A Deterministic Annealing-Based Training Algorithm For Statistical Machine Translation Models

Pascual Martínez Gómez*[1]*, Kei Hashimoto*[2]*,
***Yoshihiko Nankaku*[2]*, Keiichi Tokuda*[2]*, Germán Sanchis-Trilles*[1]**
(1) Instituto Tecnológico de Informática
Universidad Politécnica de Valencia
{pmartinez,gsanchis}@dsic.upv.es
(2) Tokuda & Lee Laboratory
Nagoya Institute of Technology
{bonanza,nankaku,tokuda}@sp.nitech.ac.jp

Abstract

This paper proposes the use of a Deterministic Annealing Expectation-Maximization (DAEM) algorithm to estimate the word-alignments involved in the statistical translation process. This approach is aimed to overcome the problem of the local maxima in complex alignment models, thus making unnecessary to iterate with previous and simpler ones.

Using the DAEM algorithm allows us to explore the power of highly expressive statistical alignment models without the experimental limitations of working with non-convex models, while, at the same time, observing consistent improvements in translation quality.

Experimental results show that, by using an appropriate temperature scheduling, equal or better estimations are obtained independently of the initial parameter estimates.

1 Introduction

Statistical Machine Translation (SMT) systems use mathematical models to describe the translation task and to estimate the probabilities involved in the process.

Brown et al. (1993) established the SMT grounds formulating the probability of translating a source sentence $\mathbf{f}$ into a target sentence $\mathbf{e}$, as

$$\hat{\mathbf{e}} = \underset{\mathbf{e}}{\operatorname{argmax}} \ \Pr(\mathbf{f} \mid \mathbf{e}) \cdot \Pr(\mathbf{e}) \qquad (1)$$

where $\Pr(\mathbf{e})$ stands for the *language model* and $\Pr(\mathbf{f}|\mathbf{e})$ is the *translation model*. The language model is usually based on n-grams, accounting mainly for the word-order with the purpose of avoiding ill-formed sentence $\mathbf{e}$, and the translation model accounts for the probability of the words of $\mathbf{e}$ being a good translation of the words of $\mathbf{f}$.

In word-to-word SMT, word-alignments were used to model the distribution $\Pr(\mathbf{f} \mid \mathbf{e})$. In this context, a hidden variable $\mathbf{a}$ is introduced to represent word-correspondences in a bilingual sentence pair. Introducing the alignment concept:

$$\Pr(\mathbf{f} \mid \mathbf{e}) = \sum_{\mathbf{a}} \Pr(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) \qquad (2)$$

In practice, however, the direct modelling of the posterior probability $\Pr(\mathbf{f} \mid \mathbf{e})$ has been widely adopted. To this purpose, different authors (Papineni et al., 1998; Och and Ney, 2002) propose the use of the so-called log-linear models, where the decision rule is given by the expression

$$\hat{\mathbf{e}} = \underset{\mathbf{e}}{\operatorname{argmax}} \sum_{m=1}^{M} \lambda_m h_m(\mathbf{f}, \mathbf{e}) \qquad (3)$$

where $h_m(\mathbf{x}, \mathbf{y})$ is a score function representing an important feature for the translation of $\mathbf{f}$ into $\mathbf{e}$, M is the number of models (or features) and λ_m are the weights of the log-linear combination.

In order to introduce context information, modern state-of-the-art SMT systems no longer implement a word-by-word translation model as described by Equation 2, but rather segment-to-segment. *Phrase-Based* (PB) SMT systems, which constitute the most popular implementation of log-linear SMT models, still rely heavily on word-alignment models in order to obtain these segments. Once the word alignments have been obtained, all bilingual phrases (i.e. subsequent word segments) coherent with the word alignment are extracted. Hence, the quality of the final translations produced by such systems still depends greatly on the quality of the word-alignment produced.

In order to obtain good estimations of the hidden alignment variable $\mathbf{a}$ in Equation 2, the log-likelihood expression derived from such equation is maximized by using, traditionally, the *Expectation-Maximization* (EM) algorithm (Dempster et al., 1977). From a given set of sample observations, EM algorithm has been used so far to maximize the log-likelihood of a given model that depends on unobserved (latent) variables. This is an iterative procedure consisting of two steps. The Expectation (E) step computes the expected value of the hidden variables and of the log-likelihood, and then the Maximization (M) step computes the parameter values that maximize the expected log-likelihood from the E step. The process is then repeated in a new iteration of the algorithm, using the parameter estimates from the previous iteration to obtain new estimates of the hidden variables, and so forth.

Since the EM algorithm uses at every iteration the parameter estimates from the previous one, such parameters need to be initialized for the first iteration. However, depending on this initialization, the EM algorithm may not converge to the global maximum if the solution space is not convex and the search is stuck in a local maximum of the log-likelihood surface. This problem is known as the *local maxima problem*.

Aware of the local maxima problem, Brown et al. (1993) proposed several models with increasing complexity, with the purpose of training them sequentially from the simplest to the most complex one. After performing a certain amount of iterations with the EM algorithm in the alignment model i, the estimated parameters are transferred conveniently to the alignment model $i+1$ with the purpose of achieving a better initialization yielding a better final estimation. In practice, the number of iterations performed by the EM algorithm for every model is set experimentally in order to maximize the score representing the quality of the automatic translations.

Transferring parameters between models during their sequential training is a useful technique but it can still prevent us from finding the global maximum in subsequent models (Fig. 1). Due to the different nature of the alignment models, estimates obtained from previous models may not be the best initial values for the EM algorithm to train the next model. Our intention in the present paper is to explore this scenario in detail.

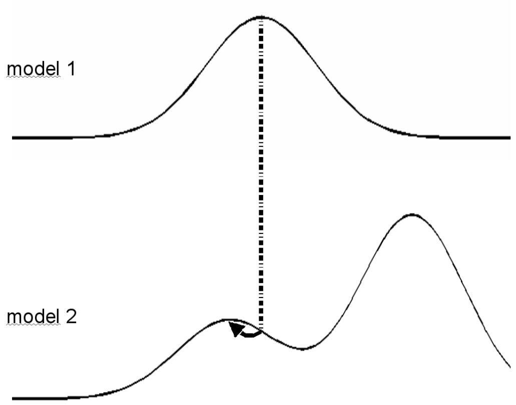

Figure 1: Inaccuracies during the transfer of parameters between models may occur.

Ueda and Nakano (1998) proposed a Deterministic Annealing version of the EM algorithm (DAEM), where the final estimation of the parameters are independent of the initial chosen values. DAEM has not been proved to solve theoretically the local maxima problem, but it has been used successfully in other applications where the maximization of functions with incomplete data is required (Park et al. (2005) applied DAEM for Image Segmentation, and Yohei et al. (2005) for Speaker and Speech Recognition) obtaining equal or better results than those achieved with EM.

This paper shows how the DAEM algorithm can be successfully applied to complex alignment models. Specifically, in this paper we will be applying DAEM to IBM Model 4. As results show, performing EM iterations with previous models is not necessary anymore.

The paper is organized as follows. Section 2 reviews the key features of IBM Model 4 and describes how DAEM may be applied for estimating parameters. Section 3 describes the SMT system in which we will be applying DAEM, while section 4 presents experimental results. Finally, a discussion is presented in the section 5 and future work is the final section.

2 The DAEM algorithm in SMT

2.1 IBM Model 4

Alignment Model 4 (Brown et al., 1993) is a complex model which is widely used in state-of-the-art SMT systems. They key idea behind such model can be expressed as follows: Given a sentence $\mathbf{e} = e_1, \ldots, e_i, \ldots, e_l$ of length $l = |\mathbf{e}|$ from a target language, and a source sentence $\mathbf{f} = f_1, \ldots, f_j, \ldots, f_m$ of length $m = |\mathbf{f}|$ from a source language, we first choose a source word e_i and the number ϕ_i of source words that e_i is going to be aligned to, according to the fertility probability distribution $n(\phi|e_i)$. A target word can be aligned to zero, one or more source words, and this property is called *fertility*.

Once e_i and ϕ_i have been chosen, select $f_j^{j+\phi_i} = f_j, \ldots, f_{j+\phi_i}$ following a certain probability distribution $t(f_j^{j+\phi_i}|e_i)$ modelling the probability of translating $f_j^{j+\phi_i}$ into e_i. With the help of the functions $\mathcal{A}$ and $\mathcal{B}$ that map words from the source and the target language into a small number of classes, we have to decide the final position in the source sentence for every word f_j. Such a decision is determined by the distortion probability distributions $d_1(\Delta j|\mathcal{A},\mathcal{B})$ and $d_{>1}(\Delta j|\mathcal{B})$. $d_1(\Delta j|\mathcal{A},\mathcal{B})$ is used to place the word $f \in f_j^{j+\phi_i}$ whose position in $\mathbf{f}$ is smallest and $d_{>1}(\Delta j|\mathcal{B})$ to place the rest of the words in $f_j^{j+\phi_i}$.

Taking into consideration extraneous words appearing in the source sentence, the expression described by IBM Model 4 for the probability of a sentence $\mathbf{f}$ and an alignment $\mathbf{a}$ given a target sentence $\mathbf{e}$ is

$$\Pr(\mathbf{f}, \mathbf{a}|\mathbf{e}) = \sum_{(\tau,\pi)\in\langle\mathbf{f},\mathbf{a}\rangle} \Pr(\tau, \pi|\mathbf{e}) \qquad (4)$$

In this equation, a tablet τ is the set of words that can be generated by a given target word e_i and the distortions π are the final positions of each subset of words in τ in the source sentence.

The probability of a tablet and a set of distortions given a target sentence can be expressed as a product of probabilities:

$$\begin{aligned}\Pr(\tau, \pi|\mathbf{e}) &= n_0\left(\phi_0|\sum_{i=1}^{l}\phi_i\right) \\ &\quad \prod_{i=1}^{l} n(\phi_i|e_i) \prod_{i=0}^{l}\prod_{k=1}^{\phi_i} t(\tau_{ik}|e_i) \\ &\quad \frac{1}{\phi_0!}\prod_{i=1}^{l}\prod_{k=1}^{\phi_i} p_{ik}(\pi_{ik}) \qquad (5)\end{aligned}$$

where $n_0(\phi_0|m')$ is a function depending on the parameters p_0 and $p_1 = 1 - p_0$ describing the amount of extraneous words appearing in the source sentence.

$$n_0(\phi_0|m') = \binom{m' - \phi_0}{\phi_0} p_0^{m'-2\phi_0} p_1^{\phi_0} \qquad (6)$$

Here, p_1 is the probability that a word f requires an extraneous word not connected with any word of $\mathbf{e}$. The distribution probability $p(\pi)$ is a function of the distortion probabilities, used here to keep notation simple. Refer to (Brown et al., 1993) for further details. There are many configurations of tablets and distortions that give as a result the same source sentence $\mathbf{f}$ aligned by means of the same $\mathbf{a}$. Note that the summation takes into consideration every possible configuration of (τ, π) that is consistent with the pair $\langle\mathbf{f}, \mathbf{a}\rangle$.

Using standard methods for function maximization at Equation 2, a generic expression to re-estimate parameters is obtained:

$$p(\omega; \mathbf{f}, \mathbf{e}) = \xi^{-1} \sum_{\mathbf{a}} \Pr(\mathbf{f}, \mathbf{a}|\mathbf{e}) c(\omega; \mathbf{a}, \mathbf{f}, \mathbf{e}) \qquad (7)$$

where ξ^{-1} acts as a reminder that the probabilities have to be normalized. The generalized probability distribution $p(\omega; \mathbf{f}, \mathbf{e})$ instantiates to $n_0(\phi_0|m')$, $n(\phi|e)$, $t(f|e)$, $d_1(\Delta j|\mathcal{A},\mathcal{B})$ and $d_{>1}(\Delta j|\mathcal{B})$ for IBM Model 4 parameter estimation. Here, $c(\omega; \mathbf{a}, \mathbf{f}, \mathbf{e})$ are generic counters that count the number of times that the event ω occurs in the pair $(\mathbf{f}, \mathbf{e})$ with the alignment $\mathbf{a}$. Examples of the counters are

$$c(f|e; \mathbf{a}, \mathbf{f}, \mathbf{e}) = \sum_{j=1}^{m} \delta(f, f_j)\delta(e, e_{a_j}) \qquad (8)$$

is the number of times that the word f is related to word e by alignment $\mathbf{a}$ in the pair $(\mathbf{f}, \mathbf{e})$, and

$$c(\phi|e; \mathbf{a}, \mathbf{f}, \mathbf{e}) = \sum_{i=1}^{l} \delta(\phi, \phi_i)\delta(e, e_i) \qquad (9)$$

is the number of times that the word e has a fertility of ϕ words by alignment $\mathbf{a}$ in the pair $(\mathbf{f}, \mathbf{e})$.

Traditionally, the EM algorithm has been used to estimate the parameters involved in the translation process, with the risk associated to the local maxima problem. DAEM will be used in the parameter estimation aiming to overcome such a problem.

2.2 DAEM

The Deterministic Annealing EM algorithm (DAEM) (Ueda and Nakano, 1998) helps to overcome the problem of the local maxima, reformulating the maximization of the likelihood into the minimization of the *free energy*, a concept extracted from thermodynamics.

The idea behind such a procedure is to parametrize the objective function defining the hyper-surface that has to be explored, so that for high values of the temperature $\frac{1}{\beta}$, the curves are smooth enough to allow us to find *safely* the global maximum using the traditional EM algorithm.

Each time the temperature is decreased, regular EM iterations are performed and new parameter estimates are computed. As the temperature decreases iteratively, the surface of the free energy becomes more and more similar to the likelihood. The final value of the temperature makes the expression of the free-energy to be equal to the likelihood.

The free energy is defined as an effective cost function that depends on the temperature:

$$F_\beta(\Theta) = -\frac{1}{\beta} \log \sum_{\chi_{mis}} p(\chi_{obs}, \chi_{mis}; \Theta)^\beta \quad (10)$$

where Θ are the parameters of the density function, χ_{obs} and χ_{mis} are observable and unobservable data respectively. In the following subsection, it will be showed how to apply easily the DAEM algorithm to statistical models where the EM algorithm was originally used, through an example on IBM Model 4.

2.3 DAEM algorithm for IBM Model 4

In the model studied, missing variables are the alignments $\mathbf{a}$, and pairs (τ, π). The observed variables during training time are every bilingual pair of sentences $(\mathbf{f}, \mathbf{a})$. Using the expression of the free energy with the likelihood of the translation model, we obtain the following expression

$$F_\beta(n, t, d) = -\frac{1}{\beta} \log \sum_{\mathbf{a}} \sum_{(\tau,\pi)\in\langle\mathbf{f},\mathbf{a}\rangle} \Pr(\tau, \pi|\mathbf{e})^\beta \quad (11)$$

Unlike the expression of the likelihood, Equation (11) has to be minimized. Adding a Lagrangian multiplier (with positive sign) for every constraint and setting the partial derivatives of the auxiliary function to zero, generalized parameter re-estimation expressions parametrized by β arise:

$$\begin{aligned} p(\omega; \beta, \mathbf{f}, \mathbf{e}) \;=\; & \xi^{-1} \sum_{\mathbf{a}} n_0 \left(\phi_0 | \sum_{i=1}^{l} \phi_i \right)^\beta \\ & \prod_{i=1}^{l} n(\phi_i|e_i)^\beta \phi_i! \prod_{j=1}^{m} t(f_j|e_{a_j})^\beta \\ & \prod_{j:a_j\neq 0} d_{a_j}(\pi_{a_j})^\beta c(\omega; \mathbf{a}, \mathbf{f}, \mathbf{e}) \end{aligned} \quad (12)$$

where the counters $c(\omega; \mathbf{a}, \mathbf{f}, \mathbf{e})$ are the same as in the EM algorithm. More compactly,

$$p(\omega; \beta, \mathbf{f}, \mathbf{e}) = \xi^{-1} \Pr(\mathbf{f}, \mathbf{a}|\mathrm{e})^\beta c(\omega; \mathbf{a}, \mathbf{f}, \mathbf{e}) \quad (13)$$

where, ξ^{-1} is again a reminder that the probabilities have to be properly normalized.

The proposed DAEM algorithm to train IBM Model 4 is as follows:

1. Initialize uniformly and different from zero the parameters $p(\omega)$: $n(\phi|e)$, $t(f|e)$, p_0, p_1, $d_1(\Delta j|\mathcal{A}, \mathcal{B})$ and $d_{>1}(\Delta j|\mathcal{A})$.
2. For $0 < \beta << 1$ to $\beta = 1$, compute

$$\tilde{p}(\omega; \beta) = \sum_{s=1}^{S} p(\omega; \beta, \mathbf{f}^s, \mathbf{e}^s) \quad (14)$$

   using the EM algorithm with the desired amount of iterations, with a corpus of S pair of sentences.

2.4 Scheduling

The initial idea behind the DAEM approach is to smooth the hyper-surface representing the cost function with a low value of β. Starting DAEM iterations with a positive value of β close to zero is intended to be equivalent to a minimization of a concave function. At the first step of the DAEM algorithm, parameters are initialized uniformly with values different from zero. Then, EM is used regularly to maximize the objective function parametrized by the starting β. The estimates obtained by the last iteration of the EM for β are used for the first iteration of EM for the next value of β, and so on.

This process is repeated iteratively with increasing values of β up to one. Minimizing Equation (10) for $\beta = 1$ is equivalent to maximizing the traditional likelihood function, with the advantage of having excellent initial values for the EM algorithm during the last iteration of DAEM.

Since at least one EM iteration must be completed for every step of β, over-training may occur before the temperature $\frac{1}{\beta}$ achieves a value of one. For this reason, an early stopping of the decrease of the temperature $\frac{1}{\beta}$ may result in better estimations (Fig. 2).

Over-training is a difficult problem to solve using theoretical methods so that some experimentation is required to avoid it. When working with DAEM, the following considerations must be taken into account:

- Selecting a β positive and close to zero as initial value for the DAEM algorithm.
- Selecting an appropriate step size for β. Smaller steps provide a higher accuracy but increases the computational cost.
- Selecting the number of EM iterations at every value of the temperature.

3 Experimental Setup

We performed our experiments on the Europarl corpus (Koehn, 2005), which is a corpus widely used in SMT and that has been used in several MT evaluation campaigns. We performed our experiments on the partition established for the Workshop on Statistical Machine Translation of the ACL 2008. The Europarl corpus was extracted from the proceedings of the European Parliament, and is divided into three separated sets: one for training, one for development and one for test.

For our experiments, we focused on the French→English translation task. The characteristics of this task can be seen in Table 1. As baseline for our experiments, the Moses-toolkit (Koehn and others, 2007) was used in its default setup, as proposed in the WMT08 task. This includes regular EM training of several low-order word-alignment models, in order to obtain good initial parameter estimates for training Model 4 and obtaining convenient segmentations of the training sentence pairs. Specifically, we performed 5 iterations of IBM Model 1, 5 HMM iterations, 3 iterations of IBM Model 3 and 3 more iterations of IBM Model 4.

Table 1: Characteristics of Europarl corpus French-English. Develop. stands for Development, OoV for "Out of Vocabulary" words, K for thousands of elements and M for millions of elements. Data statistics were collected after tokenizing, lowercasing and filtering out long sentences.

		Fr	En
Training	Sentences	948K	
	Run. words	20.7M	19.5M
	Avg. leng.	21.8	20.5
	Vocabulary	98K	81K
Develop.	Sentences	2000	
	Run. words	63K	59K
	Avg. leng.	31.6	29.4
	OoV	60	67
Test	Sentences	2000	
	Run. words	62K	58K
	Avg. leng.	31.1	29.0
	OoV	69	75

For our system, GIZA++ (Och and Ney, 2003) was conveniently modified to add the DAEM loop with a linear progression of the temperature: $\beta = 0.001, 0.02, 0.04, \ldots, 1.0$ on the IBM Model 4 using the same hill-climbing technique implemented in GIZA++, performing three iterations of the EM algorithm for every value of β.

Once the bilingual phrase pairs have been extracted and probabilities have been associated to them, the weights of the log-linear models were optimized for the development set using the MERT procedure (Och, 2003).

We measured the quality of the translations provided by both systems with two scores, BLEU and TER. BLEU (Papineni et al., 2002) computes the precision of unigrams, bigrams, trigrams and 4-grams with a penalty for too short sentences, and the Translation Error Rate (TER) criterion (Snover et al., 2006) computes the minimum number of editions (substitutions, insertions and deletions) needed to convert the translated sentence into the reference sentence, including shift of words as edition operations, and normalizes with the average number of reference words.

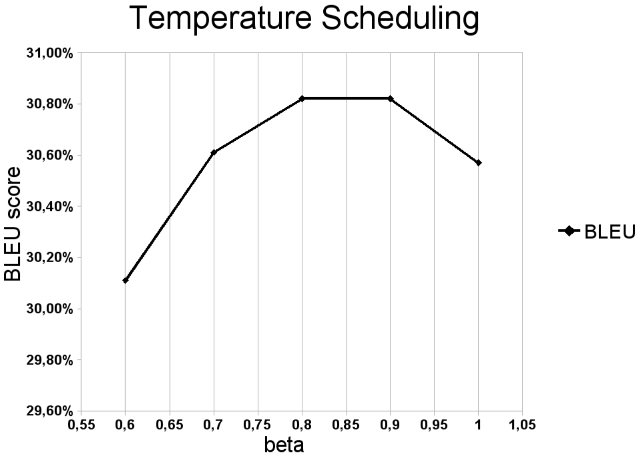

Figure 2: Scheduling on a small corpus consisting of 200K sentence pairs. Beta values were a linear progression $\beta = 0.1, \ldots, 1.0$. Better BLEU score may be achieved stopping DAEM earlier and making smaller β-steps.

4 Experimental Results

Preliminary experiments with a reduced Europarl training corpus of 200K sentence pairs were done, stopping at different values of β (Fig. 2). The best results obtained with this reduced version of the Europarl were around a value of $\beta = 0.8$. Specifically, in our experimental environment, the value $\beta = 0.76$ yielded the best results in terms of BLEU and TER in the full Europarl corpus. As mentioned, we performed 3 EM iterations with IBM Model 4, dropping out the initialization with other models.

The pairwise BLEU and TER improvement intervals (see Table 4) computed by the bootstrapping technique (Koehn, 2004) at a 95% confidence level show that improvements obtained by applying DAEM are statistically significant. For computing these intervals, 10.000 bootstrap repetitions were performed. Specifically, using the DAEM algorithm leads to an increase of BLEU in the range of $[0.04, 0.57]$, while obtaining a decrease on the error rate (TER) in the range of $[-0.56, -0.04]$. Improvements in these two measures with a different nature support our thesis that better results can be obtained by using DAEM than those obtained by using EM.

Table 2: Using the DAEM algorithm to train IBM Model 4, better scores were obtained.

	baseline	DAEM	Pairwise improv. interval
BLEU	32.46	$\mathbf{32.75}$	$[\mathbf{0.04}, \mathbf{0.57}]$
TER	52.39	$\mathbf{52.09}$	$[\mathbf{-0.56}, \mathbf{-0.04}]$

5 Conclusions

Statistical machine translation systems require complex models with a large amount of parameters and DAEM provides promising results on this field.

In the case of Statistical Machine Translation, using the DAEM approach in function maximization (minimization) problems related to the estimation of statistical alignments, improves the overall quality of the translations obtained, as measured by BLEU and TER, when compared to those obtained by the traditional EM algorithm.

The fact that the estimations obtained by the DAEM algorithm are independent of the initial parameter values adds a special focus of interest. By training IBM Model 4 with DAEM instead of regular EM, it is more likely to obtain those parameter estimates for which the objective function presents a global maximum.

Furthermore, we discarded all the previous models while using DAEM algorithm on IBM Model 4, since their parameter estimations were not longer necessary to initialize the parameters in subsequent models. For this reason, we do not face the risk of losing the global maximum while transferring the parameters from one model to another and we may have a more realistic perspective of the potential of the studied model.

The ease of implementation of the DAEM algorithm on already-implemented models makes it worth of consideration. Moreover, by using DAEM we can focus on the design of new complex models without being forced to provide its corresponding simplifications with a gradual increase of complexity, trusting in DAEM to perform the minimization of the free energy function associated to the expression of the likelihood. Hence, DAEM can also be a good method to objectively compare single models with a different nature, rather than compare a set of sequential models with an arbitrary number of EM iterations per model.

6 Future Work

Future work involves applying DAEM with IBM Models 5 and 6. Some authors propose the use of a Hidden Markov Model instead of Model 4. This model is also a good candidate to test the performance of the DAEM algorithm because its results are close in quality to those obtained by IBM Model 4 and its computational cost at every EM iteration is much lower.

Scheduling is a practical aspect that must be explored with much more detail. Smaller steps on the temperature may help to select a better stop value of β and we will explore this possibility.

EM iterations in Model 4 are much more time-demanding than those performed in previous models. Since DAEM performs EM iterations at every decrease of temperature, the entire training time is much higher than the traditional one. Using an appropriate scheduling configuration to select a convenient number of temperature steps and its corresponding number of EM iterations, training can be performed in a more efficient manner.

It can also be interesting to study the performance of a geometrical progression of the temperature and eventually reduce the number of EM iterations for low temperature levels, since it can drastically reduce the current amount of time required for training.

Due to the fact that the present work is not dependent on the used languages, further experiments with different pairs will be carried out, as we expect to have similar improvements on the quality of the translations. In addition, we also plan on analysing the scalability of the approach proposed in this paper when dealing with bigger corpora, such as the NIST corpus or the more recent versions of the Europarl corpus.

In this paper, results were reported in terms of BLEU and TER to show the actual improvements in terms of translation quality. Nevertheless, such scores are computed after employing the alignments within the Moses-pipeline, which makes use of the alignments only to extract phrases. Hence, the impact of this method on the training of the statistical alignment models may fade. For this reason, we intend to conduct a deeper analysis of the alignment quality produced by DAEM, by studying comparable results in terms of log-likelihood and Alignment Error Rate.

Acknowledgement

Work supported by the Generalitat Valenciana under grant Prometeo/2009/014 and GV/2010/067, by the Spanish MICINN under the iTrans2 project (TIN2009-14511) and by the Spanish MITyC under the erudito.com (TSI-020110-2009-439) project.

References

P.F. Brown, S.A. Della Pietra, V.J. Della Pietra, and R.L. Mercer. 1993. The mathematics of machine translation. In *Computational Linguistics*, volume 19, pages 263–311, June.

A.P. Dempster, N.M. Laird, and D.B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*.

P. Koehn et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. of the ACL'07*.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation.

P. Koehn. 2005. EuroParl: A parallel corpus for statistical machine translation. In *Proc. of the MT Summit X*, pages 79–86.

F. Och and H. Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proc. of the ACL'02*, pages 295–302.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

F.J. Och. 2003. Minimum error rate training for statistical machine translation. In *Proc. of ACL'03*, pages 160–167.

K. Papineni, S. Roukos, and T. Ward. 1998. Maximum likelihood and discriminative training of direct translation models. In *Proc. of ICASSP'98*, pages 189–192.

K. Papineni, S. Roukos, T. Ward, and W.J. Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proc. of ACL'02*.

Jonghyun Park, Wanhyun Cho, and Soonyoung Park, 2005. *Deterministic Annealing EM and Its Application in Natural Image Segmentation*, volume 3314 of *Lecture Notes in Computer Science*, pages 639–644. SpringerLink.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, Cambridge, Massachusetts, USA.

Naonori Ueda and Ryohei Nakano. 1998. Deterministic annealing EM algorithm. *Neural Networks 11*, pages 271–282.

Itaya Yohei, Zen Heiga, Nankaku Yoshihiko, Miyajima Chiyomi, Tokuda Keiichi, and Kitamura Tadashi. 2005. Deterministic annealing EM algorithm in acoustic modeling for speaker and speech recognition. *IEICE Transactions on Information and Systems*.