Semi-supervised model adaptation for statistical machine translation

Nicola Ueffing
Interactive Language Technologies Group, National Research Council Canada, Gatineau, QC, Canada (nicola.ueffing@gmail.com)

Gholamreza Haffari
School of Computing Science, Simon Fraser University, Burnaby, BC, Canada (ghaffar1@cs.sfu.ca)

Anoop Sarkar
School of Computing Science, Simon Fraser University, Burnaby, BC, Canada (anoop@cs.sfu.ca)

Abstract. Statistical machine translation systems are usually trained on large amounts of bilingual text (used to learn a translation model), and also large amounts of monolingual text in the target language (used to train a language model). In this article we explore the use of semi-supervised model adaptation methods for the effective use of monolingual data from the source language in order to improve translation quality. We propose several algorithms with this aim, and present the strengths and weaknesses of each one. We present detailed experimental evaluations on the French–English EuroParl data set and on data from the NIST Chinese–English large-data track. We show a significant improvement in translation quality on both tasks.

Keywords: statistical machine translation, self-training, semi-supervised learning, domain adaptation, model adaptation

1. Introduction

In statistical machine translation (SMT), translation is modeled as a decision process. The goal is to find the translation $\mathbf{t}$ of source sentence $\mathbf{s}$ which maximizes the posterior probability:

$$\arg\max_{\mathbf{t}} p(\mathbf{t} \mid \mathbf{s}) = \arg\max_{\mathbf{t}} p(\mathbf{s} \mid \mathbf{t}) \cdot p(\mathbf{t}) \tag{1}$$

This decomposition of the probability yields two different statistical models which can be trained independently of each other: the translation model $p(\mathbf{s} \mid \mathbf{t})$ and the target language model $p(\mathbf{t})$.

State-of-the-art SMT systems are trained on large collections of text which consist of bilingual corpora, to learn the parameters of the translation model, $p(\mathbf{s} \mid \mathbf{t})$, and of monolingual target language corpora, for the target language model, $p(\mathbf{t})$. It has been shown that adding large amounts of target language text improves translation quality considerably, as improved language model estimates about potential output translations can be used by the decoder in order to improve translation quality (Brants et al., 2007). However, the availability of monolingual corpora in the source language has not been shown to help improve the system's performance. In this article, we aim to show how such corpora can be used to achieve higher translation quality.

Even if large amounts of bilingual text are given, the training of the statistical models usually suffers from sparse data. The number of possible events, eg. word pairs or phrase pairs or pairs of subtrees in the two languages, is too big to reliably estimate a probability distribution over such pairs. Another problem is that for many language pairs the amount of available bilingual text is very limited. In this work, we will address this problem and propose a general framework to solve it. Our hypothesis is that adding information from source language text can also provide improvements. Unlike adding target language text, this hypothesis is a natural semi-supervised learning problem. To tackle this problem, we propose algorithms for semi-supervised model adaptation. We translate sentences from the development set or test set and use the generated translations to improve the performance of the SMT system. In this article, we show that such an approach can lead to better translations despite the fact that the development and test data are typically much smaller in size than typical training data for SMT systems.

The proposed semi-supervised learning can be seen as a means to adapt the SMT system to a new type of text. Say a system trained on newswire is used to translate weblog texts. The proposed method adapts the trained models to the style and domain of the new input without requiring bilingual data from this domain.

2. Baseline MT System

The SMT system we applied in our experiments is PORTAGE. This is a state-of-the-art phrase-based translation system developed by the National Research Council Canada which has been made available to Canadian universities for research and education purposes. We provide a basic description here; for a detailed description see (Ueffing et al., 2007).

The models (or features) which are employed by the decoder are:

- one or several phrase table(s), which model the translation direction $p(\mathbf{s}\,|\,\mathbf{t})$. They are smoothed using the methods described in (Foster et al., 2006),

- one or several n-gram language model(s) trained with the SRILM toolkit (Stolcke, 2002); in the experiments reported here, we used three different 4-gram models on the NIST data, and a trigram model on EuroParl,

- a distortion model which assigns a penalty based on the number of source words which are skipped when generating a new target phrase,

- a word penalty assigning a constant cost to each generated target word.

These different models are combined log-linearly. Their weights are optimized w.r.t. BLEU score using the algorithm described in (Och, 2003). This is done on a development corpus which we will call dev1 in this article.

The search algorithm implemented in the decoder is a dynamic-programming beam-search algorithm. After the main decoding step, rescoring with additional models is performed. The system generates a 5,000-best list of alternative translations for each source sentence. These lists are rescored with the following models:

- the different models used in the decoder which are described above,

- two different features based on IBM Model 1 (Brown et al., 1993): a Model 1 probability calculated over the whole sentence, and a feature estimating the number of source words which have a reliable translation. Both features are determined for both translation directions,

- posterior probabilities for words, phrases, n-grams, and sentence length (Zens and Ney, 2006; Ueffing and Ney, 2007), all calculated over the N-best list and using the sentence probabilities which the baseline system assigns to the translation hypotheses.

The weights of these additional models and of the decoder models are again optimized to maximize BLEU score. This is performed on a second development corpus, which we call dev2 in this article.

Algorithm 1 Model adaptation algorithm for statistical machine translation

```
Input: training set $L$ of parallel sentence pairs.
// Bilingual training data.
Input: unlabeled set $U$ of source text.
// Monolingual source language data.
Input: number of iterations $R$, and size of $N$-best list.
$T_{-1} := \{\}$.    // Additional bilingual training data.
$i := 0$.    // Iteration counter.
repeat
    Training step: $\pi^{(i)} := \textbf{Estimate}(L, T_{i-1})$.
    $X_i := \{\}$.    // The set of generated translations for this iteration.
    for sentence $\mathbf{s} \in U$ do
        Labeling step: Decode $\mathbf{s}$ using $\pi^{(i)}$ to obtain $N$ best target sentences $(t_n)_{n=1}^N$ with their scores
        $X_i := X_i \cup \{(\mathbf{t}_n, \mathbf{s}, \pi^{(i)}(\mathbf{t}_n \,|\, \mathbf{s}))_{n=1}^N\}$
    end for
    Scoring step: $S_i := \textbf{Score}(X_i)$
    // Assign a score to sentence pairs $(\mathbf{t}, \mathbf{s})$ from $X_i$.
    Selection step: $T_i := \textbf{Select}(X_i, S_i)$
    // Choose a subset of good sentence pairs $(\mathbf{t}, \mathbf{s})$ from $X_i$.
    $i := i + 1$.
until $i > R$
```

3. The Framework

3.1. The Algorithm

Our model adaptation algorithm, Algorithm 1, is inspired by the Yarowsky algorithm (Yarowsky, 1995; Abney, 2004). We will describe it here for (re-)training of the translation model. However, the same algorithm can be used to (re-)train other SMT models, such as the language model as we will show later.

The algorithm works as follows: First, the translation model $\pi^{(i)}$ is estimated based on the sentence pairs in the bilingual training data L. Then a set of source language sentences, U, is translated based on the current model. A subset of good translations and their sources, T_i, is selected in each iteration and added to the training data. These selected sentence pairs are replaced in each iteration, and only the original bilingual training data, L, is kept fixed throughout the algorithm. The process of generating sentence pairs, selecting a subset of good sentence pairs, and updating the model is continued until the stopping condition is met. Note that the set of sentences U is drawn from a development set or the test set that will be used eventually to evaluate the SMT system. However, the evaluation step is still done just once at the end of our learning process and all optimization steps are carried out on development data.

In Algorithm 1, changing the definition of **Estimate**, **Score** and **Select** will give us the different semi-supervised learning algorithms we will discuss in this article. We will present experimental results for applying semi-supervised adaptation to the phrase translation model, the language model, and both of them at the same time.

Given the probability model $p(\mathbf{t} \mid \mathbf{s})$, consider the distribution over all possible translations $\mathbf{t}$ for a particular input sentence $\mathbf{s}$. We can initialize this probability distribution to the uniform distribution for each sentence $\mathbf{s}$ in the unlabeled data U. Thus, this distribution over translations of sentences from U will have the maximum entropy. Under certain precise conditions, as described in (Abney, 2004), we can analyze Algorithm 1 as minimizing the entropy of the distribution over translations of U. However, this is true only when the functions **Estimate**, **Score** and **Select** have very prescribed definitions. In this article, rather than analyzing the convergence of Algorithm 1 we run it for a fixed number of iterations and instead focus on finding useful definitions for **Estimate**, **Score** and **Select** that can be experimentally shown to improve MT performance.

3.2. The Estimate Function

We consider the following different definitions for **Estimate** in Algorithm 1:

Full Re-training (of the model): If **Estimate**(L, T) estimates the model parameters based on $L \cup T$, then we have a semi-supervised algorithm that re-trains a model on the original training data L plus the sentences decoded in the last iteration.

Additional Model: On the other hand, a new model can be learned on T alone and then this model is added as a new component in the log-linear SMT model. This is an attractive alternative as the full re-training of the model on labeled and unlabeled data is computationally expensive if L is very large (as on the Chinese–English data set). This additional model is small and specific to the development or test set it is trained on. As the analysis of such an additional phrase table in Section 4.2.2 shows, it overlaps with the original phrase tables, but also contains many new phrase pairs.

Mixture Model: Another alternative for **Estimate** is to create a mixture model of the original model probabilities with the newly trained one. In the case of the phrase translation model, this yields

$$p(\mathbf{s}\,|\,\mathbf{t}) = \lambda \cdot p_L(\mathbf{s}\,|\,\mathbf{t}) + (1-\lambda) \cdot p_T(\mathbf{s}\,|\,\mathbf{t}) \qquad (2)$$

where p_L and p_T are phrase table probabilities estimated on L and T, respectively. In cases where new phrase pairs are learned from T, they get added into the merged phrase table.

3.3. The Scoring Function

In Algorithm 1, the **Score** function assigns a score to each translation hypothesis $\mathbf{t}$. We used the following scoring functions in our experiments:

Length-normalized Score: Each translated sentence pair $(\mathbf{t}, \mathbf{s})$ is scored according to the model probability $p(\mathbf{t}\,|\,\mathbf{s})$ (assigned by the SMT system) normalized by the length $|\mathbf{t}|$ of the target sentence:

$$\mathbf{Score}(\mathbf{t}, \mathbf{s}) \;=\; p(\mathbf{t}\,|\,\mathbf{s})^{\frac{1}{|\mathbf{t}|}} \qquad (3)$$

Confidence Estimation: The goal of confidence estimation is to estimate how reliable a translation $\mathbf{t}$ is, given the corresponding source sentence $\mathbf{s}$. The confidence estimation which we implemented follows the approaches suggested in (Blatz et al., 2003; Ueffing and Ney, 2007): The confidence score of a target sentence $\mathbf{t}$ is calculated as a log-linear combination of several different sentence scores. These scores are Levenshtein-based word posterior probabilities, phrase posterior probabilities, and a target language model score. The posterior probabilities are determined over the N-best list generated by the SMT system.

The word posterior probabilities are calculated on basis of the Levenshtein alignment between the hypothesis under consideration and all other translations contained in the N-best list. The Levenshtein alignment is performed between a given hypothesis $\mathbf{t}$ and every sentence $\mathbf{t}_n$ contained in the N-best list individually. To calculate the posterior probability of target word t occurring in position i of the translation, the probabilities of all sentences containing t in position i or in a position Levenshtein-aligned to i is summed up. This sum is then normalized by the total probability mass of the N-best list. Let $\mathcal{L}(\mathbf{t}, \mathbf{t}_n)$ be the Levenshtein alignment between sentences $\mathbf{t}$ and $\mathbf{t}_n$, and $\mathcal{L}_i(\mathbf{t}, \mathbf{t}_n)$ that of word t in position i in $\mathbf{t}$.

Consider the following example: Calculating the Levenshtein alignment between the sentences $\mathbf{t}$ =”A B C D E” and $\mathbf{t}_n$ =”B C G E F” yields

$$\mathcal{L}(\mathbf{t}, \mathbf{t}_n) = \text{“– B C G E”}$$

where ”–” represents insertion of the word A into $\mathbf{t}$ and in the above alignment F is deleted from $\mathbf{t}_n$. Using this representation, the word posterior probability of word t occurring in a position Levenshtein-aligned to i is given by

$$p_{\text{lev}}(t \,|\, \mathbf{s}, \mathbf{t}, \mathcal{L}) \;=\; \frac{\sum\limits_{n=1}^{N} \delta(t, \mathcal{L}_i(\mathbf{t}, \mathbf{t}_n)) \cdot p(\mathbf{s}, \mathbf{t}_n)}{\sum\limits_{n=1}^{N} p(\mathbf{s}, \mathbf{t}_n)} \tag{4}$$

To obtain a score for the whole target sentence, the posterior probabilities of all target words are multiplied. The sentence probability is approximated by the probability which the SMT system assigns to the sentence pair. More details on computing word posterior probabilities are available in (Ueffing and Ney, 2007).

The phrase posterior probabilities are determined in a similar manner by summing the sentence probabilities of all translation hypotheses in the N-best list which contain the phrase pair. The segmentation of the sentence into phrases is provided by the SMT system. Again, the single values are multiplied to obtain a score for the whole sentence.

The language model score is determined using a 5-gram model trained on the English Gigaword corpus for NIST. On French–English, we used the trigram model which was provided for the NAACL 2006 shared task.

The log-linear combination of the different sentence scores into one confidence score is optimized w.r.t. sentence classification error rate (CER) on the development corpus. The weights in this combination are optimized using the Downhill Simplex algorithm (Press et al., 2002). In order to carry out the optimization, reference classes are needed which label a given translation as either correct or incorrect. These are created by calculating the word error rate (WER) of each translation and labeling the sentence as incorrect if the WER exceeds a certain value, and correct otherwise. Then the confidence score $c(\mathbf{t})$ of translation $\mathbf{t}$ is computed, and the sentence is classified as correct or incorrect by comparing its confidence to a threshold τ:

$$c(\mathbf{t}) \begin{cases} > \tau & \Rightarrow \mathbf{t} \;\; \text{correct} \\ \leq \tau & \Rightarrow \mathbf{t} \;\; \text{incorrect} \end{cases}$$

The threshold τ is optimized to minimize CER.

We then compare the assigned classes to the reference classes, determine the CER and update the weights accordingly. This process is iterated until the CER converges.

3.4. The Selection Function

The **Select** function in Algorithm 1 is used to create the additional training data T_i which will be used in the next iteration $i+1$ by **Estimate** to augment the information from the original bilingual training data. This augmentation is accomplished in different ways, depending on the definition of the **Estimate** function. We use the following selection functions:

Importance Sampling: For each sentence $\mathbf{s}$ in the set of unlabeled sentences U, the Labeling step in Algorithm 1 generates an N-best list of translations, and the subsequent Scoring step assigns a score to each translation $\mathbf{t}$ in this list. The set of generated translations for all sentences in U is the event space and the scores are used to put a probability distribution over this space, simply by renormalizing the scores described in Section 3.3. We use importance sampling to select K translations from this distribution. Sampling is done with replacement which means that the same translation may be chosen several times. Furthermore, several different translations of the same source sentence can be sampled from the N-best list. The K sampled translations and their associated source sentences make up the additional training data T_i.

Selection using a Threshold: This method compares the score of each single-best translation to a threshold. The translation is considered reliable and added to the set T_i if its score exceeds the threshold. Else it is discarded and not used in the additional training data. The threshold is optimized on the development beforehand. Since the scores of the translations change in each iteration, the size of T_i also changes.

Keep All: This method does not perform any selection at all. It is simply assumed that all translations in the set X_i are reliable, and none of them are discarded. Thus, in each iteration, the result of the selection step will be $T_i = X_i$. This method was implemented mainly for comparison with other selection methods.

4. Experimental Results

4.1. Setting

We ran experiments on two different corpora: one is the French–English translation task from the EuroParl corpus, and the other one is Chinese–English translation as performed in the NIST MT evaluation[1].

[1] www.nist.gov/speech/tests/mt

Table I. French–English EuroParl corpora

corpus	use	sentences
EuroParl	phrase table + language model	688K
dev06	dev1	2,000
test06	test in-domain / out-of-domain	2,000 / 1,064

Table II. NIST Chinese–English corpora

corpus	use	sentences	domains
non-UN	phrase table + language model	3.2M	news, magazines, laws
UN	phrase table + language model	5.0M	UN Bulletin
English Gigaword	language model	11.7M	news
multi-p3	dev1	935	news
multi-p4	dev2	919	news
eval-04	test	1,788	newswire, editorials, political speeches
eval-06 GALE	test	2,276	broadcast conversations, broadcast news, newsgroups, newswire
eval-06 NIST	test	1,664	broadcast news, newsgroups, newswire

For the French–English translation task, we used the EuroParl corpus as distributed for the shared task in the NAACL 2006 workshop on statistical machine translation[2]. The corpus statistics are shown in Table I. The development set is used to optimize the model weights in the decoder, and the evaluation is done on the test set provided for the NAACL 2006 shared task. Note that this test set contains 2,000 in-domain sentences and 1,064 out-of-domain sentences collected from news commentary. We will carry out evaluation separately for these two domains to investigate the adaptation capabilities of our methods.

For the Chinese–English translation task, we used the corpora distributed for the large-data track in the 2006 NIST evaluation (see Table II). We used the LDC segmenter for Chinese. A subset of the English Gigaword corpus was used as additional language model training material. The multiple translation corpora multi-p3 and multi-p4 were

[2] www.statmt.org/wmt06/shared-task/

Table III. Feasibility of settings for Algorithm 1

estimate	select	score	EuroParl	NIST
full re-training			∗	†
mixture model			∗	∗
additional model	keep all		∗∗	∗
	importance sampling	norm. scores	∗∗	∗
		confidence	∗∗	∗
	threshold	norm. scores	∗∗	∗
		confidence	∗∗	∗

used as development corpora. Evaluation was performed on the 2004 and 2006 test sets. Note that the training data consist mainly of written text, whereas the test sets comprise three and four different genres: editorials, newswire and political speeches in the 2004 test set, and broadcast conversations, broadcast news, newsgroups and newswire in the 2006 test set. Most of these domains have characteristics which are different from those of the training data, e.g., broadcast conversations have characteristics of spontaneous speech, and the newsgroup data is comparatively unstructured.

Given the particular data sets described above, Table III shows the various options for the **Estimate**, **Score** and **Select** functions in Algorithm 1 (see Section 3). The table provides a quick guide to the experiments we present in this article vs. those we did not attempt due to computational infeasibility. We ran experiments corresponding to all entries marked with ∗ (see Section 4.2). For those marked ∗∗ the experiments produced only minimal improvement over the baseline and so we do not discuss them in this article. The entry marked as † (full re-training on the NIST data) was not attempted because this is not feasible. However, it was run on the smaller EuroParl corpus.

4.1.1. *Evaluation Metrics*

We evaluated the generated translations using three different evaluation metrics: BLEU score (Papineni et al., 2002), mWER (multi-reference word error rate), and mPER (multi-reference position-independent word error rate) (Nießen et al., 2000). Note that BLEU score measures translation quality, whereas mWER and mPER measure translation errors.

We will present 95%-confidence intervals for the baseline system which are calculated using bootstrap resampling. The metrics are calculated w.r.t. one and four English references: the EuroParl data comes

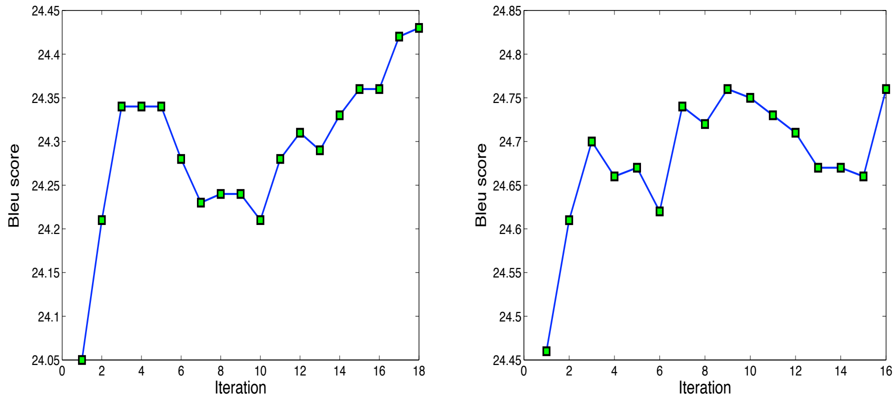

Figure 1. Translation quality for importance sampling with full re-training on train100k (left) and train150k (right). EuroParl French–English task.

with one reference, the NIST 2004 evaluation set and the NIST section of the 2006 evaluation set are provided with four references each, and the GALE section of the 2006 evaluation set comes with one reference only. This results in much lower BLEU scores and higher error rates for the translations of the GALE section in comparison with the NIST section (see Section 4.2). Note that these values do not indicate lower translation quality, but are simply a result of using only one reference.

4.2. Results

4.2.1. *EuroParl*

We ran our initial experiments on EuroParl to explore the behavior of the model adaptation algorithm. In all experiments reported in this subsection, the test set was used as unlabeled data. In one set of experiments, we reduced the size of the bilingual training data according to the similarity of its sentences to the test set. This was done by computing the similarity of each source sentence in the training set to the set of sentences in the test set, defined as the average similarity of the source sentence to the test set sentences in the set. The similarity between two sentences was measured based on the fraction of n-grams shared by them. We used the 100K and 150K training sentences filtered according to n-gram coverage over the test set. The selection and scoring was carried out using importance sampling with normalized scores. We fully re-trained the phrase tables on these data and 8,000 test sentence pairs sampled from 20-best lists in each iteration. The results on the test set can be seen in Figure 1. The BLEU score increases, although with slight variation, over the iterations. In total, it increases

Table IV. EuroParl results based on the mixture of phrase tables with $\lambda = 0.9$.

test set	selection method	BLEU[%]
French–English EuroParl (1 ref.)		
in-domain	baseline	28.1 ±0.8
	importance sampling	28.4
	keep all	28.3
out-of-domain	baseline	18.8 ±0.8
	importance sampling	18.9
	keep all	19.0

from 24.1 to 24.4 for the 100K filtered corpus, and from 24.5 to 24.8 for 150K, respectively. Moreover, we see that the BLEU score of the system using 100K training sentence pairs and model adaptation is the same as that of the one trained on 150K sentence pairs. So the information extracted from untranslated test sentences is equivalent to having an additional 50K sentence pairs.

In a second set of experiments, we used the whole EuroParl corpus and the sampled sentences for fully re-training the phrase tables in each iteration. We ran the algorithm for three iterations and the BLEU score increased from 25.3 to 25.6. Even though this is a small increase, it shows that the unlabeled data contains some information which can be exploited using model adaptation.

In a third experiment, we applied the mixture model idea as explained in Section 3.2. The initially learned phrase table was merged with the newly learned phrase table in each iteration with a weight of $\lambda = 0.9$. This value for λ was found based on cross validation on a development set. We ran the experiments on in-domain and out-of-domain sentences separately, and the results can be seen in Table IV. Again this is a small increase in the performance of the model, but it shows that the unlabeled data contains some information which can be explored in semi-supervised learning.

The main point of the EuroParl experiments described in this section was to explore the nature of full-retraining and multiple iterations in model adaptation. Our experiments show that full re-training can be problematic as the new phrase pairs from the test set are swamped by the training data. As we show, this effect can be alleviated by filtering the training data when doing full re-training, or by using a mixture model. The significance of the results in the EuroParl experiments is compromised by having only one reference translation.

Table V. NIST Chinese–English results with an additional phrase table trained on the dev/test set. Best results printed in boldface.

selection method	scoring method	BLEU[%]	mWER[%]	mPER[%]
eval-04 (4 refs.)				
baseline		31.8±0.7	66.8±0.7	41.5±0.5
keep all		33.1	66.0	41.3
importance sampling	norm. scores	**33.5**	65.8	40.9
	confidence	33.2	65.6	**40.4**
threshold	norm. scores	**33.5**	65.9	40.8
	confidence	**33.5**	**65.3**	40.8
eval-06 GALE (1 ref.)				
baseline		12.7±0.5	75.8±0.6	54.6±0.6
keep all		12.9	75.7	55.0
importance sampling	norm. scores	13.2	74.7	54.1
	confidence	12.9	74.4	53.5
threshold	norm. scores	12.7	75.2	54.2
	confidence	**13.6**	**73.4**	**53.2**
eval-06 NIST (4 refs.)				
baseline		27.9±0.7	67.2±0.6	44.0±0.5
keep all		28.1	66.5	44.2
importance sampling	norm. scores	28.7	66.1	43.6
	confidence	28.4	65.8	**43.2**
threshold	norm. scores	28.3	66.1	43.5
	confidence	**29.3**	**65.6**	**43.2**

The improvements achieved on the EuroParl corpus are slightly below the 95%-significance level. However, we observe them consistently in all settings. Furthermore, these experiments on the smaller EuroParl dataset provide us with insight on how to organize the experiments on the NIST large data track Chinese-English MT task.

4.2.2. *NIST*

Table V presents translation results on NIST with different versions of the scoring and selection methods introduced in Section 3. For each corpus U of unlabeled data (i.e. the development or test set), 5,000-best lists were generated using the baseline SMT system. Since re-training the full phrase tables is not feasible here, a (small) additional phrase table, specific to U, was trained and plugged into the SMT system as an

Table VI. NIST results with an additional phrase table and language model. Scoring: confidence estimation, selection: threshold. Best results printed in boldface.

test corpus	model	BLEU[%]	mWER[%]	mPER[%]
eval-04 (4 refs.)	baseline	31.8±0.7	66.8±0.7	41.5±0.5
	language model	33.1	65.9	41.1
	phrase table	**33.5**	**65.3**	40.8
	both	33.3	65.4	**40.6**
eval-06 GALE (1 ref.)	baseline	12.7±0.5	75.8±0.6	54.6±0.6
	language model	13.3	74.2	53.8
	phrase table	**13.6**	**73.4**	**53.2**
	both	13.3	74.2	53.4
eval-06 NIST (4 refs.)	baseline	27.9±0.7	67.2±0.6	44.0±0.5
	language model	28.6	66.0	43.8
	phrase table	**29.3**	**65.6**	**43.2**
	both	28.6	65.9	43.4

additional model. The decoder weights thus had to be optimized again to determine the appropriate weight for this new phrase table. This was done on the dev1 corpus, using the phrase table specific to dev1. Every time a new corpus is to be translated, an adapted phrase table is created using semi-supervised learning and used with the weight which has been learned on dev1. In the first experiment presented in Table V, all of the generated 1-best translations were kept and used for training the adapted phrase tables. This method (which was mainly tested as comparative method to assess the impact of the scoring and selection steps) yields slightly higher translation quality than the baseline system. The second approach we studied is the use of importance sampling over 20-best lists, based either on length-normalized sentence scores or confidence scores. As the results in Table V show, both variants outperform the first method, with a consistent improvement over the baseline across all test corpora and evaluation metrics. The third method uses a threshold-based selection method. Combined with confidence estimation as scoring method, this yields the best results. All improvements over the baseline are significant at the 95%-level.

We used this scoring and selection method to run experiments on semi-supervised language model adaptation on the NIST data. A unigram language model was trained on the translations selected by the

Table VII. NIST results with a mixture-model phrase table, $\lambda = 0.9$.
Scoring: confidence estimation, selection: threshold.

test corpus	model	BLEU[%]	mWER[%]	mPER[%]
eval-04 (4 refs.)	baseline	31.8±0.7	66.8±0.7	41.5±0.5
	mixture model	32.5	66.7	41.2
eval-06 GALE (1 ref.)	baseline	12.7±0.5	75.8±0.6	54.6±0.6
	mixture model	13.1	75.3	54.0
eval-06 NIST (4 refs.)	baseline	27.9±0.7	67.2±0.6	44.0±0.5
	mixture model	28.5	66.7	43.9

semi-supervised learning algorithm and plugged into the SMT system as additional model. Again, we obtained a small and very specific model for each development or test set. We also investigated learning higher-order n-gram language models on the selected data. We found that a unigram language model (or sometimes a bigram) yields best results. So the model adaptation seems to mainly boost the relevant vocabulary in the language model. The results of these experiments are shown in Table VI. The table compares the translation quality achieved by semi-supervised learning of a language model, a phrase translation model, and the combination of both. We see that semi-supervised language model adaptation yields a significant improvement in translation quality over the baseline on all three test sets. However, a higher improvement is obtained from adapting the phrase translation model. Unfortunately, the improvements achieved by adapting the two models do not add up. Adaptation of the phrase translation model only yields the best results.

Rather than using the phrase table learned on the selected data as additional model in the SMT system, we can also interpolate it with the original phrase table as shown in Equation 2 on page 6. This has the advantage that we do not have to re-optimize the system for learning a decoder weight for the additional model. However, as we see in Table VII, this method yields an improvement in translation quality which is much lower than the ones achieved in the other experiments reported earlier in this section. There is a small gain on all test corpora and according to all three evaluation metrics, but it is not significant in most cases.

Table VIII. Statistics of the phrase tables trained on the genres of the NIST test corpora. Scoring: confidence estimation, selection: threshold.

eval-04	editorials	newswire	speeches	
sentences	449	901	438	
selected translations	101	187	113	
size of adapted phrase table	1,981	3,591	2,321	
adapted phrases used	707	1,314	815	
new phrases	679	1,359	657	
new phrases used	23	47	25	
eval-06	broadcast conversations	broadcast news	newsgroup	newswire
sentences	979	1,083	898	980
selected translations	477	274	226	172
size of adapted phrase table	2,155	4,027	2,905	2,804
adapted phrases used	759	1,479	1,077	1,115
new phrases	1,058	1,645	1,259	1,058
new phrases used	90	86	88	41

In all experiments on NIST, Algorithm 1 was run for one iteration. We also investigated the use of an iterative procedure here, but this did not yield any improvement in translation quality.

In order to see how useful the new models are for translation, we analyzed the phrase tables generated in semi-supervised learning (with confidence estimation for scoring and threshold-based selection) and the phrases which the SMT system actually used. The statistics are presented in Table VIII, separately analyzed for the different genres in the NIST test sets. It shows how many of the machine translations of the unlabeled data were considered reliable: In most cases, this is roughly a quarter of the translations. The exception is the broadcast conversation part of the 2006 data where almost half the translations are kept. On these sentence pairs, between 1,900 and 4,000 phrase pairs were learned for the different sub-corpora. The average phrase length is slightly above 2 words for both source and target phrases for all phrase tables (as opposed to an average length of 3-3.5 words in the original phrase tables). To see how useful this new phrase table actually is, we analyzed how many of the phrases which have been learned from the test corpus are used later in generating the best translations (after rescoring). The fourth row in each block shows that for all corpora, about 40% of the phrase pairs from the adaptive model are actually used in translation. So this phrase table which was trained using semi-supervised learning provides a very important source of information for the SMT system.

Out of the phrase pairs in the adaptive phrase table, 28% to 48% are entries which are not contained in the original phrase tables (see row 5, titled 'new phrases'). This shows that the system has learned new phrases through model adaptation. However, an analysis of the number of new phrase pairs which are actually used in translation (presented in the last row of each block of Table VIII) shows that the newly learned phrases are rarely employed. A comparative experiment showed that removing them from the adapted phrase table yields about the same gain in translation quality as the use of the full adapted phrase table. So the reward from model adaptation seems to come from the reinforcement of the relevant phrases in the existing phrase tables.

We leave it to future work to see whether the new phrases could actually be more useful in improving translation quality, and if the minimum error rate training of the feature weights could be modified to encourage the use of these new phrases from the domain to which we wish to adapt our translation system.

4.3. Translation examples

Table IX presents some French-English translation examples of the baseline and the adapted system where for Estimate, the mixture model was used; for Selection, all generated sentence pairs are used, and the test set consists of out of domain sentences. The examples are taken from the WMT 2006 EuroParl shared task test set. The parts of the translations which improve through model adaptation are highlighted in bold.

Table X presents some Chinese-English translation examples of the baseline and the adapted system using confidence estimation with a threshold in model adaptation. The square brackets indicate phrase boundaries. All examples are taken from the GALE portion of the 2006 test corpus. The domains are broadcast news and broadcast conversation. The examples show that the adapted system outperforms the baseline system both in terms of adequacy and fluency. Italics indicate bad phrase translations generated by the baseline system, and the better translations picked by the adapted system are printed in bold. The third example is especially interesting. An analysis showed that the target phrase "what we advocate" which is used by the baseline system is an overly confident entry in the original phrase table. The adapted

Table IX. Translation examples[a] from the WMT 2006 EuroParl corpus.

baseline	indeed , in *external policy* , *is inconsistency* is often a virtue .
adapted	indeed , in **foreign policy** , **the inconsistency** is often a virtue .
reference	indeed , in foreign policymaking , inconsistency is often a virtue .
baseline	but the faith in the taking of detailed information on the structure high-protein from patterns x-ray continues , *but only if this information could come to light* .
adapted	but the faith in the taking of information about the structure high-protein from x-ray detailed , **but only if information can be discovered** .
reference	but the faith remained that detailed information about protein structure could be obtained from the x-ray patterns in some way , if only it could be discovered .
baseline	the opportunities to achieve this are good *but are special efforts are indispensable .*
adapted	the chances of achieving this are good **but special efforts are still necessary .**
reference	there is a good chance of this , but particular efforts are still needed .
baseline	*this does not want to say first of all , as a result .*
adapted	**it does not mean that everything is going on .**
reference	this does not mean that everything has to happen at once .

[a] lower-cased output, punctuation marks tokenized

system, however, does not use this phrase here. This indicates that the shorter and more reliable phrases have been reinforced in model adaptation.

5. Previous Work

While many researchers have studied language model adaptation, there is not much work on translation model adaptation. One notable exception is (Hildebrand et al., 2005) where information retrieval is used to select training sentences similar to those in the test set. Unlike the work presented here, this approach still requires bilingual data from the domain it adapts to.

Semi-supervised learning has been previously applied to improve word alignments. In (Callison-Burch et al., 2004), a generative model for word alignment is trained using unsupervised learning on paral-

Table X. Translation examples[a] from the 2006 GALE corpus.

baseline	[the report said] [that the] [united states] [is] [a potential] [problem] [, the] [practice of] [china 's] [foreign policy] [is] [likely to] [*weaken us*] [*influence*] [.]
adapted	[the report] [said that] [this is] [a potential] [problem] [in] [the united states] [,] [china] [is] [likely to] [**weaken**] [**the impact of**] [**american foreign policy**] [.]
reference	the report said that this is a potential problem for america . china 's course of action could possibly weaken the influence of american foreign policy .
baseline	[*what we advocate*] [*his*] [*name*]
adapted	[**we**] [**advocate**] [**him**] [**.**]
reference	we advocate him .
baseline	[*the fact*] [*that this*] [is] [.]
adapted	[**this**] [is] [**the point**] [.]
reference	that is actually the point .
baseline	[*”*] [*we should*] [really be] [male] [*nominees*] [..] [....]
adapted	[**he**] [**should**] [be] [**nominated**] [male] [,] [really] [**.**]
reference	he should be nominated as the best actor , really .

[a] lower-cased output, punctuation marks tokenized

lel text. In addition, another model is trained on a small amount of hand-annotated word alignment data. A mixture model provides a probability for word alignment. Experiments showed that putting a large weight on the model trained on labeled data performs best. Along similar lines, (Fraser and Marcu, 2006) combine a generative model of word alignment with a log-linear discriminative model trained on a small set of hand aligned sentences. The word alignments are used to train a standard phrase-based SMT system, resulting in increased translation quality .

In (Callison-Burch, 2002) co-training is applied to MT. This approach requires several source languages which are sentence-aligned with each other and all translate into the same target language. One language pair creates data for another language pair and can be naturally used in a (Blum and Mitchell, 1998)-style co-training algorithm. Experiments on the EuroParl corpus show a decrease in WER. However, the selection algorithm applied there is actually supervised because it takes the reference translation into account. Moreover, when the algorithm is run long enough, large amounts of co-trained data injected too much noise and performance degraded.

Self-training has been investigated in other NLP areas, such as parsing. (McClosky et al., 2006a) introduces self-training techniques for two-step parsers. In (McClosky et al., 2006b), these methods are then used to adapt a parser trained on Wall Street Journal data, without using labeled data from the latter domain.

Self-training for SMT was proposed in (Ueffing, 2006). An existing SMT system is used to translate the development or test corpus. Among the generated machine translations, the reliable ones are automatically identified using thresholding on confidence scores. The work which we presented here differs from (Ueffing, 2006) as follows:

- We investigated different ways of scoring and selecting the reliable translations and compared our method to this work. In addition to the confidence estimation used there, we applied importance sampling and combined it with confidence estimation for semi-supervised model adaptation (see Table V).

- We studied additional ways of exploring the newly created bilingual data, namely re-training the full phrase translation model or creating a mixture model (see Section 4.2.1).

- We proposed an iterative procedure which translates the monolingual source language data anew in each iteration and then re-trains the phrase translation model (see Figure 1).

- We applied semi-supervised model adaptation not only to the phrase translation model, but also to the language model (see Table VI).

6. Discussion

It is not intuitively clear why the SMT system can learn something from its own output and is improved through semi-supervised model adaptation. There are two main reasons for this improvement: Firstly, the selection step provides important feedback for the system. The confidence estimation, for example, discards translations with low language model scores or posterior probabilities. The selection step discards bad machine translations and reinforces phrases of high quality. As a result, the probabilities of low-quality phrase pairs, such as noise in the table or overly confident singletons, degrade. Our experiments comparing the various settings for model adaptation shows that selection clearly outperforms the method which keeps all generated translations as additional training data. The selection methods investigated here have been shown to be well-suited to boost the performance of semi-supervised model adaptation for SMT.

Secondly, our algorithm constitutes a way of adapting the SMT system to a new domain or style without requiring bilingual training or development data from this domain. Those phrases or n-grams in the existing phrase tables or language model which are relevant for translating the new data are reinforced. The probability distribution over the events thus gets more focused on the (reliable) parts which are relevant for the test data.

We showed in this article how a phrase-based SMT system can benefit from semi-supervised learning. However, the method is applicable to other types of systems, such as syntax-based ones, as well.

In this work, the unlabeled data used in the algorithm is always the development or test data. In (Ueffing et al., 2008), the use of additional source language data in semi-supervised learning is explored. Unlike the experiments in this article, in that work the additional source data is from the same domain as the test data but does not include the test data itself. As a result, typically larger amounts of source data are required. These data are filtered w.r.t. the development or test set to identify the source data that is relevant to the domain and then used in a similar manner as described in this article. In scenarios where bilingual training data are scarce, an SMT system could be trained on a small amount of data and then iteratively improved by translating additional monolingual source language data and adding the reliable translations to the training material. This would be similar to bootstrapping approaches in speech recognition.

Acknowledgements

This material is partially based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-06-C-0023. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Defense Advanced Research Projects Agency (DARPA). The last author was partially supported by NSERC, Canada (RGPIN: 264905).

References

Abney, S.: 2004, ‘Understanding the Yarowsky Algorithm’. *Computational Linguistics* **30**(3), 365–395.

Blatz, J., E. Fitzgerald, G. Foster, S. Gandrabur, C. Goutte, A. Kulesza, A. Sanchis, and N. Ueffing: 2003, 'Confidence Estimation for Machine Translation'. Final report, JHU/CLSP Summer Workshop. http://www.clsp.jhu.edu/ws2003/groups/estimate/.

Blum, A. and T. Mitchell: 1998, 'Combining Labeled and Unlabeled Data with Co-Training'. In: *Proceedings of the Eleventh Annual Conference on Computational Learning Theory (COLT 1998)*. Madison, Wisconsin, USA.

Brants, T., A. C. Popat, P. Xu, F. J. Och, and J. Dean: 2007, 'Large Language Models in Machine Translation'. In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. pp. 858–867.

Brown, P. F., S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer: 1993, 'The Mathematics of Statistical Machine Translation: Parameter Estimation'. *Computational Linguistics* **19**(2), 263–311.

Callison-Burch, C.: 2002, 'Co-Training for Statistical Machine Translation'. Master's thesis, School of Informatics, University of Edinburgh.

Callison-Burch, C., D. Talbot, and M. Osborne: 2004, 'Statistical machine translation with word- and sentence-aligned parallel corpora'. In: *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*. Barcelona, Spain, Association for Computational Linguistics.

Foster, G., R. Kuhn, and H. Johnson: 2006, 'Phrasetable Smoothing for Statistical Machine Translation'. In: *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*. Sydney, Australia, pp. 53–61, Association for Computational Linguistics.

Fraser, A. and D. Marcu: 2006, 'Semi-Supervised Training for Statistical Word Alignment'. In: *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. Sydney, Australia, pp. 769–776, Association for Computational Linguistics.

Hildebrand, A. S., M. Eck, S. Vogel, and A. Waibel: 2005, 'Adaptation of the Translation Model for Statistical Machine Translation based on Information Retrieval'. In: *Proceedings of 10th Annual Conference of the European Association of Machine Translation (EAMT 2005)*. Budapest, Hungary.

McClosky, D., E. Charniak, and M. Johnson: 2006a, 'Effective Self-Training for Parsing'. In: *Proceedings of the Human Language Technology Conference of the North American chapter of the ACL, Main Conference (HLT NAACL 2006)*. New York City, USA, pp. 152–159, Association for Computational Linguistics.

McClosky, D., E. Charniak, and M. Johnson: 2006b, 'Reranking and Self-Training for Parser Adaptation'. In: *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. Sydney, Australia, pp. 337–344, Association for Computational Linguistics.

Nießen, S., F. J. Och, G. Leusch, and H. Ney: 2000, 'An evaluation tool for machine translation: Fast evaluation for MT research'. In: *Proceedings of the 2nd International Conference on Language Resources & Evaluation (LREC 2000)*. Athens, Greece, pp. 39–45.

Och, F. J.: 2003, 'Minimum Error Rate Training in Statistical Machine Translation'. In: *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL 2003)*. Sapporo, Japan, pp. 160–167.

Papineni, K., S. Roukos, T. Ward, and W.-J. Zhu: 2002, 'BLEU: a Method for Automatic Evaluation of Machine Translation'. In: *Proceedings of the 40th*

Annual Meeting of the Association for Computational Linguistics (ACL 2002). Philadelphia, PA, pp. 311–318.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery: 2002, *Numerical Recipes in C++*. Cambridge, UK: Cambridge University Press.

Stolcke, A.: 2002, 'SRILM - an extensible language modeling toolkit'. In: *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP 2002)*. Denver, Colorado, pp. 901–904.

Ueffing, N.: 2006, 'Using Monolingual Source-Language Data to Improve MT Performance'. In: *Proceedings of the International Workshop on Spoken Language Translation (IWSLT 2006)*. Kyoto, Japan, pp. 174–181.

Ueffing, N., G. Haffari, and A. Sarkar: 2008, 'Semi-supervised learning for Machine Translation'. In: *Learning Machine Translation*. NIPS Series, MIT Press.

Ueffing, N. and H. Ney: 2007, 'Word-Level Confidence Estimation for Machine Translation.'. *Computational Linguistics* **33**(1), 9–40.

Ueffing, N., M. Simard, S. Larkin, and H. Johnson: 2007, 'NRC's PORTAGE System for WMT 2007'. In: *Proceedings of the Second Workshop on Statistical Machine Translation*. Prague, Czech Republic, pp. 185–188, Association for Computational Linguistics.

Yarowsky, D.: 1995, 'Unsupervised Word Sense Disambiguation Rivaling Supervised Methods'. In: *Proceedings of 33rd Annual Meeting of the Association for Computational Linguistics (ACL 1995)*. Cambridge, Massachusetts, USA, pp. 189–196, Association for Computational Linguistics.

Zens, R. and H. Ney: 2006, 'N-Gram Posterior Probabilities for Statistical Machine Translation'. In: *Proceedings on the Workshop on Statistical Machine Translation*. New York City, pp. 72–77, Association for Computational Linguistics.