Discriminative Adaptation of Continuous Space Translation Models

Quoc-Khanh Do[1,2]*, Alexandre Allauzen*[1,2]*, François Yvon*[1]

LIMSI-CNRS[1] and Univ. Paris-Sud[2], rue John von Neumann, F 91 403 Orsay

{dokhanh,allauzen,yvon}@limsi.fr

Abstract

In this paper we explore various adaptation techniques for continuous space translation models (CSTMs). We consider the following practical situation: given a large scale, state-of-the-art SMT system containing a CSTM, the task is to adapt the CSTM to a new domain using a (relatively) small in-domain parallel corpus. Our method relies on the definition of a new discriminative loss function for the CSTM that borrows from both the max-margin and pair-wise ranking approaches. In our experiments, the baseline out-of-domain SMT system is initially trained for the WMT News translation task, and the CSTM is to be adapted to the lecture translation task as defined by IWSLT evaluation campaign. Experimental results show that an improvement of 1.5 BLEU points can be achieved with the proposed adaptation method.

1. Introduction

Domain adaptation (DA) is an important and active research topic in Statistical Natural Language Processing [1, 2]. In a nutshell, domain adaptation aims to mitigate the well-known problem of *covariate shift* which stems from statistical distribution differences between train and test samples. This often happens in NLP, especially when train and test documents correspond to different genres, registers or domains. Domain adaptation is often expressed in terms of finding an optimal combination of a small in-domain dataset with large amounts of out-of-domain data.

To avoid the dilution of domain-specific knowledge, most approaches consider various kinds of data weighting schemes in order to balance the importance of in-domain *vs* out-of-domain data. In such adaptation scenarios, the NLP component needs to be retrained, entirely or partly, to integrate these new samples, which can be very time consuming or even unrealistic in many situations. This is especially problematic for SMT systems, that are typically made of several layers of statistical models. DA for SMT has therefore received considerable attention in the recent years (for instance [3, 4, 5, 6]). This situation is compounded when, as we do here, SMT systems rely on Continuous Space Language Models (CSLMs) or Translation Models (CSTMs), which have recently gained a lot of popularity [7, 8, 9, 10, 11, 12].

As demonstrated for many NLP tasks [13], such as language modelling [7, 14, 15, 16], syntactic parsing [17] and machine translation [8, 9, 18, 19], CSLMs and CSTMs can remedy to two well-know issues of statistical modelling for linguistic data. Typical statistical models use discrete random variables to represent the realization of words, phrases or phrase pairs. The corresponding parameter estimates are based on relative frequencies and are unreliable for rare events. Furthermore, the resulting representations ignore morphological, syntactic and semantic relationships that exist among linguistic units. This lack of structure hinders the generalization power of statistical models and reduces their ability to adapt to other domains. By contrast, continuous models manipulate numerical representations of linguistic units that are automatically trained from large corpora and that implicitly capture some similarity relationships, thereby introducing some smoothing in the probability estimates.

The adaptation of Continuous Models for SMT has thus far received little attention. We study here the following practical situation: a large scale, state-of-the-art SMT system is available and needs to be ported to a new domain, using a small in-domain parallel corpus. In this setting, our main contribution is the definition and evaluation of new loss functions, that aim at discriminatively adapting the CSTMs to the new data. These objective functions derive from both the max-margin [20, 21] and pair-wise ranking [22, 23] approaches. In our experiments, the baseline, out-of-domain system is preliminarily trained for the News translation task, and the CSTMs must be adapted to the lecture translation task as defined in recent IWSLT evaluation campaigns [24].

The rest of the paper is organized as follows. Section 2 briefly describes the model structure that will be used in our experiments. Section 3 proposes new discriminative loss functions on N-best lists, along with the corresponding adaptation algorithms. The next section gives details about our experimental conditions and analyzes our main results. We finally provide a short review of similar works both on Discriminative Machine Translation and on Continuous Space Translation Models, before concluding with some perspectives for future work.

2. Continuous space translation models

This section provides an overview of the CSTM used in our baseline system and subsequently adapted. This model was introduced and fully described in [9].

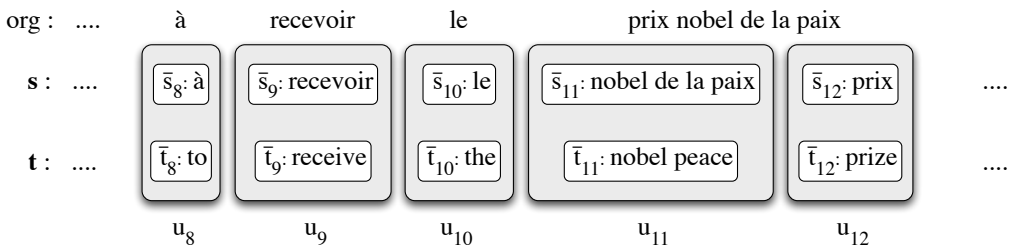

Figure 1: Extract of a French-English sentence pair segmented in bilingual units. The original (*org*) French sentence appears at the top of the figure, just above the reordered source $\mathbf{s}$ and target $\mathbf{t}$. The pair $(\mathbf{s}, \mathbf{t})$ decomposes into a sequence of L bilingual units (*tuples*) $\mathrm{u}_1, ..., \mathrm{u}_L$. Each tuple u_i contains a source and a target phrase: $\bar{s}_i$ and $\bar{t}_i$.

2.1. The n-gram translation model

n-gram translation models (TMs) rely on a specific decomposition of the joint probability $P(\mathbf{s}, \mathbf{t})$, where $\mathbf{s}$ is a sequence of I *reordered* source words $(s_1, ..., s_I)$[1] and $\mathbf{t}$ contains J target words $(t_1, ..., t_J)$. This sentence pair is assumed to be decomposed into a sequence of L bilingual units called *tuples* defining a joint segmentation: $(\mathbf{s}, \mathbf{t}) = (\mathrm{u}_1, ..., \mathrm{u}_L)$. In this framework, the basic translation units are *tuples*, which are the analogous of phrase pairs, and represent a matching $\mathrm{u} = (\bar{s}, \bar{t})$ between a source $\bar{s}$ and a target $\bar{t}$ phrase (Figure 1). Using the n-gram assumption, the joint probability of a *synchronized* and *segmented* sentence pair is:

$$P(\mathbf{s}, \mathbf{t}) = \prod_{i=1}^{L} P(\mathrm{u}_i | \mathrm{u}_{i-n+1}^{i-1}), \quad (1)$$

where u_{i-n+1}^{i-1} denotes the tuple sequence $\mathrm{u}_{i-n+1}, \ldots, \mathrm{u}_{i-1}$. The complete model for a sentence pair thus involves latent variables that specify the reordering applied to the source sentence, as well as its segmentation into translation units. These latent variables define the derivation of the source sentence that generates the target sentence. They are omitted for the sake of clarity. During the training step, the segmentation is a by-product of source reordering, and ultimately derives from initial word and phrase alignments (see [25, 26] for details). During the inference step, the SMT decoder will compute and output the best derivation.

In this model, the elementary units are bilingual pairs, which means that the underlying vocabulary, hence the number of parameters, can be quite large, even for small translation tasks. Due to data sparsity issues, such models face severe estimation problems. Equation (1) can therefore be factored by decomposing tuples in two (source and target) parts and in two equivalent ways:

$$\begin{aligned} & P(\mathrm{u}_i | \mathrm{u}_{i-n+1}^{i-1}) \\ & \quad = P(\bar{t}_i | \bar{s}_{i-n+1}^{i}, \bar{t}_{i-n+1}^{i-1}) P(\bar{s}_i | \bar{s}_{i-n+1}^{i-1}, \bar{t}_{i-n+1}^{i-1}) \quad (2) \\ & \quad = P(\bar{s}_i | \bar{t}_{i-n+1}^{i}, \bar{t}_{i-n+1}^{i-1}) P(\bar{t}_i | \bar{s}_{i-n+1}^{i-1}, \bar{t}_{i-n+1}^{i-1}) \end{aligned}$$

[1] In the context of the n-gram translation model, $(\mathbf{s}, \mathbf{t})$ thus denotes an *aligned* sentence pair, where the source words are reordered.

Each decomposition involves two bilingual conditional distributions that can also be decomposed at the level of words, using again the n-gram assumption.

2.2. Continuous translation modeling with SOUL

The n-gram distributions described in Section 2.1 are defined over potentially large vocabularies. As proposed in [9], these distributions can be estimated using the SOUL model introduced in [27]. Following [28], the SOUL model combines the feed-forward neural network approach for n-gram models [7] with a class-based prediction [29]. Structuring the output layer with word-class information makes the estimation of distributions over the entire vocabulary computationally feasible. Neural network architectures are also interesting as they can easily handle larger contexts than typical n-gram models. In the SOUL architecture, enlarging the context mainly consists in increasing the size of the projection layer, which corresponds to a simple look-up operation. Increasing the context length at the input layer thus causes only a linear growth in complexity in the worst case [14].

2.3. Training and initialization issues

The word-based translation model described in section 2.1 involves two different languages and thus two different vocabularies: the predicted unit is a target or source word, whereas the context is made of both source and target words. As proposed in [9], the SOUL architecture is modified to make up for *mixed* contexts by considering two different sets of word embeddings, one for each language. Training this kind of model can be achieved by maximizing the log-likelihood on some parallel corpus. Following [9], this optimization is performed by stochastic back-propagation, while the derivation (source reordering and segmentation in translation units) are derived by the usual procedure (see [30]).

However, for multi-layered neural networks, the non-convexity of the objective function implies that the parameter initialization can highly impact the training process in terms of its convergence speed and of its performance. In the bilingual context of translation modeling, two monolingual language models can first be estimated for initialization

purpose[2]. In a domain adaptation context, we assume that an existing CSTM –trained on the out-of-domain data– already exists. This model is thus well suited to bootstrap the adaptation process.

3. Objective functions for adaptation

In most previous works (eg. [8, 9]), CSTMs are estimated by maximizing the regularized conditional log-likelihood (CLL) on parallel training corpora. This estimation procedure is used to train a baseline CSTM on the out-of-domain corpus, producing a baseline model that will serve as an initial point for domain adaptation. Given a small in-domain parallel corpus, the same training procedure can also be used. A straightforward adaptation algorithm consists in running a few epochs of the standard back-propagation algorithm on the in-domain data to maximize the conditional likelihood using, as initial parameters, the out-of-domain model.

There is however only a loose relationship between the CLL criterion and the final translation quality. The CSTM is usually integrated in the translation process through a reranking step, the goal of which is to reorder a reduced set of candidate translations, called N-best list. Therefore, to better take advantage of the small amount of in-domain data, we propose to explore alternative objective functions that are more directly related to the translation quality (as reflected by the BLEU score) after reranking. We first present the general learning algorithm, then the various objective functions.

3.1. Rescoring N-best lists with CSTMs

Due to the high computational cost of normalizing the output layer, continuous models are in most cases[3] introduced in a post-processing step called N-best reranking.

We thus assume that for each source sentence $\mathbf{s}$, the decoder can generate an N-best list $\{\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_N\}$ of N top translation candidates. Each hypothesis $\mathbf{h}_i = (\mathbf{t}_i, \mathbf{a}_i)$ is associated with the decoder score $F_{\boldsymbol{\lambda}}(\mathbf{s}, \mathbf{h})$ computed as:

$$F_{\boldsymbol{\lambda}}(\mathbf{s}, \mathbf{h}) = \sum_{k=1}^{K} \lambda_k f_k(\mathbf{s}, \mathbf{h}), \tag{3}$$

where K feature functions (f_k) are weighted by a set of coefficients (λ_k). The n-gram approach differs from other approaches by the hidden variables associated to derivations, such as the source word reordering and the segmentation of the resulting parallel sentence. The basic feature functions used in this study are very similar to those used by standard phrase-based SMT systems (see [30] for instance).

When reranking with a continuous space model, $F_{\boldsymbol{\lambda}}(.)$ is augmented to also include an additional feature denoted $f_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h})$. As explained in Section 2.2, $f_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h})$ typically corresponds to the negated log-probability of the derivation: $f_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}) = -\log P_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h})$, where $\boldsymbol{\theta}$ is the vector containing the CSTM's free parameters. The scoring function used in reranking is then:

$$G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}) = F_{\boldsymbol{\lambda}}(\mathbf{s}, \mathbf{h}) + \lambda_{K+1} f_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}) \tag{4}$$

This scoring function depends on the CSTM's parameters $\boldsymbol{\theta}$, as well as on the coefficients $\boldsymbol{\lambda}$ of the scoring function. In the approach proposed here, optimizing the reranking step will thus requires *to alternatively tune the vector of coefficients $\boldsymbol{\lambda}$ and to adapt the CSTM's weight vector $\boldsymbol{\theta}$*: the former procedure uses the development data, while the latter will use the in-domain parallel corpus.

Algorithm 1 Joint optimization procedure for $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$

```
Initialize θ and λ
for each iteration do
    for M mini-batches do                          ▷ λ is fixed
        Compute the sub-gradient of L(θ, s) for all s in the mini-batch
        Update θ
    end for
    Update λ using dev set                         ▷ θ is fixed
end for
```

The corresponding proposed optimization procedure splits the in-domain set in mini-batches of a fixed size (typically 128 subsequent sentence pairs). As sketched in Algorithm 1, each mini-batch is used to update the parameters $\boldsymbol{\theta}$ of the CSTM while keeping $\boldsymbol{\lambda}$ fixed. The vector $\boldsymbol{\lambda}$ is updated every M mini-batches.

In our study, tuning $\boldsymbol{\lambda}$ is performed using standard tools (here, the K-Best Mira algorithm described in [21] as implemented in MOSES [4]). The training of CSTMs (with fixed $\boldsymbol{\lambda}$) is more interesting and we compare two discriminative objective functions, which aim at better taking the translation quality into account. These two objectives are in turn compared to the conventional maximization of the conditional likelihood criterion on parallel data.

3.2. A max-margin approach

As explained above, each hypothesis $\mathbf{h}_i$ produced by the decoder is scored according to (4). Its quality can also be evaluated by the sentence-level approximation of the BLEU score $sBLEU(\mathbf{h}_i)$. Let $\mathbf{h}^*$ denote the hypothesis with the best sentence BLEU score. A max-margin loss function [33, 34, 20] for estimating $\boldsymbol{\theta}$ can then be formulated as follows:

$$\mathcal{L}_{mm}(\boldsymbol{\theta}, \mathbf{s}) = -G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}^*) + \max_{1 \le j \le N} \left(G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}_j) + \text{cost}_\alpha(\mathbf{h}_j) \right), \tag{5}$$

where $\text{cost}_\alpha(\mathbf{h}_j) = \alpha\big(sBLEU(\mathbf{h}^*) - sBLEU(\mathbf{h}_j)\big)$. The parameter α mitigates the contribution of the cost function

[2] The following parameters can be initialized given a source and target language monolingual models: the source and target word embeddings respectively, and the structured output layer's structure.

[3] See however [31, 32, 19] for early attempts to integrate Neural Network Translation Models within the decoder.

[4] http://www.statmt.org/moses/

to the objective function. When $alpha > 0$, the objective defined in (5) is a general max-margin training criterion; taking $\alpha = 0$ corresponds to the structured perceptron loss [35]. This objective function aims to discriminatively learn to give the highest model score to the hypothesis $\mathbf{h}^*$ having the best sentence level BLEU. Moreover, the margin term enforces the scoring difference between $\mathbf{h}^*$ and the rest of the N-best list to be greater than the margin.

However, a source sentence can have, among the N-best list, several good translations that differ only slightly from the best hypothesis. The max-margin objective function defined above nevertheless considers that all hypotheses, except the best one, are wrong. The ranking-based approach defined below tries to correct this weakness.

3.3. Pairwise ranking

Inspired by [22], we define another objective function that aims to learn the ranking of a set of hypotheses with respect to their BLEU scores. Assuming that r_i denotes the rank of the hypothesis $\mathbf{h}_i$ when the N-best list is reordered according to the sentence-level BLEU, this objective is defined as:

$$\mathcal{L}_{pro}(\boldsymbol{\theta}, \mathbf{s}) = \sum_{1 \leq i,k \leq N} \mathbb{I}_{\{r_i + \delta \leq r_k, G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s},\mathbf{h}_i) < G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s},\mathbf{h}_k)\}} \left(-G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}_i) + G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}_k)\right). \quad (6)$$

Note that this loss function only involves a subset of the $N(N-1)/2$ pairs of hypotheses, since two hypotheses are included in the sum only if they are sufficiently apart in terms of their ranks: formally, the absolute difference of ranks should be greater than a predefined threshold δ. As in PRO [22], the ranking problem is thus reduced to a binary classification task taking candidate translation pairs as inputs. A major difference to PRO though, is the fact that we use this loss function to train the CSTM's parameters $\boldsymbol{\theta}$, rather than the feature weights $\boldsymbol{\lambda}$.

This ranking criterion can finally be generalized again with the notion of margin: for a pair of hypotheses $(\mathbf{h}_i, \mathbf{h}_k)$ such as $r_i + \delta < r_k$, the scoring difference $G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}_i) - G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}_k)$ should exceed a positive margin. As in section 3.2, the margin is based on the sentence-level BLEU score via the use of the cost function cost_α. Let us define the set of all critical pairs of hypotheses as:

$$\mathcal{C}_\delta^\alpha = \{(i,k) : 1 \leq i,k \leq N, r_i + \delta \leq r_k, \quad (7)$$
$$G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}_i) - G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}_k) < \text{cost}_\alpha(\mathbf{h}_k) - \text{cost}_\alpha(\mathbf{h}_i)\}.$$

The objective function that combines both the pairwise ranking and max-margin criterion is defined as follows:

$$\mathcal{L}_{pro-mm}(\boldsymbol{\theta}, \mathbf{s}) = \sum_{(i,k) \in \mathcal{C}_\delta^\alpha} \text{cost}_\alpha(\mathbf{h}_k) - \text{cost}_\alpha(\mathbf{h}_i) - G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}_i) + G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}_k). \quad (8)$$

Taking $\alpha = 0$, this function is equivalent to the pairwise ranking criterion (6).

4. Experiments

We now turn to an experimental comparison of the adaptation methods described in Section 3. In our experimental framework, the lecture translation task defines the targeted (or in) domain, while the baseline system corresponds to a state-of-the-art SMT system, intensively trained for the News translation task, as defined by the WMT evaluation. The goal is therefore to quickly and efficiently adapt this out-of-domain system by only updating the CSTM.

4.1. Task and corpora

The task considered here is derived from the text translation track of IWSLT 2011 from English to French (the TED Talks task [24]), where a (in-domain) training dataset containing $107,058$ aligned sentence pairs was made available. As explained above, this corpus only serves to adapt the continuous space translation models, *i.e* to adapt the parameters $\boldsymbol{\theta}$. The baseline and out-of-domain system is trained in the condition of the shared translation task of WMT 2013 evaluation campaign.[5] This system includes CSTMs that will be used as starting points for adaptation.

The official development and test sets respectively contain 934 and $1,664$ sentence pairs. Following [9], these sets are swapped, the tuning of the feature weights $\boldsymbol{\lambda}$ is carried out on $1,664$ sentences of the latter, while the final test is on 934 sentences of the former. Translations are evaluated using the BLEU score [36]. For a fair comparison, all BLEU scores reported are obtained after a tuning phase on the dev set, including the baseline system. For Algorithm 1, $(\boldsymbol{\theta}, \boldsymbol{\lambda})$ are selected by maximizing the BLEU score on the dev set (line 7).

4.2. Baseline system and models

The n-gram-based system used here is based on an open source implementation[6] of the bilingual n-gram approach to Statistical Machine Translation [37]. In a nutshell, the translation model is implemented as a stochastic finite-state transducer trained using an n-gram model of (source, target) pairs as described in section 2.1. Training this model requires to reorder source sentences so as to match the target word order. This is performed by a non-deterministic finite-state reordering model, which uses part-of-speech information generated by the TreeTagger to generalize reordering patterns beyond lexical regularities.

In addition to the TM, fourteen feature functions are included that are similar to the standard phrase-based system: *target-language model*; four *lexicon models*; six *lexicalized reordering models*; a distance-based *distortion model*; and finally a *word-bonus model* and a *tuple-bonus model*. A more detailed description is in [30].

[5] http://www.statmt.org/wmt13/

[6] `perso.limsi.fr/Individu/jmcrego/bincoder`

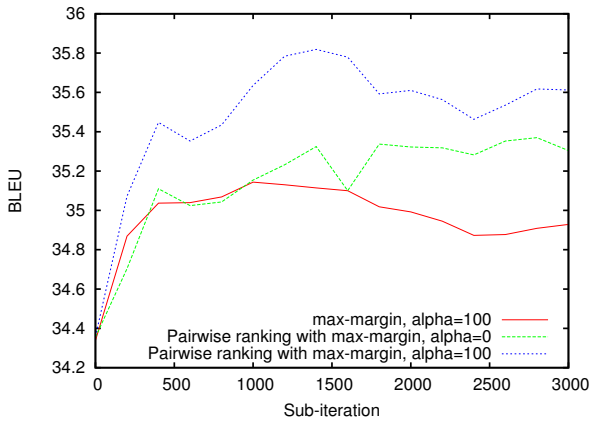

Figure 2: Evolution of BLEU scores on the dev set using three discriminative criteria described in (5), (6) and (8). Vector $\boldsymbol{\lambda}$ is updated every 200 sub-iterations (mini-batches).

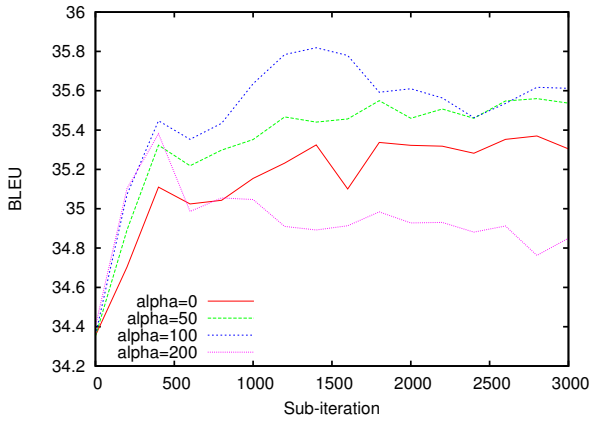

Figure 3: Evolution of BLEU scores on the dev set with different values of α. $\mathcal{L}_{pro-mm}$ is used in all cases.

4.3. Experimental results

The baseline, out-of-domain, system is used to generate the 300-best list for the in-domain corpus. It takes approximatively an half an hour if this process is parallelized by dividing the corpus in about 50 parts of $20,000$ sentences. δ is set to 250 (equations (6) and (7)) in all our experiments with the pairwise ranking criterion.

As reflected in equation (2), 4 translation models can be defined by various factorizations of $P(\mathbf{s}, \mathbf{t})$. For the sake of clarity, we focus our study on models estimating $P(\bar{t}_i|\bar{s}_{i-n+1}^{i}, \bar{t}_{i-n+1}^{i-1})$ and $P(\bar{t}_i|\bar{s}_{i-n+1}^{i-1}, \bar{t}_{i-n+1}^{i-1})$. We first compare the different objective functions defined in Section 3 and examine the impact of the margin on the former model. We then choose the best configuration to adapt the latter. Similar trends were observed with other CSTMs.

Figure 2 compares the three discriminative criteria respectively defined by (5), (6) and (8) in terms of BLEU scores on the dev set when adapting $P(\bar{t}_i|\bar{s}_{i-n+1}^{i}, \bar{t}_{i-n+1}^{i-1})$.

System	*dev*	*test*
Baseline systems (out-of-domain)		
n-code	33.9	27.6
n-code + CSTM WMT	34.4	28.5
Adapted systems		
n-code + CSTM CLL adapted	35.0	29.1
n-code + CSTM $\mathcal{L}_{mm}$ adapted $\alpha = 100$	35.1	29.4
n-code + CSTM $\mathcal{L}_{pro}$ adapted	35.4	29.5
n-code + CSTM $\mathcal{L}_{pro-mm}$ adapted $\alpha = 100$	35.8	29.6

Table 1: BLEU scores obtained for different adaptation schemes of the CSTM for $P(\bar{t}_i|\bar{s}_{i-n+1}^{i}, \bar{t}_{i-n+1}^{i-1})$ with WMT baselines: maximum conditional likelihood (CLL) *vs* discriminative adaptation. Log-linear coefficients of the baseline systems are re-tuned using the in-domain dev set.

Table 1 gives BLEU scores on both dev and test sets. According to these results, the pairwise ranking criterion, with or without max-margin((6) and (8)) clearly outperforms the max-margin approach (5) on the dev set. Further analyses (not detailed here) on each criterion's behaviour on the training set suggest that continuous space models quickly overfit the training data when adapted with the max-margin criterion. This result may outline the benefit of using criteria based on multiples hypotheses from different parts of the N-best list, rather than only on the best hypothesis and the most critical one as does the max-margin loss. Because of the superiority of the pairwise ranking approach, the rest of this section focuses on this criterion.

To assess the impact of the margin in $\mathcal{L}_{pro-mm}$, we plot on Figure 3 the evolution of the BLEU score on the dev set as a function of α. When $\alpha = 0$, the objective function only considers the pairwise ranking criterion $\mathcal{L}_{pro}$. By increasing α, we observe an improvement of 0.4 BLEU point, while beyond $\alpha = 100$, the performance starts to drop.

The results of adapting $P(\bar{t}_i|\bar{s}_{i-n+1}^{i}, \bar{t}_{i-n+1}^{i-1})$ are in Table 1. The upper part reports the baseline BLEU scores. Initial results were obtained with the out-of-domain one-pass system, and a 0.9 BLEU point improvement was obtained when reranking its output with the out-of-domain CSTM. The lower part of Table 1 summarizes the results obtained with various adaptation methods: the conditional likelihood (CLL) adaptation technique yields an additional increase of 0.6 BLEU point, which is nearly doubled when using the discriminative objective function $\mathcal{L}_{pro-mm}$ to perform adaptation. As showed in the middle part of Table 2, similar improvements are obtained with the adaptation of $P(\bar{t}_i|\bar{s}_{i-n+1}^{i-1}, \bar{t}_{i-n+1}^{i-1})$.

Finally, the lower part of Table 2 compares the performance obtained by our discriminative adaptation method to the one published in [9] for the same experimental setup. In our experiment (the last line), in-domain data are only used in two phases: the retuning of feature weights $\boldsymbol{\lambda}$; and the separate discriminative adaptation of two CSTMs. In [9], the

System	*dev*	*test*
Baseline systems (out-of-domain)		
n-code	33.9	27.6
n-code + CSTM WMT	34.6	28.2
Adapted systems		
n-code + CSTM CLL adapted	35.1	28.7
n-code + CSTM $\mathcal{L}_{pro-mm}$ adapted $\alpha = 100$	35.3	29.4
Model combination		
n-code (+TED) + all CSTMs CLL adapted [9]	36	29.7
n-code + all WMT CSTMs + 2 CSTMs $\mathcal{L}_{pro-mm}$	**36.4**	**29.9**

Table 2: BLEU scores obtained for different adaptation schemes of the CSTM for $P(\overline{t}_i|\overline{s}_{i-n+1}^{i-1}, \overline{t}_{i-n+1}^{i-1})$ in the middle part, and with model combination in the lower part. The notation n-code (+TED) emphasizes that for this system the baseline SMT system is *re-trained* with out-of-domain and in-domain data, while in all other cases the baseline system only uses out-of-domain data.

SMT system is entirely re-trained from scratch to integrate in-domain data (from word alignments to the large scale target language model), and all four CSTMs defined by (2) are adapted using the CLL criterion. This experiment shows that we can achieve slightly better performance by only adapting two CSTMs with the proposed objective function.

5. Related work

Most recent works in domain adaptation for SMT focuses on the modification of the sufficient statistics required by conventional discrete models [3, 4, 38], or on data selection [5, 6]. Our work owes much to recent contributions in discriminative training and tuning of SMT systems. While perceptron-based learning has been first introduced in [39, 40], margin-based algorithms such as MIRA [20, 21] are nowadays considered as more efficient to train Feature-Rich Translation systems. This property is especially relevant in our case, since we intend to learn a large set of parameters ($\boldsymbol{\theta}$). Another trend considers the optimization problem as ranking [41, 39, 22, 23]. Note that the ranking task corresponds to the integration of the CSTM that is actually used for N-best reranking. In this work, the proposed objective functions borrow from these two lines of research to both adapt the CSTM ($\boldsymbol{\theta}$) and tune its contribution ($\boldsymbol{\lambda}$) to the whole SMT system.

To the best of our knowledge, the most similar work on discriminative training or adaptation of neural network models is [12]. In this article, the authors propose to estimate the parameters of a neural network towards the expected BLEU score, while tuning $\boldsymbol{\lambda}$ by standard tools. Algorithm 1 is very similar to the optimization algorithm they describe, except that in our case, the feature weights $\boldsymbol{\lambda}$ are regularly updated for a better and tighter integration of the CSTM into the SMT system. Moreover, their proposed model only considers phrase pairs in isolation, while we use a probabilistic model of the joint distribution of sentence pairs. Expected BLEU training was also applied to recurrent neural network language model in [42].

In [13], the authors also introduce a ranking-type objective function that only aims to estimate word embeddings in a multitask-learning framework. Furthermore, [17] investigates the use of a large-margin criterion to train a recursive neural network for syntactic parsing. Interestingly, their model is also used to rerank N-best derivations generated by a conventional probabilistic context-free grammar. However, as showed by experimental results, the max-margin criterion alone is less adapted to machine translation. One explanation is that the N-best lists generated by the SMT system are not sufficiently diverse.

6. Conclusions

This paper has proposed and evaluated the use of discriminative criteria to adapt continuous space translation models. Instead of using a standard maximum likelihood method, the newly proposed algorithm discriminatively contrasts good and bad hypotheses from an N-best list produced by the baseline system into which the CSTM will be incorporated. A new adaptation method has been tested, consisting in jointly optimizing parameters from the neural network and from the SMT system so that the algorithm directly improves the system's overall quality. BLEU-based margins have also been included into these new loss functions and are proved to be useful. Our experiments consist in adapting out-of-domain CSTMs using a small quantity of in-domain parallel data, while keeping intact the out-of-domain baseline system. Our conclusions are two-fold. Firstly, we prove empirically the effectiveness of using discriminative criteria to adapt CSTMs, compared to the traditional maximum likelihood method. Secondly, our comparison shows that the pairwise ranking criterion is more suitable to Discriminative Reranking task in SMT than the max-margin approach, and that combining both criterion can deliver additional gains. In general, this work confirms the effective use of neural networks in Domain Adaptation for SMT systems.

For future work, we plan to combine our framework with other objective functions on N-best lists, such as *expected* BLEU [43]. We will also try an intensified use of the proposed algorithm by iteratively adding multiple feature functions into the SMT system; each model is trained using baseline system's N-best lists rescored with previously added models, in the hope that each model will capture complementary information and correct errors of the previous pass. Moreover, even though this work focuses on probabilistic n-gram translation models, our framework could be applied to any model structure [44, 18, 11] giving a score to each translation hypothesis.

7. References

[1] H. Daume III and D. Marcu, "Domain adaptation for statistical classifiers," *Journal of Artificial Intelligence Research*, vol. 26, pp. 101–126, 2006.

[2] J. Blitzer, "Domain adaptation of natural language processing systems," Ph.D. dissertation, University of Pennsylvania, 2008.

[3] G. Foster and R. Kuhn, "Mixture-model adaptation for SMT," in *Proceedings of the Second Workshop on Statistical Machine Translation*, Prague, Czech Republic, 2007, pp. 128–135.

[4] N. Bertoldi and M. Federico, "Domain adaptation for statistical machine translation with monolingual resources," in *Proceedings of the Fourth Workshop on Statistical Machine Translation*, Athens, Greece, 2009, pp. 182–189.

[5] A. Axelrod, X. He, and J. Gao, "Domain adaptation via pseudo in-domain data selection," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2011, pp. 355–362.

[6] R. Sennrich, "Perplexity minimization for translation model domain adaptation in statistical machine translation," in *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, 2012, pp. 539–549.

[7] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[8] H. Schwenk, M. R. Costa-jussa, and J. A. R. Fonollosa, "Smooth bilingual n-gram translation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Prague, Czech Republic, 2007, pp. 430–438.

[9] H.-S. Le, A. Allauzen, and F. Yvon, "Continuous space translation models with neural networks," in *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Montréal, Canada, 2012, pp. 39–48.

[10] Y. Hu, M. Auli, Q. Gao, and J. Gao, "Minimum translation modeling with recurrent neural networks," in *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, 2014, pp. 20–29.

[11] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[12] J. Gao, X. He, W.-t. Yih, and L. Deng, "Learning continuous phrase representations for translation modeling," in *Proc. ACL*, 2014.

[13] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

[14] H. Schwenk, "Continuous space language models," *Computer Speech and Language*, vol. 21, no. 3, pp. 492–518, July 2007.

[15] T. Mikolov, S. Kombrink, L. Burget, J. Cernocký, and S. Khudanpur, "Extensions of recurrent neural network language model," in *Proceedings of ICASSP*, 2011, pp. 5528–5531.

[16] H.-S. Le, I. Oparin, A. Allauzen, J.-L. Gauvain, and F. Yvon, "Structured output layer neural network language models for speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 21, no. 1, pp. 197–206, 2013.

[17] R. Socher, J. Bauer, C. D. Manning, and N. Andrew Y., "Parsing with compositional vector grammars," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, Sofia, Bulgaria, 2013, pp. 455–465.

[18] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Seattle, Washington, USA, 2013, pp. 1700–1709.

[19] J. Devlin, R. Zbib, Z. Huang, T. Lamar, R. Schwartz, and J. Makhoul, "Fast and robust neural network joint models for statistical machine translation," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Baltimore, Maryland, 2014, pp. 1370–1380.

[20] T. Watanabe, J. Suzuki, H. Tsukada, and H. Isozaki, "Online large-margin training for statistical machine translation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Citeseer, 2007.

[21] C. Cherry and G. Foster, "Batch tuning strategies for statistical machine translation," in *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, June 2012, pp. 427–436.

[22] M. Hopkins and J. May, "Tuning as ranking," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, Edinburgh, Scotland, UK., July 2011, pp. 1352–1362.

[23] P. Simianer, S. Riezler, and C. Dyer, "Joint feature selection in distributed stochastic learning for large-scale discriminative training in SMT," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, 2012, pp. 11–21.

[24] M. Federico, S. Stüker, L. Bentivogli, M. Paul, M. Cettolo, T. Herrmann, J. Niehues, and G. Moretti, "The IWSLT 2011 evaluation campaign on automatic talk translation," in *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. European Language Resources Association (ELRA), 2012.

[25] J. B. Mariño, R. E. Banchs, J. M. Crego, A. de Gispert, P. Lambert, J. A. Fonollosa, and M. R. Costa-Jussà, "N-gram-based machine translation," *Computational Linguistics*, vol. 32, no. 4, pp. 527–549, 2006.

[26] J. M. Crego and J. B. Mariño, "Improving statistical MT by coupling reordering and decoding," *Machine Translation*, vol. 20, no. 3, pp. 199–215, 2006.

[27] H.-S. Le, I. Oparin, A. Allauzen, J.-L. Gauvain, and F. Yvon, "Structured output layer neural network language model," in *Proceedings of ICASSP*, 2011, pp. 5524–5527.

[28] A. Mnih and G. E. Hinton, "A scalable hierarchical distributed language model," in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds., vol. 21, 2008, pp. 1081–1088.

[29] P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, "Class-based n-gram models of natural language," *Computational Linguistics*, vol. 18, no. 4, pp. 467–479, 1992.

[30] J. M. Crego, F. Yvon, and J. B. Mariño, "N-code: an open-source bilingual N-gram SMT toolkit," *Prague Bulletin of Mathematical Linguistics*, vol. 96, pp. 49–58, 2011.

[31] J. Niehues and A. Waibel, "Continuous space language models using restricted Boltzmann machines." in *Proceedings of International Workshop on Spoken Language Translation (IWSLT)*, Hong-Kong, China, 2012, pp. 164–170.

[32] A. Vaswani, Y. Zhao, V. Fossum, and D. Chiang, "Decoding with large-scale neural language models improves translation," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, Washington, USA, 2013, pp. 1387–1392.

[33] Y. Freund and R. E. Schapire, "Large margin classification using the perceptron algorithm," *Machine learning*, vol. 37, no. 3, pp. 277–296, 1999.

[34] R. McDonald, K. Crammer, and F. Pereira, "Online large-margin training of dependency parsers," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2005, pp. 91–98.

[35] M. Collins, "Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2002, pp. 1–8.

[36] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *in Proceedings of the 40th Annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[37] F. Casacuberta and E. Vidal, "Machine translation with inferred stochastic finite-state transducers," *Computational Linguistics*, vol. 30, no. 3, pp. 205–225, 2004.

[38] B. Chen, R. Kuhn, and G. Foster, "Vector space model for adaptation in statistical machine translation," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2013, pp. 1285–1293.

[39] L. Shen and A. K. Joshi, "Ranking and reranking with perceptron," *Machine Learning*, vol. 60, no. 1-3, pp. 73–96, 2005.

[40] P. Liang, A. Bouchard-Côté, D. Klein, and B. Taskar, "An end-to-end discriminative approach to machine translation," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, 2006, pp. 761–768.

[41] L. Shen, A. Sarkar, and F. J. Och, "Discriminative reranking for machine translation." in *HLT-NAACL*, 2004, pp. 177–184.

[42] M. Auli and J. Gao, "Decoder integration and expected bleu training for recurrent neural network language models," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, June 2014, pp. 136–142.

[43] J. Gao and X. He, "Training mrf-based phrase translation models using gradient ascent," in *Proceedings of NAACL-HLT*, 2013, pp. 450–459.

[44] M. Auli, M. Galley, C. Quirk, and G. Zweig, "Joint language and translation modeling with recurrent neural networks." in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2013, pp. 1044–1054.