Transductive learning for statistical machine translation

Nicola Ueffing
National Research Council Canada
Gatineau, QC, Canada
nicola.ueffing@nrc.gc.ca

Gholamreza Haffari and **Anoop Sarkar**
Simon Fraser University
Burnaby, BC, Canada
{ghaffar1,anoop}@cs.sfu.ca

Abstract

Statistical machine translation systems are usually trained on large amounts of bilingual text and monolingual text in the target language. In this paper we explore the use of transductive semi-supervised 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.

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:

$$\arg\max_{\mathbf{t}} p(\mathbf{t} \,|\, \mathbf{s}) = \arg\max_{\mathbf{t}} p(\mathbf{s} \,|\, \mathbf{t}) \cdot p(\mathbf{t}) \qquad (1)$$

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 $p(\mathbf{s} \,|\, \mathbf{t})$), and of monolingual target language corpora (for $p(\mathbf{t})$). It has been shown that adding large amounts of target language text improves translation quality considerably. However, the availability of monolingual corpora in the source language does not 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 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 transductive semi-supervised learning. By transductive, we mean that we repeatedly translate sentences from the development set or test set and use the generated translations to improve the performance of the SMT system. Note that the evaluation step is still done just once at the end of our learning process. In this paper, 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.

Transductive 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.

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 which has been made available

Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 25–32,
Prague, Czech Republic, June 2007.

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: (a) one or several phrase table(s), which model the translation direction $p(\mathbf{s} \mid \mathbf{t})$, (b) one or several n-gram language model(s) trained with the SRILM toolkit (Stolcke, 2002); in the experiments reported here, we used 4-gram models on the NIST data, and a trigram model on EuroParl, (c) a distortion model which assigns a penalty based on the number of source words which are skipped when generating a new target phrase, and (d) a word penalty. 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 paper. 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: (a) the different models used in the decoder which are described above, (b) two different features based on IBM Model 1 (Brown et al., 1993), (c) 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, dev2.

3 The Framework

3.1 The Algorithm

Our transductive learning algorithm, Algorithm 1, is inspired by the Yarowsky algorithm (Yarowsky, 1995; Abney, 2004). The algorithm works as follows: First, the translation model 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 a stopping condition is met. Note that we run this algorithm in a transductive setting which means that the set of sentences U is drawn either from a development set or the test set that will be used eventually to evaluate the SMT system or from additional data which is relevant to the development or test set. 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 paper.

Given the probability model $p(\mathbf{t} \mid \mathbf{s})$, consider the distribution over all possible valid translations **t** for a particular input sentence **s**. We can initialize this probability distribution to the uniform distribution for each sentence **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 paper, rather than analyze 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 all translation models): 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. The size of L can be controlled by **filtering** the training data (see Section 3.5).

Additional Phrase Table: If, on the other hand, a new phrase translation table is learned on T only and then added as a new component in the log-linear model, we have an alternative to the full re-training

Algorithm 1 Transductive 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: number of iterations R, and size of n-best list N.
$T_{-1} := \{\}$. // 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 sentence pairs 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.
    Selection step: $T_i := \textbf{Select}(X_i, S_i)$ // Choose a subset of *good* sentence pairs $(\mathbf{t}, \mathbf{s})$ from X.
    $i := i + 1$.
until $i > R$

of the model on labeled and unlabeled data which can be very expensive if L is very large (as on the Chinese–English data set). This additional phrase table is small and specific to the development or test set it is trained on. It overlaps with the original phrase tables, but also contains many new phrase pairs (Ueffing, 2006).

Mixture Model: Another alternative for **Estimate** is to create a mixture model of the phrase table probabilities with new phrase table probabilities

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

where L_p and T_p 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 $p(\mathbf{t} \,|\, \mathbf{s})$ normalized by the length $|\mathbf{t}|$ of the target sentence:

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

Confidence Estimation: 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 phrase posterior probabilities, Levenshtein-based word posterior probabilities, and a target language model score. The weights of the different scores are optimized w.r.t. classification error rate (CER).

The phrase posterior probabilities are determined 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 decoder. This sum is then 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 phrases are multiplied. 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. For details, see (Ueffing and Ney, 2007). Again, the single values are multiplied to obtain a score for the whole sentence. For NIST, the language model score is determined using a 5-gram model trained on the English Gigaword corpus, and on French–English, we use the trigram model which was provided for the NAACL 2006 shared task.

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 original bilingual training data. We use the following selection functions:

Importance Sampling: For each sentence **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 for each translation **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. These 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 filtering 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.

3.5 Filtering the Training Data

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 computational complexity significantly.

We propose to **Filter** the training data, either bilingual or monolingual text, to identify the parts which are relevant w.r.t. the test set. This filtering is based on n-gram coverage. For a source sentence **s** in the training data, its n-gram coverage over the sentences in the test set is computed. The average over several n-gram lengths is used as a measure of relevance of this training sentence w.r.t. the test corpus. Based on this, we select the top K source sentences or sentence pairs.

corpus	use	sentences
EuroParl	phrase table+LM	688K
train100k	phrase table	100K
train150k	phrase table	150K
dev06	dev1	2,000
test06	test	3,064

Table 1: French–English corpora

corpus	use	sentences
non-UN	phrase table+LM	3.2M
UN	phrase table+LM	5.0M
English Gigaword	LM	11.7M
multi-p3	dev1	935
multi-p4	dev2	919
eval-04	test	1,788
eval-06	test	3,940

Table 2: NIST Chinese–English corpora

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 (www.nist.gov/speech/tests/mt).

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. The corpus statistics are shown in Table 1. Furthermore we filtered the EuroParl corpus, as explained in Section 3.5, to create two smaller bilingual corpora (train100k and train150k in Table 1). 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.

For the Chinese–English translation task, we used the corpora distributed for the large-data track in the

setting		EuroParl	NIST
full re-training w/ filtering		$*$	$**$
full re-training		$**$	†
mixture model		$*$	†
new phrase table ff:			
keep all		$**$	$*$
imp. sampling	norm.	$**$	$*$
	conf.	$**$	$*$
threshold	norm.	$**$	$*$
	conf.	$**$	$*$

Table 3: Feasibility of settings for Algorithm 1

2006 NIST evaluation (see Table 2). We used the LDC segmenter for Chinese. The multiple translation corpora multi-p3 and multi-p4 were used as development corpora. Evaluation was performed on the 2004 and 2006 test sets. 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.

Given the particular data sets described above, Table 3 shows the various options for the **Estimate**, **Score** and **Select** functions (see Section 3). The table provides a quick guide to the experiments we present in this paper 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 paper. The entries marked as † were not attempted because they are not feasible (e.g. full re-training on the NIST data). However, these were run on the smaller EuroParl corpus.

Evaluation Metrics

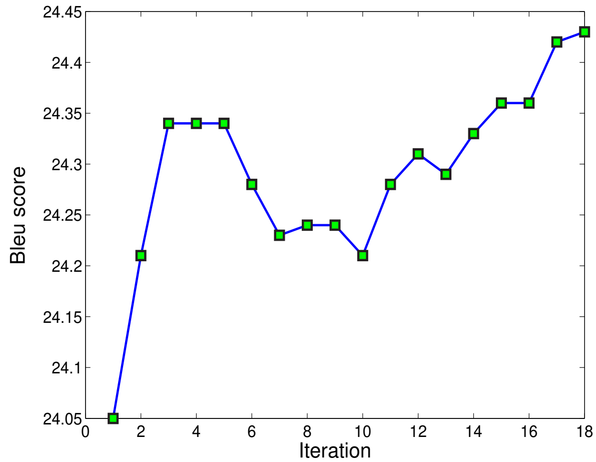
We evaluated the generated translations using three different evaluation metrics: BLEU score (Papineni et al., 2002), mWER (multi-reference word error rate), and mPER (multi-reference position-independent word error rate) (Nießen et al., 2000). Note that BLEU score measures quality, whereas mWER and 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 with one reference, the NIST 2004 evaluation set and the NIST section of the 2006 evaluation set are provided with four references each, whereas 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 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

EuroParl

We ran our initial experiments on EuroParl to explore the behavior of the transductive learning algorithm. In all experiments reported in this subsection, the test set was used as unlabeled data. The selection and scoring was carried out using importance sampling with normalized scores. In one set of experiments, we used the 100K and 150K training sentences filtered according to n-gram coverage over the test set. 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 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 transductive learning 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



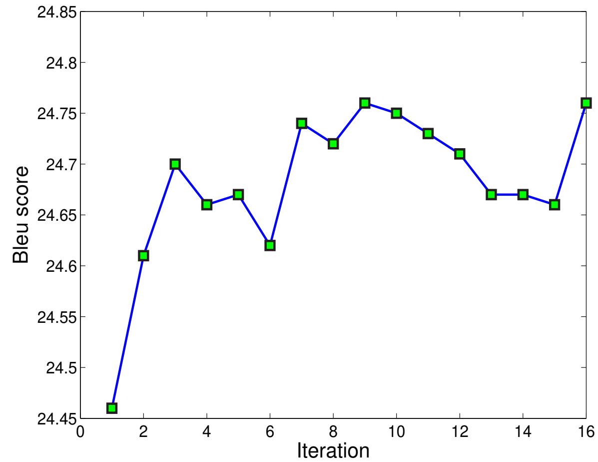


Figure 1: Translation quality for importance sampling with full re-training on train100k (left) and train150k (right). EuroParl French–English task.

is a small increase, it shows that the unlabeled data contains some information which can be explored in transductive learning.

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 learned phrase table in each iteration with a weight of $\lambda = 0.1$. This value for λ was found based on cross validation on a development set. We ran the algorithm for 20 iterations and BLEU score increased from 25.3 to 25.7. Since this is very similar to the result obtained with the previous method, but with an additional parameter λ to optimize, we did not use mixture models on NIST.

Note that the single improvements achieved here are slightly below the 95%-significance level. However, we observe them consistently in all settings.

NIST

Table 4 presents translation results on NIST with different versions of the scoring and selection methods introduced in Section 3. In these experiments, the unlabeled data U for Algorithm 1 is the development or test corpus. For this corpus U, 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 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 transductive learning and used with the weight which has been learned on dev1. In the first experiment presented in Table 4, all of the generated 1-best translations were kept and used for training the adapted phrase tables. This method yields slightly higher translation quality than the baseline system. The second approach we studied is the use of importance sampling (IS) over 20-best lists, based either on length-normalized sentence scores (norm.) or confidence scores (conf.). As the results in Table 4 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.

Table 5 shows the translation quality achieved on the NIST test sets when additional source language data from the Chinese Gigaword corpus comprising newswire text is used for transductive learning. These Chinese sentences were sorted according to their n-gram overlap (see Section 3.5) with the development corpus, and the top 5,000 Chinese sentences were used. The selection and scoring in Algorithm 1 were performed using confidence estimation with a threshold. Again, a new phrase table was trained on these data. As can be seen in Table 5, this

select	score	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
IS	norm.	**33.5**	65.8	40.9
	conf.	33.2	65.6	**40.4**
thr	norm.	**33.5**	65.9	40.8
	conf.	**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
IS	norm.	13.2	74.7	54.1
	conf.	12.9	74.4	53.5
thr	norm.	12.7	75.2	54.2
	conf.	**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
IS	norm.	28.7	66.1	43.6
	conf.	28.4	65.8	**43.2**
thr	norm.	28.3	66.1	43.5
	conf.	**29.3**	**65.6**	**43.2**

Table 4: Translation quality using an additional adapted phrase table trained on the dev/test sets. Different selection and scoring methods. NIST Chinese–English, best results printed in boldface.

system	BLEU[%]	mWER[%]	mPER[%]
eval-04 (4 refs.)			
baseline	31.8±0.7	66.8±0.7	41.5±0.5
add Chin. data	32.8	65.7	40.9
eval-06 GALE (1 ref.)			
baseline	12.7±0.5	75.8±0.6	54.6±0.6
add Chin. data	13.1	73.9	53.5
eval-06 NIST (4 refs.)			
baseline	27.9±0.7	67.2±0.6	44.0±0.5
add Chin. data	28.1	65.8	43.2

Table 5: Translation quality using an additional phrase table trained on monolingual Chinese news data. Selection step using threshold on confidence scores. NIST Chinese–English.

system outperforms the baseline system on all test corpora. The error rates are significantly reduced in all three settings, and BLEU score increases in all cases. A comparison with Table 4 shows that transductive learning on the development set and test corpora, adapting the system to their domain and style, is more effective in improving the SMT system than the use of additional source language data.

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.

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 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 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 transductive learning.

- We studied additional ways of exploring the newly created bilingual data, namely re-training the full phrase translation model or creating a mixture model.
- We proposed an iterative procedure which translates the monolingual source language data anew in each iteration and then re-trains the phrase translation model.
- We showed how additional monolingual source-language data can be used in transductive learning to improve the SMT system.

6 Discussion

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. Our experiments comparing the various settings for transductive learning 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 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. For an analysis of the self-trained phrase tables, examples of translated sentences, and the phrases used in translation, see (Ueffing, 2006).

References

S. Abney. 2004. Understanding the Yarowsky Algorithm. *Comput. Ling.*, 30(3).

J. Blatz, 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. www.clsp.jhu.edu/ws2003/groups/estimate/.

A. Blum and T. Mitchell. 1998. Combining Labeled and Unlabeled Data with Co-Training. In *Proc. Computational Learning Theory*.

P. F. Brown, 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).

C. Callison-Burch, D. Talbot, and M. Osborne. 2004. Statistical machine translation with word- and sentence-aligned parallel corpora. In *Proc. ACL*.

C. Callison-Burch. 2002. Co-training for statistical machine translation. Master's thesis, School of Informatics, University of Edinburgh.

A. Fraser and D. Marcu. 2006. Semi-supervised training for statistical word alignment. In *Proc. ACL*.

S. Nießen, F. J. Och, G. Leusch, and H. Ney. 2000. An evaluation tool for machine translation: Fast evaluation for MT research. In *Proc. LREC*.

F. J. Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. ACL*.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. ACL*.

A. Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proc. ICSLP*.

N. Ueffing and H. Ney. 2007. Word-level confidence estimation for machine translation. *Computational Linguistics*, 33(1):9–40.

N. Ueffing, M. Simard, S. Larkin, and J. H. Johnson. 2007. NRC's Portage system for WMT 2007. In *Proc. ACL Workshop on SMT*.

N. Ueffing. 2006. Using monolingual source-language data to improve MT performance. In *Proc. IWSLT*.

D. Yarowsky. 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *Proc. ACL*.

R. Zens and H. Ney. 2006. N-gram posterior probabilities for statistical machine translation. In *Proc. HLT/NAACL Workshop on SMT*.