Domain Adaptation for Statistical Machine Translation with Monolingual Resources

Nicola Bertoldi **Marcello Federico**
FBK-irst - Ricerca Scientifica e Tecnologica
Via Sommarive 18, Povo (TN), Italy
{bertoldi, federico}@fbk.eu

Abstract

Domain adaptation has recently gained interest in statistical machine translation to cope with the performance drop observed when testing conditions deviate from training conditions. The basic idea is that in-domain training data can be exploited to adapt all components of an already developed system. Previous work showed small performance gains by adapting from limited in-domain bilingual data. Here, we aim instead at significant performance gains by exploiting large but cheap monolingual in-domain data, either in the source or in the target language. We propose to synthesize a bilingual corpus by translating the monolingual adaptation data into the counterpart language. Investigations were conducted on a state-of-the-art phrase-based system trained on the Spanish–English part of the UN corpus, and adapted on the corresponding Europarl data. Translation, re-ordering, and language models were estimated after translating in-domain texts with the baseline. By optimizing the interpolation of these models on a development set the BLEU score was improved from 22.60% to 28.10% on a test set.

1 Introduction

A well-known problem of Statistical Machine Translation (SMT) is that performance quickly degrades as soon as testing conditions deviate from training conditions. The very simple reason is that the underlying statistical models always tend to closely approximate the empirical distributions of the training data, which typically consist of bilingual texts and monolingual target-language texts. The former provide a means to learn likely translations pairs, the latter to form correct sentences with translated words. Besides the general difficulties of language translation, which we do not consider here, there are two aspects that make machine learning of this task particularly hard. First, human language has intrinsically very sparse statistics at the surface level, hence gaining complete knowledge on translation phrase pairs or target language n-grams is almost impractical. Second, language is highly variable with respect to several dimensions, style, genre, domain, topics, etc. Even apparently small differences in domain might result in significant deviations in the underlying statistical models. While data sparseness corroborates the need of large language samples in SMT, linguistic variability would indeed suggest to consider many alternative data sources as well. By rephrasing a famous saying we could say that "no data is better than more *and assorted* data".

The availability of language resources for SMT has dramatically increased over the last decade, at least for a subset of relevant languages and especially for what concerns monolingual corpora. Unfortunately, the increase in quantity has not gone in parallel with an increase in assortment, especially for what concerns the most valuable resource, that is bilingual corpora. Large parallel data available to the research community are for the moment limited to texts produced by international organizations (European Parliament, United Nations, Canadian Hansard), press agencies, and technical manuals.

The limited availability of parallel data poses challenging questions regarding the portability of SMT across different application domains and language pairs, and its adaptability with respect to language variability within the same application domain.

This work focused on the second issue, namely the adaptation of a Spanish-to-English phrase-based SMT system across two apparently close domains: the United Nation corpus and the Euro-

Proceedings of the 4th EACL Workshop on Statistical Machine Translation , pages 182–189,
Athens, Greece, 30 March – 31 March 2009.

pean Parliament corpus. Cross-domain adaptation is faced under the assumption that only monolingual texts are available, either in the source language or in the target language.

The paper is organized as follows. Section 2 presents previous work on the problem of adaptation in SMT; Section 3 introduces the exemplar task and research questions we addressed; Section 4 describes the SMT system and the adaptation techniques that were investigated; Section 5 presents and discusses experimental results; and Section 6 provides conclusions.

2 Previous Work

Domain adaptation in SMT has been investigated only recently. In (Eck et al., 2004) adaptation is limited to the target language model (LM). The background LM is combined with one estimated on documents retrieved from the WEB by using the input sentence as query and applying cross-language information retrieval techniques. Refinements of this approach are described in (Zhao et al., 2004).

In (Hildebrand et al., 2005) information retrieval techniques are applied to retrieve sentence pairs from the training corpus that are relevant to the test sentences. Both the language and the translation models are retrained on the extracted data.

In (Foster and Kuhn, 2007) two basic settings are investigated: cross-domain adaptation, in which a small sample of parallel in-domain text is assumed, and dynamic adaptation, in which only the current input source text is considered. Adaptation relies on mixture models estimated on the training data through some unsupervised clustering method. Given available adaptation data, mixture weights are re-estimated ad-hoc. A variation of this approach was also recently proposed in (Finch and Sumita, 2008). In (Civera and Juan, 2007) mixture models are instead employed to adapt a word alignment model to in-domain parallel data.

In (Koehn and Schroeder, 2007) cross-domain adaptation techniques were applied on a phrase-based SMT trained on the Europarl task, in order to translate news commentaries, from French to English. In particular, a small portion of in-domain bilingual data was exploited to adapt the Europarl language model and translation models by means of linear interpolation techniques. Ueffing et al. (2007) proposed several elaborate adaptation methods relying on additional bilingual data synthesized from the development or test set.

Our work is mostly related to (Koehn and Schroeder, 2007) but explores different assumptions about available adaptation data: i.e. only monolingual in-domain texts are available. The adaptation of the translation and re-ordering models is performed by generating synthetic bilingual data from monolingual texts, similarly to what proposed in (Schwenk, 2008). Interpolation of multiple phrase tables is applied in a more principled way than in (Koehn and Schroeder, 2007): all entries are merged into one single table, corresponding feature functions are concatenated and smoothing is applied when observations are missing. The approach proposed in this paper has many similarities with the simplest technique in (Ueffing et al., 2007), but it is applied to a much larger monolingual corpus.

Finally, with respect to previous work we also investigate the behavior of the minimum error training procedure to optimize the combination of feature functions on a small in-domain bilingual sample.

3 Task description

This paper addresses the issue of adapting an already developed phrase-based translation system in order to work properly on a different domain, for which almost no parallel data are available but only monolingual texts.[1]

The main components of the SMT system are the translation model, which aims at porting the content from the source to the target language, and the language model, which aims at building fluent sentences in the target language. While the former is trained with bilingual data, the latter just needs monolingual target texts. In this work, a lexicalized re-ordering model is also exploited to control re-ordering of target words. This model is also learnable from parallel data.

Assuming some large monolingual in-domain texts are available, two basic adaptation approaches are pursued here: (i) generating synthetic bilingual data with an available SMT system and use this data to adapt its translation and re-ordering models; (ii) using synthetic or provided target texts to also, or only, adapt its language model. The following research questions

[1] We assume only availability of a development set and an evaluation set.

summarize our basic interest in this work:

- Is automatic generation of bilingual data effective to tackle the lack of parallel data?
- Is it more effective to use source language adaptation data or target language adaptation data?
- Is it convenient to combine models learned from adaptation data with models learned from training data?
- How can interpolation of models be effectively learned from small amounts of in-domain parallel data?

4 System description

The investigation presented in this paper was carried out with the Moses toolkit (Koehn et al., 2007), a state-of-the-art open-source phrase-based SMT system. We trained Moses in a standard configuration, including a 4-feature translation model, a 7-feature lexicalized re-ordering model, one LM, word and phrase penalties.

The translation and the re-ordering model relied on "grow-diag-final" symmetrized word-to-word alignments built using GIZA++ (Och and Ney, 2003) and the training script of Moses. A 5-gram language model was trained on the target side of the training parallel corpus using the IRSTLM toolkit (Federico et al., 2008), exploiting Modified Kneser-Ney smoothing, and quantizing both probabilities and backoff weights. Decoding was performed applying cube-pruning with a pop-limit of 6000 hypotheses.

Log-linear interpolations of feature functions were estimated with the parallel version of minimum error rate training procedure distributed with Moses.

4.1 Fast Training from Synthetic Data

The standard procedure of Moses for the estimation of the translation and re-ordering models from a bilingual corpus consists in three main steps:

1. A word-to-word alignment is generated with GIZA++.
2. Phrase pairs are extracted from the word-to-word alignment using the method proposed by (Och and Ney, 2003); countings and re-ordering statistics of all pairs are stored. A word-to-word lexicon is built as well.
3. Frequency-based and lexicon-based direct and inverted probabilities, and re-ordering probabilities are computed using statistics from step 2.

Recently, we enhanced Moses decoder to also output the word-to-word alignment between the input sentence and its translation, given that they have been added to the phrase table at training time. Notice that the additional information introduces an overhead in disk usage of about 70%, but practically no overhead at decoding time. However, when training translation and re-ordering models from synthetic data generated by the decoder, this feature allows to completely skip the time-expensive step 1.[2]

We tested the efficiency of this solution for training a translation model on a synthesized corpus of about 300K Spanish sentences and 8.8M running words, extracted from the EuroParl corpus. With respect to the standard procedure, the total training time was reduced by almost 50%, phrase extraction produced 10% more phrase pairs, and the final translation system showed a loss in translation performance (BLEU score) below 1% relative. Given this outcome we decided to apply the faster procedure in all experiments.

4.2 Model combination

Once monolingual adaptation data is automatically translated, we can use the synthetic parallel corpus to estimate new language, translation, and re-ordering models. Such models can either replace or be combined with the original models of the SMT system. There is another simple option which is to concatenate the synthetic parallel data with the original training data and re-build the system. We did not investigate this approach because it does not allow to properly balance the contribution of different data sources, and also showed to underperform in preliminary work.

Concerning the combination of models, in the following we explain how Moses was extended to manage multiple translation models (TMs) and multiple re-ordering models (RMs).

4.3 Using multiple models in Moses

In Moses, a TM is provided as a phrase table, which is a set $\mathcal{S} = \{(\tilde{f}, \tilde{e})\}$ of phrase pairs associated with a given number of features values

[2] Authors are aware of an enhanced version of GIZA++, which allows parallel computation, but it was not taken into account in this work.

$h(\tilde{f}, \tilde{e}; \mathcal{S})$. In our configuration, 5 features for the TM (the phrase penalty is included) are taken into account.

In the first phase of the decoding process, Moses generates translation options for all possible input phrases $\tilde{f}$ through a lookup into $\mathcal{S}$; it simply extracts alternative phrase pairs $(\tilde{f}, \tilde{e})$ for a specific $\tilde{f}$ and optionally applies pruning (based on the feature values and weights) to limit the number of such pairs. In the second phase of decoding, it creates translation hypotheses of the full input sentence by combining in all possible ways (satisfying given re-ordering constraints) the prefetched translation options. In this phase the hypotheses are scored, according to all features functions, ranked, and possibly pruned.

When more TMs $\mathcal{S}_j$ are available, Moses can behave in two different ways in pre-fetching the translation options. It searches a given $\tilde{f}$ in all sets and keeps a phrase pair $(\tilde{f}, \tilde{e})$ if it belongs to either i) their intersection or ii) their union. The former method corresponds to building one new TM $\mathcal{S}_I$, whose set is the intersection of all given sets:

$$\mathcal{S}_I = \{(\tilde{f}, \tilde{e}) \mid \forall j \ (\tilde{f}, \tilde{e}) \in \mathcal{S}_j\}$$

The set of features of the new TM is the union of the features of all single TMs. Straightforwardly, all feature values are well-defined.

The second method corresponds to building one new TM $\mathcal{S}_U$, whose set is the union of all given sets:

$$\mathcal{S}_U = \{(\tilde{f}, \tilde{e}) \mid \exists j \ (\tilde{f}, \tilde{e}) \in \mathcal{S}_j\}$$

Again, the set of features of the new TM is the union of the features of all single TMs; but for a phrase pair $(\tilde{f}, \tilde{e})$ belonging to $\mathcal{S}_U \setminus \mathcal{S}_j$, the feature values $h(\tilde{f}, \tilde{e}; \mathcal{S}_j)$ are undefined. In these undefined situations, Moses provides a default value of 0, which is the highest available score, as the feature values come from probabilistic distributions and are expressed as logarithms. Henceforth, a phrase pair belonging to all original sets is penalized with respect to phrase pairs belonging to few of them only.

To address this drawback, we proposed a new method[3] to compute a more reliable and smoothed score in the undefined case, based on the IBM model 1 (Brown et al., 1993). If $(\tilde{f} = f_1, \ldots, f_l, \tilde{e} = e_1, \ldots, e_l) \in \mathcal{S}_U \setminus \mathcal{S}_j$ for any j the phrase-based and lexical-based direct features are defined as follows:

$$h(\tilde{f}, \tilde{e}; \mathcal{S}_j) = \frac{\epsilon}{(l+1)^m} \prod_{k=1}^{m} \sum_{h=0}^{l} \phi(e_k \mid f_h)$$

Here, $\phi(e_k \mid f_h)$ is the probability of e_k given f_h provided by the word-to-word lexicon computed on $\mathcal{S}_j$. The inverted features are defined similarly. The phrase penalty is trivially set to 1. The same approach has been applied to build the union of re-ordering models. In this case, however, the smoothing value is constant and set to 0.001.

As concerns as the use of multiple LMs, Moses has a very easy policy, consisting of querying each of them to get the likelihood of a translation hypotheses, and uses all these scores as features.

It is worth noting that the exploitation of multiple models increases the number of features of the whole system, because each model adds its set of features. Furthermore, the first approach of Moses for model combination shrinks the size of the phrase table, while the second one enlarges it.

5 Evaluation

5.1 Data Description

In this work, the background domain is given by the Spanish-English portion of the UN parallel corpus,[4] composed by documents coming from the Office of Conference Services at the UN in New York spanning the period between 1988 and 1993. The adaptation data come from the European Parliament corpus (Koehn, 2002) (EP) as provided for the shared translation task of the 2008 Workshop on Statistical Machine Translation.[5] Development and test sets for this task, namely dev2006 and test2008, are supplied as well, and belong to the European Parliament domain.

We use the symbol $\bar{\text{S}}$ ($\bar{\text{E}}$) to denote synthetic Spanish (English) data. Spanish-to-English and English-to-Spanish systems trained on UN data were exploited to generate English and Spanish synthetic portions of the original EP corpus, respectively. In this way, we created two synthetic versions of the EP corpus, named S$\bar{\text{E}}$-EP and $\bar{\text{S}}$E-EP, respectively. All presented translation systems were optimized on the dev2006 set with respect to

[3] Authors are not aware of any work addressing this issue.

[4] Distributed by the Linguistic Data Consortium, catalogue # LDC94T4A.

[5] http://www.statmt.org/wmt08

the BLEU score (Papineni et al., 2002), and tested on test2008. (Notice that one reference translation is available for both sets.) Table 1 reports statistics of original and synthetic parallel corpora, as well of the employed development and evaluation data sets. All the texts were just tokenized and mixed case was kept. Hence, all systems were developed to produce case-sensitive translations.

corpus	sent	Spanish		English	
		word	dict	word	dict
UN	2.5M	50.5M	253K	45.2M	224K
EP	1.3M	36.4M	164K	35.0M	109K
SĒ-EP	1.3M	36.4M	164K	35.4M	133K
S̄E-EP	1.3M	36.2M	120K	35.0M	109K
dev	2,000	60,438	8,173	58,653	6,548
test	2,000	61,756	8,331	60,058	6,497

Table 1: Statistics of bilingual training corpora, development and test data (after tokenization).

5.2 Baseline systems

Three Spanish-to-English baseline systems were trained by exploiting different parallel or monolingual corpora summarized in the first three lines in Table 2. For each system, the table reports the perplexity and out-of-vocabulary (OOV) percentage of their LM, and its translation performance achieved on the test set in terms of BLEU score, NIST score, WER (word error rate) and PER (position independent error rate).

The distance in style, genre, jargon, etc. between the UN and the EP corpora is made evident by the gap in perplexity (Federico and De Mori, 1998) and OOV percentage between their English LMs: 286 vs 74 and 1.12% vs 0.15%, respectively.

Performance of the system trained on the EP corpus (third row) can be taken as an upper bound for any adaptation strategy trying to exploit parts of the EP corpus, while those of the first line clearly provide the corresponding lower-bound. The system in the second row can instead be consider as the lower bound when only monolingual English adaptation data are assumed.

The synthesis of the SĒ-EP corpus was performed with the system trained just on the UN training data (first row of Table 2), because we had assumed that the in-domain data were only monolingual Spanish and thus not useful for neither the TM, RM nor target LM estimation.

Similarly, the system in the last row of Table 2 was developed on the UN corpus to translate the English part of the EP data to generate the synthetic S̄E-EP corpus. Again, any in-domain data were exploited to train this sytem. Of course, this system cannot be compared with any other because of the different translation direction.

In order to compare reported performance with the state-of-the-art, Table 2 also reports results of the best system published in the EuroMatrix project website[6] and of the Google online translation engine.[7]

5.3 Analysis of the tuning process

It is well-known that tuning the SMT system is fundamental to achieve good performance. The standard tuning procedure consists of a minimum error rate training (mert) (Och and Ney, 2003) which relies on the availability of a development data set. On the other hand, the most important assumption we make is that almost no parallel in-domain data are available.

conf	sent	n-best	time (min)	BLEU (Δ)
–	–	–	–	22.28
a	2000	1000	2034	23.68 (1.40)
b	2000	200	391	23.67 (1.39)
c	200	1000	866	23.13 (0.85)
d	200	200	551	23.54 (1.26)

Table 3: Global time, not including decoding, of the tuning process and BLEU score achieved on the test set by the uniform interpolation weights (first row), and by the optimal weights with different configurations of the tuning parameters.

In a preliminary phase, we investigated different settings of the tuning process in order to understand how much development data is required to perform a reliable weight optimization. Our models were trained on the SĒ-EP parallel corpus and by using uniform interpolation weights the system achieved a BLEU score of 22.28% on the test set (see Table 3).

We assumed to dispose of either a regular in-domain development set of 2,000 sentences (dev2006), or a small portion of it of just 200 sen-

[6] http://www.euromatrix.net. Translations of the best system were downloaded on November 7th, 2008. Published results differ because we performed a case-sensitive evaluation.

[7] Google was queried on November 3rd, 2008.

language pair	training data		PP	OOV (%)	BLEU	NIST	WER	PER
	TM/RM	LM						
Spanish-English	UN	UN	286	1.12	22.60	6.51	64.60	48.52
”	UN	EP	74	0.15	27.83	7.12	60.93	45.19
”	EP	EP	”	”	32.80	7.84	56.47	41.15
”	UN	SĒ-EP	89	0.21	23.52	6.64	63.86	47.68
”	SĒ-EP	SĒ-EP	”	”	23.68	6.65	63.64	47.56
”	S̄E-EP	EP	74	0.15	28.10	7.18	60.86	44.85
”	Google		na	na	28.60	7.55	57.38	57.38
”	Euromatrix		na	na	32.99	7.86	56.36	41.12
English-Spanish	UN	UN	281	1.39	23.24	6.44	65.81	49.61

Table 2: Description and performance on the test set of compared systems in terms of perplexity, out-of-vocabulary percentage of their language model, and four translation scores: BLEU, NIST, word-error-rate, and position-independent error rate. Systems were optimized on the dev2006 development set.

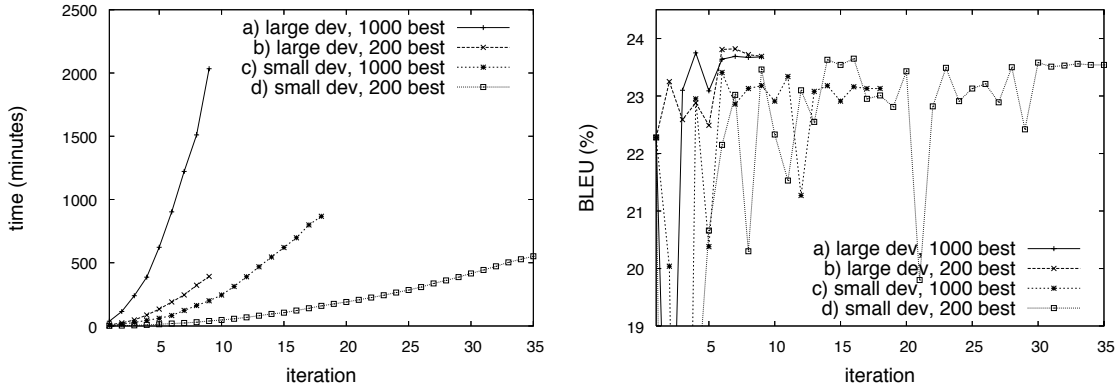

Figure 1: Incremental time of the tuning process (not including decoding phase) (left) and BLEU score on the test set using weights produced at each iteration of the tuning process. Four different configurations of the tuning parameters are considered.

tences. Moreover, we tried to employ either 1,000-best or 200-best translation candidates during the mert process.

From a theoretical point of view, computational effort of the tuning process is proportional to the square of the number of translation alternatives generated at each iteration times the number of iterations until convergence.

Figure 1 reports incremental tuning time and translation performance on the test set at each iteration. Notice that the four tuning configurations are ranked in order of complexity. Table 3 summaries the final performance of each tuning process, after convergence was reached.

Notice that decoding time is not included in this plot, as Moses allows to perform this step in parallel on a computer cluster. Hence, to our view the real bottleneck of the tuning process is actually related to the strictly serial part of the mert implementation of Moses.

As already observed in previous literature (Macherey et al., 2008), first iterations of the tuning process produces very bad weights (even close to 0); this exceptional performance drop is attributed to an over-fitting on the candidate repository.

Configurations exploiting the small development set (c,d) show a slower and more unstable convergence; however, their final performance in Table 3 result only slightly lower than that obtained with the standard dev sets (a, b). Due to the larger number of iterations they needed, both configurations are indeed more time consuming than the intermediate configuration (b), which seems the best one. In conclusion, we found that the size of the n-best list has essentially no effect on the quality of the final weights, but it impacts significantly on the computational time. Moreover, using the regular development set with few translation alternatives ends up to be the most efficient

configuration in terms of computational effort, robustness, and performance.

Our analysis suggests that it is important to dispose of a sufficiently large development set although reasonably good weights can be obtained even if such data are very few.

5.4 LM adaptation

A set of experiments was devoted to the adaptation of the LM only. We trained three different LMs on increasing portions of the EP and we employed them either alone or in combination with the background LM trained on the UN corpus.

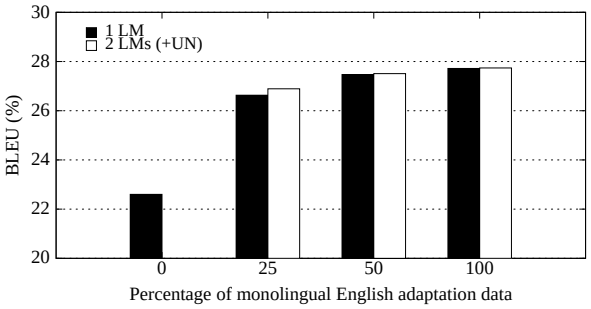

Figure 2: BLEU scores achieved by systems exploiting one or two LMs trained on increasing percentages of English in-domain data.

Figure 2 reports BLEU score achieved by these systems. The absolute gain with respect to the baseline is fairly high, even with the smallest amount of adaptation data (+4.02). The benefit of using the background data together with in-domain data is very small, and rapidly vanishes as the amount of such data increases.

If English synthetic texts are employed to adapt the LM component, the increase in performance is significantly lower but still remarkable (see Table 2). By employing all the available data, the gain in BLEU% score was of 4% relative, that is from 22.60 to 23.52.

5.5 TM and RM adaptation

Another set of experiments relates to the adaptation of the TM and the RM. In-domain TMs and RMs were estimated on three different versions of the full parallel EP corpus, namely EP, SĒ-EP, and S̄E-EP. In-domain LMs were trained on the corresponding English side. All in-domain models were either used alone or combined with the baseline models according to multiple-model paradigm explained in Section 4.3. Tuning of the interpolation weights was performed on the standard development set as usual. Results of these experiments are reported in Figure 3.

Results suggest that regardless of the used bilingual corpora the in-domain TMs and RMs work better alone than combined with the original models. We think that this behavior can be explained by a limited disciminative power of the resulting combined model. The background translation model could contain phrases which either do or do not fit the adaptation domain. As the weights are optimized to balance the contribution of all phrases, the system is not able to well separate the positive examples from the negative ones. In addition to it, system tuning is much more complex because the number of features increases from 14 to 26.

Finally, TMs and RMs estimated from synthetic data show to provide smaller, but consistent, contributions than the corresponding LMs. When English in-domain data is provided, BLEU% score increases from 22.60 to 28.10; TM and RM contribute by about 5% relative, by covering the gap from 27.83 to 28.10. When Spanish in-domain data is provided BLEU% score increases from 22.60 to 23.68; TM and RM contribute by about 15% relative, by covering the gap from 23.52 to 23.68 .

Summarizing, the most important role in the domain adaptation is played by the LM; nevertheless the adaptation of the TM and RM gives a small further improvement..

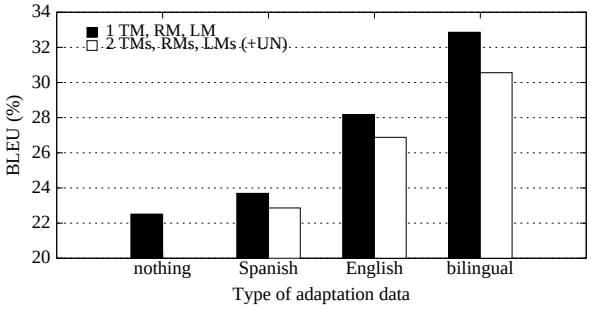

Figure 3: BLEU scores achieved by system exploiting both TM, RM and LM trained on different corpora.

6 Conclusion

This paper investigated cross-domain adaptation of a state-of-the-art SMT system (Moses), by exploiting large but cheap monolingual data. We proposed to generate synthetic parallel data by

translating monolingual adaptation data with a background system and to train statistical models from the synthetic corpus.

We found that the largest gain (25% relative) is achieved when in-domain data are available for the target language. A smaller performance improvement is still observed (5% relative) if source adaptation data are available. We also observed that the most important role is played by the LM adaptation, while the adaptation of the TM and RM gives consistent but small improvement.

We also showed that a very tiny development set of only 200 parallel sentences is adequate enough to get comparable performance as a 2000-sentence set.

Finally, we described how to reduce the time for training models from a synthetic corpus generated through Moses by 50% at least, by exploiting word-alignment information provided during decoding.

References

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–312.

Jorge Civera and Alfons Juan. 2007. Domain adaptation in statistical machine translation with mixture modelling. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 177–180, Prague, Czech Republic.

Matthias Eck, Stephan Vogel, and Alex Waibel. 2004. Language model adaptation for statistical machine translation based on information retrieval. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pages 327–330, Lisbon, Portugal.

Marcello Federico and Renato De Mori. 1998. Language modelling. In Renato De Mori, editor, *Spoken Dialogues with Computers*, chapter 7, pages 199–230. Academy Press, London, UK.

Marcello Federico, Nicola Bertoldi, and Mauro Cettolo. 2008. Irstlm: an open source toolkit for handling large scale language models. In *Proceedings of Interspeech*, pages 1618–1621, Melbourne, Australia.

Andrew Finch and Eiichiro Sumita. 2008. Dynamic model interpolation for statistical machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 208–215, Columbus, Ohio.

George Foster and Roland Kuhn. 2007. Mixture-model adaptation for SMT. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 128–135, Prague, Czech Republic.

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

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, Prague, Czech Republic.

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

Philipp Koehn. 2002. Europarl: A multilingual corpus for evaluation of machine translation. Unpublished, `http://www.isi.edu/~koehn/europarl/`.

Wolfgang Macherey, Franz Och, Ignacio Thayer, and Jakob Uszkoreit. 2008. Lattice-based minimum error rate training for statistical machine translation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 725–734, Honolulu, Hawaii.

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 *Proceedings of the 40th Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 311–318, Philadelphia, PA.

Holger Schwenk. 2008. Investigations on Large-Scale Lightly-Supervised Training for Statistical Machine Translation. In *Proc. of the International Workshop on Spoken Language Translation*, pages 182–189, Hawaii, USA.

Nicola Ueffing, Gholamreza Haffari, and Anoop Sarkar. 2007. Semi-supervised model adaptation for statistical machine translation. *Machine Translation*, 21(2):77–94.

Bing Zhao, Matthias Eck, and Stephan Vogel. 2004. Language model adaptation for statistical machine translation via structured query models. In *Proceedings of Coling 2004*, pages 411–417, Geneva, Switzerland.