with the NIST evaluation task. Most of the training data used in our experiments for the NIST task is made available through the LDC. The bitexts consist of texts from the GALE project[1] (1.6M words), various news wire translations[2] (8.0M words) on development data from previous years (1.6M words), LDC treebank data (0.4M words) and the ISI extracted bitexts (43.7M words). The official NIST06 evaluation data was used as development set and the NIST08 evaluation data was used as test set. The same procedure was adapted for the NIST task as for the IWSLT task. Results are shown in table 1 by using different weights and one time resampling. Further improvements in the results are shown in table 2 with the optimum number of resampling which is 10 for this task. Finally, results by weighting alignments along with weighting corpora are shown in table 3. Our final system achieved an improvement of 0.79 BLEU points on the development set and 0.33 BLEU points on the test set. TER scores are also shown on test set of our final system in table 3. Note that these results are state-of-the-art when compared to the official results of the 2008 NIST evaluation[3].

The weights of the different corpora are shown in table 4 for the IWSLT and NIST task. In both cases, the weights optimized by CONDOR are substantially different form those obtained when creating an interpolated LM on the source side of the bitexts. In any case, the weights are clearly non uniform, showing that our algorithm has focused on in-domain data. This can be nicely seen for the NIST task. The Gale texts were explictely created to contain in-domain news wire and WEB texts and actually get a high weight despite their small size, in comparison to the more general news wire collection from LDC.

5 Conclusion and future work

We have proposed a new technique to adapt the translation model by resampling the alignments, giving a weight to each corpus and using the alignment score as confidence measurement of each aligned phrase pair. Our technique does not change the phrase pairs that are extracted,[4] but only the corresponding probability distributions. By these means we hope to adapt the translation model in order to increase the weight of translations that are important to the task, and to downweight the phrase pairs which result from unreliable alignments.

We experimentally verified the new method on the low-resource IWSLT and the resource-rich NIST'08 tasks. We observed significant improvement on both tasks over state-of-the-art baseline systems. This weighting scheme is generic and it can be applied to any language pair and target domain. We made no assumptions on how the phrases are extracted and it should be possible to apply the same technique to other SMT systems which rely on word-to-word alignments.

On the other hand, our method is computationally expensive since the optimisation of the coefficients requires the creation of a new phrase table and the evaluation of the resulting system in the tuning loop. Note however, that we run GIZA++ only once.

In future work, we will try to directly use the weights of the corpora and the alignments in the algorithm that extracts the phrase pairs and calculates their probabilities. This would answer the interesting question whether resampling itself is needed or whether weighting the corpora and alignments is the key to the observed improvements in the BLEU score.

Finally, it is straight forward to consider more feature functions when resampling the alignments. This may be a way to integrate linguistic knowl-

[1] LDC2005E83, 2006E24, E34, E85 and E92

[2] LDC2003T07, 2004E72, T17, T18, 2005E46 and 2006E25.

[3] http://www.nist.gov/speech/tests/mt/2008/

[4] when also including the original alignments

IWSLT Task	BTEC	Dev1	Dev2	Dev3
# of Words	194K	60K	60K	60K
LM Coeffs	0.7233	0.1030	0.0743	0.0994
Condor Coeffs	0.6572	0.1058	0.1118	0.1253

NIST TASK	Gale	NewsWire	TreeBank	Dev	ISI
# of words	1.6M	8.1M	0.4M	1.7M	43.7M
LM Coeffs	0.3215	0.1634	0.0323	0.1102	0.3726
Condor Coeffs	0.4278	0.1053	0.0489	0.1763	0.2417

Table 4: Weights of the different bitexts.

edge into the SMT system, *e.g.* giving low scores to word alignments that are *"grammatically not reasonable"*.

Acknowledgments

This work has been partially funded by the European Commission under the project Euromatrix and by the Higher Education Commission(HEC) Pakistan as Overseas scholarship. We are very thankful to SYSTRAN who provided support for the Arabic tokenization.

References

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.

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.

Almut Silja Hildebrand, Matthias Eck, Stephan Vogel, and Alex Waibel. 2005. Adaptation of the translation model for statistical machine translation based on information retrieval. In *EAMT*, pages 133–142.

Philipp Koehn and et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Association for Computational Linguistics, demonstration session.*, pages 224–227.

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.

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.

M.J.D. Powell. 1994. A direct search optimization method that models the objective and constraint functions by linar interpolation. In *In Advances in Optimization and Numerical Analysis, Proceedings of the sixth Workshop on Optimization and Numerical Analysis, Oaxaca, Mexico, volume 275*, pages 51–67. Kluwer Academic Publishers.

Holger Schwenk and Jean Senellart. 2009. Translation model adaptation for an Arabic/French news translation system by lightly-supervised training. In *MT Summit*.

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

Matthew Snover, Bonnie Dorr, and Richard Schwartz. 2008. Language and translation

model adaptation using comparalble corpora. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 857–866.

Nicola Ueffing. 2006. Using monolingual sourcce 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.

Chong Ho Yu. 2003. Resampling methods: Concepts, applications, and justification. In *Practical Assessment Research and Evaluation*.

Bing Zhao, Matthias Ech, and Stephen Vogal. 2004. 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.