11 Semi-supervised learning for Machine Translation

Nicola Ueffing
Gholamreza Haffari
Anoop Sarkar

Statistical machine translation systems are usually trained on large amounts of bilingual text which is 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 chapter we explore the use of semi-supervised methods for the effective use of monolingual data from the source language in order to improve translation quality. In particular, in this work we use monolingual source language data from the same domain as the test set (without directly using the test set itself) and use semi-supervised methods for model adaptation to the test set domain. We propose several algorithms with this aim, and present the strengths and weaknesses of each one. We present detailed experimental evaluations using French–English and Chinese–English data and show that under some settings translation quality can be improved.

11.1 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:

$$\operatorname*{argmax}_{\mathbf{t}} p(\mathbf{t} \,|\, \mathbf{s}) = \operatorname*{argmax}_{\mathbf{t}} p(\mathbf{s} \,|\, \mathbf{t}) \cdot p(\mathbf{t}) \tag{11.1}$$

This decomposition of the probability yields two different statistical models which can be trained independently of each other: the translation model $p(\mathbf{s} \,|\, \mathbf{t})$ 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), and of monolingual target language corpora (for the target language model). It has been shown, e.g. in Brants et al. (2007), 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. However, the availability of monolingual corpora in the source language has not been shown to help improve the system's performance. We will 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, i.e. phrase pairs 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 learning. We translate sentences from the source language and use them to retrain the SMT system with the hope of getting a better translation system. The evaluation is done just once at the end of the learning process. Note the difference between this approach and the transductive approach in (Ueffing et al. (2007a)), where the latter treats the test set as the additional monolingual source data. In the work presented here, the additional monolingual source data is drawn from the same domain as the test set. In particular, we *filter* the monolingual source language sentences based on their similarity to the development set as explained in Section 11.3.3. Semi-supervised learning can be seen as a means to adapt the SMT system to a new domain or style that is different from the bilingual training data. For instance, a system trained on newswire could be used to translate weblog texts. The method proposed here adapts the trained models to the style and domain of the new domain without requiring bilingual data from this domain.

We present detailed experimental evaluations using French–English and Chinese–English data. In the French–English translation task we use bilingual data from the EuroParl corpus, and use monolingual data from the same domain as our test set which is drawn from the Canadian Hansards corpus. In the Chinese–English task we use bilingual data from the NIST large-data track and use monolingual data from the Chinese Gigaword corpus.

11.2 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 at 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. (2007b).

The PORTAGE system determines the translation of a given source sentence **s** by maximizing the posterior probability over all possible translations **t** as shown in Eq. 11.1. This posterior probability is approximated by the log-linear combination of models $g_i(\mathbf{s}, \mathbf{t}), i = 1, \ldots, I$, taking both languages into account (such as the translation model in Eq. 11.1), and target-language-based models $h_j(\mathbf{t}), j = 1, \ldots, J$ (such as the language model in Eq. 11.1). The decoder solves the following equation:

$$\operatorname*{argmax}_{\mathbf{t}} p(\mathbf{t}, \mathbf{s}) = \operatorname*{argmax}_{\mathbf{t}} \prod_{i=1}^{I} g_i(\mathbf{s}, \mathbf{t})^{\alpha_i} \cdot \prod_{j=1}^{J} h_j(\mathbf{t})^{\beta_j} \tag{11.2}$$

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

- one or several phrase table(s), which model the translation direction $p(\mathbf{s} \mid \mathbf{t})$. 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 described in Stolcke (2002b); in the experiments reported here, we used several 4-gram models on the Chinese–English data, and a trigram model on French-English,
- 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 constant cost to each generated target word. This constitutes a way to control the length of the generated translation.

These different models are combined log-linearly as shown in Eq. 11.2. Their weights $\alpha_i, i = 1, \ldots, I, \beta_j, j = 1, \ldots, J$ are optimized with respect to BLEU score (Papineni et al. (2002a)) using the algorithm described in Och (2003a). This optimization is done on a development corpus.

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 from Brown et al. (1993a): 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,
- several higher-order n-gram language models,
- posterior probabilities for words, phrases, n-grams, and sentence length (Zens and Ney (2006); Ueffing and Ney (2007)). All of these posterior probabilities are calculated over the N-best list and using the sentence probabilities which the baseline system assigns to the translation hypotheses. More details on the calculation of posterior probabilities will be given in Section 11.3.5.

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.

11.3 The Framework

11.3.1 The Yarowsky Algorithm

The original Yarowsky algorithm (Yarowsky (1995)) was proposed in the context of a word-sense disambiguation task. The model was a simple decision list classifier $\Pr(j \mid x, \theta)$ where the class label $j \in \mathcal{L}$ is predicted based on the single most likely feature extracted from the input x. The pseudo-code for this algorithm is shown in Algorithm 11.1. The key idea is to use an initial classifier which was built from the seed data in such a way so as to have high precision but low recall on the unlabeled set since it could not predict any label for most examples. This is is because if a feature extracted from input x has not been observed with any class label, this event is not assigned a smoothed probability estimate (unlike the common strategy in supervised learning). Hence for examples where all the features are events of this type, the classifier labels it as $\perp$. However, even if a single feature extracted from x is observed with a class label in the labeling step, this information can be used to make a prediction for future examples by the decision list classifier (which only uses the single most likely feature to predict the class label). This classifier was then used to label the unlabeled data and those examples labeled with high confidence (above a threshold) were used along with the labeled data to train a new classifier. This process was repeated iteratively until no new labels could be found for the unlabeled data set. Each iteration could be seen as improving the recall of the classifier at the expense of its precision. In each iteration the classifier is able to provide labels for larger and larger subsets of the unlabeled data.

There are several different types of iterative self-training semi-supervised algorithms that have been proposed in the literature. Haffari and Sarkar (2007); Abney (2004) provides a more detailed discussion on the relationship between these algorithms and the Yarowsky algorithm.

11.3.2 Semi-supervised Learning Algorithm for SMT

Our semi-supervised learning algorithm, Algorithm 11.2, is inspired by the Yarowsky algorithm described in Section 11.3.1. 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 investigated in Ueffing et al. (2008).

The algorithm works as follows: First, the translation model is estimated based on the sentence pairs in the bilingual training data L. The set of source language sentences, U, is sorted according to the relevance with respect to the development

Algorithm 11.1 Bootstrapping algorithm: classifier version

```
Input: each example $x$ is either labeled $L(x)$ in some annotated data, or unlabeled as $U^0(x) := \perp$.
$t := 0$
repeat
  for each example $x$ do
    Training step: Estimate $\theta$ for $\Pr(j \mid x, \theta)$ using $L$ and $U^t(x)$
    Labeling step: $U^{t+1}(x) = \begin{cases} \underset{j \in \mathcal{L}}{\operatorname{argmax}} \Pr(j \mid x, \theta) & \text{if } \Pr(j \mid x, \theta) > \text{ threshold } \zeta \\ \perp & \text{otherwise} \end{cases}$
  end for
  $t := t + 1$
until for all $x$: $U^{t+1}(x) = U^t(x)$
```

corpus C, and a chunk of sentences, U_i, is filtered. These sentences are then translated based on the current model. The SMT system generates an N-best list of translation alternatives for each source sentence. This set X_i of translations is then scored, and a subset of good translations and their sources, T_i, is selected from X_i in each iteration and added to the training data. The process of generating sentence pairs, scoring them, selecting a subset of good sentence pairs, and updating the model is continued until a stopping condition is met. In the experiments presented here, this stopping criterion is either a fixed number of iterations, or we run for a fixed number of iterations on a development set, and pick the translation model based on the iteration that provides the highest improvement on the development set. Note that here we use the first for the French–English experiments and both for the Chinese–English experiments.

Note that the Algorithm 11.2 is a variation of the original Yarowsky algorithm in which the same unlabeled data is used in each iteration.

It has been shown by Abney (2004) that Algorithm 11.2 minimizes the entropy of the probability distribution $p(\mathbf{t} \,|\, \mathbf{s})$ over translations of the unlabeled data set U. However, this is true only when the functions **Estimate**, **Score** and **Select** have very prescribed definitions. Rather than analyzing the convergence of Algorithm 11.2, we will use definitions for Estimate, Score and Select that have been experimentally shown to improve MT performance. Following Ueffing et al. (2007a), these are different versions of the algorithm for the two different translation tasks we work on. We will present the different variants of the functions Filter, Estimate, Score, and Select in the following subsections. The exact experimental settings for the two translation tasks will be described in Section 11.4.1.

In Ueffing et al. (2007a), a transductive variant of Algorithm 11.2 was introduced which uses the development or test corpus as unlabeled data U. That is, this corpus is translated, reliable translations are selected and used in (re-)training to improve the performance of the SMT system. This approach generates a very small amount of new bilingual data of high relevance. Unlike the approach described in Ueffing et al. (2007a), we explore much larger amounts of monolingual source-language data.

Algorithm 11.2 Semi-supervised learning 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: dev corpus $C$.
Input: number of iterations $R$, and size of $N$-best list.
$T_{-1} := \{\}$.            // Additional bilingual training data.
$i := 0$.                    // Iteration counter.
repeat
    Training step: $\pi^{(i)} := \mathbf{Estimate}(L, T_{i-1})$.
    $X_i := \{\}$.           // The set of generated translations for this iteration.
    $U_i := \mathbf{Filter}(U, C, i)$   // The $i$-th chunk of unlabeled sentences.
    for sentence $\mathbf{s} \in U_i$ do
        Labeling step: Decode $\mathbf{s}$ using $\pi^{(i)}$ to obtain $N$ best sentence pairs with their scores
        $X_i := X_i \cup \{(\mathbf{t}_n, \mathbf{s}, \pi^{(i)}(\mathbf{t}_n \mid \mathbf{s}))_{n=1}^{N}\}$
    end for
    Scoring step: $S_i := \mathbf{Score}(X_i)$
    // Assign a score to sentence pairs $(\mathbf{t}, \mathbf{s})$ from $X_i$.
    Selection step: $T_i := T_i \cup \mathbf{Select}(X_i, S_i)$
    // Choose a subset of good sentence pairs $(\mathbf{t}, \mathbf{s})$ from $X_i$.
    $i := i + 1$.
until $i > R$
```

In order to identify the relevant parts of these data, we filter them as explained in the following.

11.3.3 The Filter Function

In general, having more training data improves the quality of the trained models. However, when it comes to the translation of a particular test set, the question is whether *all* of the available training data are relevant to the translation task or not. Moreover, working with large amounts of training data requires more computational power. So if we can identify a subset of training data which are relevant to the current task and use only this to (re-)train the models, we can reduce the computational complexity significantly.

We propose to **Filter** the additional monolingual source data to identify the parts which are relevant with respect to the development set. This filtering is based on n-gram coverage. For a source sentence $\mathbf{s}$ in the monolingual data, its n-gram coverage over the sentences in the development set is computed. The average over several n-gram lengths is used as a measure of relevance of this training sentence with respect to the development corpus. In the experiments presented here, this is the average over 1- to 6-gram coverage. We sort the source sentences by their coverage and successively add them as unlabeled data U_i in Algorithm 11.2.

11.3.4 The Estimate Function

The Estimate function estimates a phrase translation model from the sets of bilingual data, L and T_{i-1}. Out of the three different versions of this function presented in Ueffing et al. (2007a), we use the one which performed best for our experiments here: training an additional phrase translation model on the new bilingual data T_{i-1}. That is, the phrase translation model learned on the original bilingual data L is kept fixed, and a new model is learned on T_{i-1} only and then added as a new component in the log-linear SMT model presented in Eq. 11.2. This additional model is relatively small and specific to the test corpus C. We have to learn a weight for this new phrase translation model. To this end, the weight optimization is carried out again on the development set. After the first iteration, we re-optimize the decoder and rescoring weights for all original models and this new phrase translation model. These weights are then kept fixed throughout the following iterations. This re-optimization process is computationally expensive, so we carry it out only once.

11.3.5 The Scoring Function

In Algorithm 11.2, 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 $p(\mathbf{t} \mid \mathbf{s})$ (assigned by the SMT system) normalized by the length $|\mathbf{t}|$ of the target sentence:

$$\mathbf{Score}(\mathbf{t}, \mathbf{s}) \quad = \quad p(\mathbf{t} \mid \mathbf{s})^{\frac{1}{|\mathbf{t}|}} \tag{11.3}$$

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. 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

$$p_{\text{lev}}(t \,|\, \mathbf{s}, \mathbf{t}, \mathcal{L}) \quad = \quad \frac{\sum\limits_{n=1}^{N} \delta(t, \mathcal{L}_i(\mathbf{t}, \mathbf{t}_n)) \cdot p(\mathbf{s}, \mathbf{t}_n)}{\sum\limits_{n=1}^{N} p(\mathbf{s}, \mathbf{t}_n)} \tag{11.4}$$

The sum is normalized by the total probability mass of the N-best list. 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 $p(\mathbf{s}, \mathbf{t}_n)$ 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 this 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 Chinese–English. 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 with respect to 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.

11.3.6 The Selection Function

The **Select** function in Algorithm 11.2 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. It has been shown in Ueffing et al. (2007a) that this selection is an important step in the algorithm and that simply keeping all generated translations yields worse results. We use the following selection functions:

Importance Sampling: For each sentence $\mathbf{s}$ in the set of unlabeled sentences U_i, the Labeling step in Algorithm 11.2 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_i 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 11.3.5. 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. Otherwise 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.

Top K: This method simply keeps those translations from T_i which receive the highest scores in the scoring step.

11.4 Experimental Results

11.4.1 Setting

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

The variants of Algorithm 11.2 which we applied in the experiments are the ones which yielded the best results in Ueffing et al. (2007a). On the French–English task, we experimented with various settings which are described in detail in Section 11.4.3. For the Chinese–English task, the application of threshold-based selection in combination with confidence scores is applied.

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

Table 11.1 French–English corpora.

corpus	use	sentences
EuroParl-training	phrase table + language model	688K
EuroParl-test2006	in-domain dev1	500
EuroParl-test2006	out-of-domain dev2	500
EuroParl-devtest2006	dev3	2,000
Hansards-training	monolingual source data	1130K
EuroParl-test2006	in-domain test	1,500
EuroParl-test2006	out-of domain test	500

EuroParl French–English

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 (WMT)[2], and the Hansards corpus as distributed by ISI[3]. The corpus statistics are shown in Table 11.1. The bilingual training data from the EuroParl corpus is used to train translation and language models. The development sets dev1 and dev2 are used to optimize the model weights in the decoders for the baseline SMT system and the SMT system with an additional phrase table respectively. The evaluations are done on the test set provided for the NAACL 2006 French-English translation shared task, which contains 2,000 in-domain sentence 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.

Chinese–English

For the Chinese–English translation task, we used the corpora distributed for the large-data track in the 2006 NIST evaluation (see Table 11.2). We used the LDC segmenter for Chinese. A subset of the English Gigaword corpus was used as additional LM training material. Data from the Chinese Gigaword was filtered and used as additional monolingual source-language data for semi-supervised learning. The multiple translation corpora multi-p3 and multi-p4 were used as development corpora. Evaluation was performed on the 2004 and 2006 test sets. The 2006 test set consists of two sections: the NIST section which was created for the NIST machine translation evaluation in 2006 and which is provided with four English references, and the GALE section which was created in the DARPA project GALE and comes with one English reference.

[2] www.statmt.org/wmt06/shared-task/
[3] www.isi.edu/natural-language/download/hansard/

Table 11.2 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
Chinese Gigaword	additional source data	50K	news
multi-p3	optimize decoder	935	news
multi-p4	optimize rescoring	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

Note that the training data consists 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.

Evaluation Metrics

We evaluated the generated translations using three different automatic evaluation metrics. They all compare the generated translation to one or more given reference translations. The following criteria are used:

- BLEU (**bil**ingual **e**valuation **u**nderstudy) (Papineni et al. (2002a)):

The BLEU score is based on the notion of modified n-gram precision, for which all candidate n-gram counts in the translation are collected and clipped against their corresponding maximum reference counts. These clipped candidate counts are summed and normalized by the total number of candidate n-grams.

- WER (**w**ord **e**rror **r**ate):

The word error rate is based on the Levenshtein distance. It is computed as the minimum number of substitution, insertion and deletion operations that have to be performed to convert the generated translation into the reference translation. In the case where several reference translations are provided for a source sentence, we calculate the minimal distance to this set of references as proposed in Nießen et al. (2000a).

- PER (**p**osition-independent word **e**rror **r**ate) (Nießen et al. (2000a)):

A shortcoming of the WER is the fact that it requires a perfect word order. In order to overcome this problem, the position independent word can be used, comparing the words in the two sentences *without* taking the word order into account. Words that have no matching counterparts are counted as substitution errors, missing words are deletion and additional words are insertion errors. The PER is a lower bound for the WER.

Note that BLEU score measures translation quality, whereas WER and PER measure translation errors.

We will present 95%-confidence intervals for the baseline system which are calculated using bootstrap resampling. The metrics are calculated with respect to one or four English references: the French–English data comes with one reference, the Chinese–English 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 set (see Section 11.4.2). Note that these values do not indicate lower translation quality, but are simply a result of using only one reference.

11.4.2 Chinese-English Results

On the Chinese–English translation task, we used additional source language data from the Chinese Gigaword corpus comprising newswire text for our semi-supervised learning algorithm. The Chinese Gigaword sentences are sorted according to their n-gram overlap with the development corpus (see Section 11.3.3). It is assumed that the test set is unknown at this point. The Chinese sentences are then divided into chunks of 5,000 sentences. One of these chunks is added in each iteration as described in Algorithm 11.2.

Figure 11.1 shows the BLEU score on the development set over the iterations. As to be expected, the biggest jump occurs after the first iteration when the decoder weights are re-optimized. Up to iteration 4, which is equivalent to the use of 20,000 additional Chinese sentences in semi-supervised learning, the BLEU score increases. But after that, it drops and levels off at a point which is above the baseline, but does not significantly differ from it anymore. So it seems that if the semi-supervised training runs too long, it adds noise into the model rather than improving it. The overlap between the additional data and the development corpus decreases over the iterations, so that the added data might be less relevant and thus actually hurt translation quality rather than improving it.

After analyzing the results obtained on the development corpus, we evaluated three different systems on the test corpora: the system after the first iteration, which used 5,000 additional Chinese sentences in semi-supervised training and for

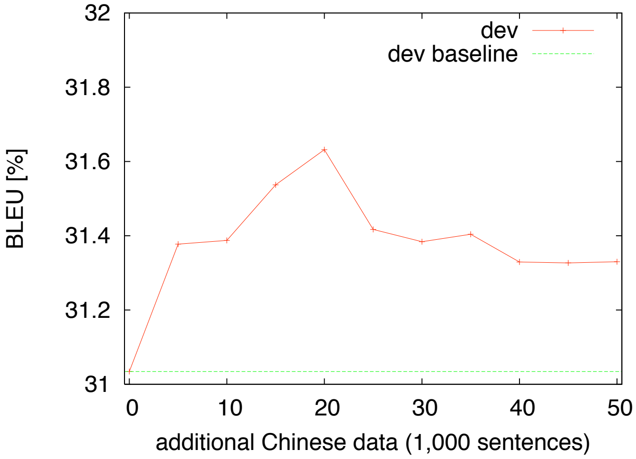

Figure 11.1 Translation quality using an additional phrase table trained on monolingual Chinese news data. Chinese–English development set.

which the weight optimization was carried out; the system after iteration 4 which performed best on the development set; and the system after the last iteration which had the highest number of additional Chinese sentences, namely 50,000, available for semi-supervised learning. The last setup was tested only for comparison, as the results presented in Figure 11.1 indicate already that this system might perform worse than the other two. Table 11.3 shows the translation quality achieved on the Chinese–English test sets with these three systems. The best system is clearly the one obtained after the first iteration. The translation quality is significantly better than the baseline in most cases. The system from iteration 4 (which performs best on the development set) shows a very similar performance in terms of translation quality as the first one. The error rates it achieves are slightly higher than those of the first system, but still significantly better than those of the baseline system. The third system, however, does not outperform the baseline system. As mentioned above, this was to be expected.

Table 11.4 analyzes the translation results achieved on the eval-04 test corpus separately for each genre: editorials, newswire, and political speeches. The most significant improvement in translation quality is achieved on the newswire section of the test corpus. Interestingly, all three semi-supervised systems perform very similarly on this genre, whereas performance decreases on the other two genres as the number of iterations increases. This difference among the genres can be explained by the fact that the additional data is drawn from the Chinese Gigaword corpus which contains newswire data.

In Ueffing et al. (2007a), the transductive approach was evaluated on the same Chinese–English test sets. The translation quality achieved by this approach is higher than that of the method presented here (yielding a BLEU score which is 0.5-1.1 points higher). We see two reasons for this difference: Firstly, transductive

Table 11.3 Translation quality using an additional phrase table trained on monolingual Chinese news data. Chinese–English test sets. **Bold:** best result, *italic:* significantly better than baseline.

system		BLEU[%]	WER[%]	PER[%]
eval-04 (4 refs.)				
baseline		31.8±0.7	66.8±0.7	41.5±0.5
add Chinese data	iteration 1	***32.8***	***65.7***	***40.9***
	iteration 4	*32.6*	*65.8*	***40.9***
	iteration 10	32.5	66.1	41.2
eval-06 GALE (1 ref.)				
baseline		12.7±0.5	75.8±0.6	54.6±0.6
add Chinese data	iteration 1	**13.1**	***73.9***	***53.5***
	iteration 4	13.0	*75.0*	*53.9*
	iteration 10	12.7	75.4	54.9
eval-06 NIST (4 refs.)				
baseline		27.9±0.7	67.2±0.6	44.0±0.5
add Chinese data	iteration 1	28.1	***65.8***	***43.2***
	iteration 4	**28.2**	*65.9*	*43.4*
	iteration 10	27.7	*66.4*	43.8

learning on the development or test corpus yields a model which is more focused on this corpus. It adapts the system directly to the domain and style by creating an additional phrase table which is specific to the development or test corpus and matches it very well. Secondly, the transductive approach adapts the SMT system to each of the genres. In the work presented here, the additional Chinese data came from the newswire domain only, and this yields a higher boost in translation quality for this genre than for the other ones. It would be interesting to see how the system performs if data from all domains in the test corpus are available for semi-supervised learning. We also investigated a combination of the two self-training methods: using additional source language data as well as the development or test corpus for transductive learning. Unfortunately, the gains achieved by the two methods do not add up, and this system does not outperform the transductively trained one.

Table 11.5 shows how many translations were identified as confident by the scoring and selection algorithm and used to extend the additional phrase table. In the first iteration, this is approximately two thirds of the data added in the iteration. But as the semi-supervised training proceeds, the number of confident sentences decreases. After ten iterations of the algorithm, less than half of the translations are kept. This confirms our assumption that noise is introduced into the procedure by running the algorithm for too long.

Table 11.4 Translation quality on Chinese–English eval-04 test set, by genre. Same experimental setup as Table 11.3.

system		BLEU[%]	WER[%]	PER[%]
editorials	baseline	30.7±1.2	67.0±1.1	42.3±0.9
	iteration 1	31.3	65.9	41.8
	iteration 4	30.9	66.2	42.0
	iteration 10	30.8	66.6	42.3
newswire	baseline	30.0±0.9	69.1±0.8	42.7±0.8
	iteration 1	31.1	68.1	42.0
	iteration 4	31.1	67.9	42.0
	iteration 10	31.3	68.1	42.1
speeches	baseline	36.1±1.4	62.5±1.2	38.6±0.9
	iteration 1	37.3	61.3	38.0
	iteration 4	36.8	61.5	38.0
	iteration 10	36.3	61.8	38.4

Table 11.5 Number of sentences added in each iteration. Chinese–English.

iteration	1	2	3	4	5
# confident sentences	3,141	3,196	3,017	2,889	2,981
iteration	6	7	8	9	10
# confident sentences	2,890	2,520	2,423	2,324	2,427

11.4.3 French–English Results

We ran our experiments on the French–English task to explore the behavior of the semi-supervised learning algorithm with respect to the different sentence selection methods on in-domain and out-of-domain test sentences. We used 688k parallel sentence pairs from the EuroParl corpus as the bilingual training data, and partitioned the NAACL 2006 WMT shared task's test set into two sets (S_{in} and S_{out}) to separate in-domain and out-of-domain test sentences. S_{in} includes the first 2000 sentences and S_{out} includes the last 1000 sentences of this test set. Then we used the first 500 sentences in S_{in} and S_{out} as the development sets dev1 and dev2, and used the rest as the test sets. As the additional monolingual source sentences, we used the French sentences in the training set of the Canadian Hansards corpus as provided by ISI.

The monolingual French sentences were sorted according to their n-gram overlap (see Section 11.3.3) with the development corpora dev1 and dev2 for in-domain and out-of-domain experiments, and 5,000 French sentences were added in each iteration of the semi-supervised algorithm. The scoring and selection of the translations (see Algorithm 11.2) were performed using:

1. confidence score with importance sampling,
2. length normalized translation score with importance sampling,
3. confidence score with a threshold, and
4. keeping the top-K sentence pairs having the highest length normalized translation scores.

We learn an additional phrase table on these data (and leave the original phrase tables unmodified) which is added as a new component in the log-linear model. The weight of this new component is optimized based on dev1 and dev2 for in-domain and out-of-domain experiments. Moreover, we use the development set dev3 for estimating the parameters of the confidence estimation model and the threshold in method 3.

The results of the four sentence selection methods can be seen in Figure 11.2. The semi-supervised algorithm deteriorates the performance of the initial SMT system for all cases except sampling with normalized translation scores. This method however, yields an improvement on the out-of-domain data, so the system seems to adapt to this new domain. This observation is confirmed by the translation examples which will be presented in Section 11.4.4. Note that the performance of top-K sentence pair selection based on the normalized scores is encouraging for both in-domain and out-of-domain experiments. It is probable that by choosing K in a more elaborate way, this method outperforms the baseline and other methods. Note that it just uses the normalized translation scores which are already generated by the decoder. Using dev1 and dev2 to train the confidence estimation models for in-domain and out-of-domain experiments, may help the methods which use the confidence to boost their performance.

11.4.4 Translation examples

Table 11.6 presents some French-English translation examples taken from out-of-domain sentences in the test set of the NAACL 2006 WMT shared task. These examples show the effect of semi-supervised learning for model adaptation to a novel domain.

Table 11.7 presents some Chinese-English translation examples of the baseline and the semi-supervised system using a phrase table learned on 5,000 additional Chinese sentences. All examples are taken from the NIST portion of the 2006 test corpus. Except for the last example, which is taken from newswire, they come from newsgroup posts. The examples show that the semi-supervised system outperforms the baseline system both in terms of adequacy and fluency.

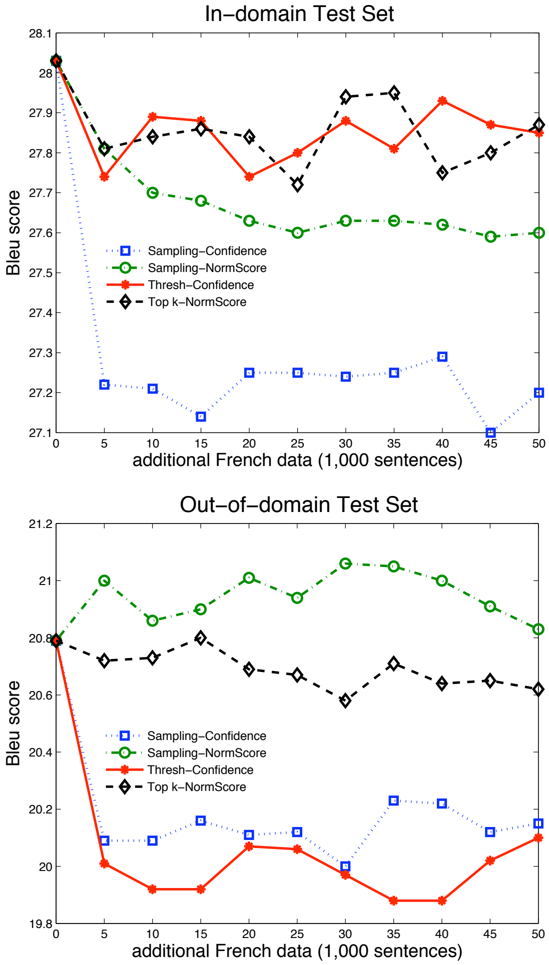

Figure 11.2 Translation quality using an additional phrase table trained on monolingual French data. The plot shows the performance of the different sentence selection and scoring schemes on in-domain and out-of-domain corpora.

11.5 Previous Work

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 parallel 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

Table 11.6 Translation examples from the out-of-domain sentences in the NAACL 2006 French–English test corpus. The *semi-supervised* examples are taken from sampling-based sentence selection with normalized scores. Lower-cased output, punctuation marks tokenized.

baseline	the so-called ’ grandfather bulldozer become the most of the israelis and the final asset of western diplomacy and for the americans , surprisingly , for europeans too .
semi-supervised	’bulldozer’ became the grandfather of most of the israelis and the final asset of western diplomacy and for the americans , surprisingly , for europeans too .
reference	the ” bulldozer ” had become the grandfather of most israelis and the last card of western diplomacy , for americans and , surprisingly , for europeans , too .
baseline	these are not all their exceptional periods which create bonaparte , which is probably better because leaders exceptional can provide the illusion that all problems have their solution , which is far from true .
semi-supervised	these are not all periods which create their exceptional bonaparte , which is probably better because leaders exceptional can provide the illusion that all problems have their solution which is far from being true .
reference	not all exceptional periods create their bonapartes , and this is probably a good thing , for exceptional leaders may give the illusion that all problems have solutions , which is far from true .
baseline	in both cases , must be anchored in good faith moving from one to another .
semi-supervised	in both cases , it must be established for faith moving from one to another .
reference	in both cases , it takes a lot of blind faith to go from one to the other .
baseline	given an initial period to experiment with growth and innovation on these fronts may prove strong paying subsequently .
semi-supervised	enjoy an initial period of growth and innovation to experiment with on these fronts may prove heavily paying subsequently .
reference	using an initial period of growth to experiment and innovate on these fronts can pay high dividends later on .

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.

Table 11.7 Translation examples from the Chinese–English eval-06 corpus, NIST section. Lower-cased output, punctuation marks tokenized.

baseline	you will continue to be arrested and beaten by villagers .
semi-supervised	you continue to arrest , beat villagers ,
reference	you have continued to arrest and beat villagers .
baseline	after all , family planning is a problem for chinese characteristics .
semi-supervised	after all , family planning is a difficult problem with chinese characteristics .
reference	after all , family planning is a difficult topic with chinese characteristics .
baseline	i am very disappointed in recognition of the chinese people do not deserve to enjoy democracy ! ! !
semi-supervised	i am very disappointed to admit that the chinese nation do not deserve democracy ! ! !
reference	i am very disappointed to admit that the chinese people do not deserve democracy !
baseline	china has refused to talk to both sides to comment .
semi-supervised	the chinese side refused to comment on both sides of the talks .
reference	china has refused to comment on the talks between the two sides .
baseline	reports said that there has been speculation that might trigger a computer in possession by the former metropolitan police chief steve vincent jazz yangguang survey of confidential information .
semi-supervised	reports said that the theft triggered speculation that the computer may be in the possession of the metropolitan police chief stevenson jazz led investigation of confidential information .
reference	the report pointed out that the theft triggered speculation that the computers may contain confidential information of the probe led by former metropolitan police commissioner lord stevens .

Self-training for SMT was proposed in Ueffing et al. (2007a) where the test data was repeatedly translated and phrase pairs from the translated test set were used to improve overall translation quality. In the work presented here, the additional monolingual source data is drawn from the same domain as the test set. In particular, we *filter* the monolingual source language sentences based on their similarity to the development set as explained in Section 11.3.3.

11.6 Conclusion and Outlook

We presented a semi-supervised learning algorithm for SMT which makes use of monolingual source-language data. The relevant parts of these data are identified, and then the SMT system is used to generate translations of those. The reliable translations are automatically determined and used to retrain and adapt the SMT system to a domain or style. It is not intuitively clear why the SMT system can learn something from its own output and is improved through semi-supervised learning. 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. The selection methods investigated here have been shown to be well-suited to boost the performance of semi-supervised learning 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. Those phrases in the existing phrase tables which are relevant for translating the new data are reinforced. The probability distribution over the phrase pairs thus gets more focused on the (reliable) parts which are relevant for the test data.

One of the key components in our approach is that translations need to be proposed for sentences in the unlabeled set (which is from the same domain as the test set), and from those translations, we would like to select the ones that are useful in improving our performance in this domain. For this problem, we plan to explore some alternatives in addition to the methods presented here in future work : in translating a source sentence **f**, the difficulty in assessing the quality of a translation into the target language **e**' comes from the fact that we do not have any reference translation **e**. However, if **e**' is a good translation, then we should probably be able to *reconstruct* the input sentence **f** from it. So we can judge the quality of **e**' based on its translation **f**' (we can compute the BLEU score in this case since we *do* have access to **f**), and for this translation direction we already have the translation probability tables. This approach is an attractive alternative to the problem of selecting good translations in our algorithm.

In addition to this, it would be interesting to study the proposed methods further, using more refined filter functions, e.g. methods applied in information retrieval.

11.7 Acknowledgments

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).