Improving translation quality stability using Bayesian predictive adaptation

Germán Sanchis-Trilles, Francisco Casacuberta

Pattern Recognition and Human Language Technologies group
Universitat Politècnica de València
46022 Valencia, Spain

Abstract

We introduce a Bayesian approach for the adaptation of the log-linear weights present in state-of-the-art statistical machine translation systems. Typically, these weights are estimated by optimising a given translation quality criterion, taking only into account a certain set of development data (e.g., the adaptation data). In this article, we show that the Bayesian framework provides appropriate estimates of such weights in conditions where adaptation data is scarce. The theoretical framework is presented, alongside with a thorough experimentation and comparison with other weight estimation methods. We provide a comparison of different sampling strategies, including an effective heuristic strategy and a theoretically sound Markov chain Monte-Carlo algorithm. Experimental results show that Bayesian predictive adaptation (BPA) outperforms the re-estimation from scratch in conditions where adaptation data is scarce. Further analysis reveals that the improvements obtained are due to the greater stability of the estimation procedure. In addition, the proposed BPA framework has a much lower computational cost than raw re-estimation.

Keywords: Bayesian methods, adaptation, natural language processing, machine translation

1. Introduction

Adaptation has become a very popular issue in natural language processing [1, 2, 3], and more specifically in *statistical machine translation* (SMT) [4]. Typically, the adaptation problem arises when two very different sets of training data are available, yielding two different sets of model parameters. The first set of data, the training data $\mathcal{T}$ (e.g., obtained from the European Parliament or the United Nations) is often very large and rather generic in domain. The second set of data, the adaptation data $\mathcal{A}$, belongs to the specific task of interest, such

Email address: gsanchis|fcn@prhlt.upv.es (Germán Sanchis-Trilles, Francisco Casacuberta)

as printer manuals or medical diagnoses, and is usually overwhelmingly smaller than $\mathcal{T}$. Then, the challenge is to modify the SMT system appropriately by taking into consideration both $\mathcal{T}$ and $\mathcal{A}$: on the one hand, $\mathcal{T}$ is should provide robustness in the estimation of the model parameters $\boldsymbol{\theta}$, and on the other hand $\mathcal{A}$ should introduce a certain bias towards the specific task.

This definition of adaptation is specially appropriate for the Bayesian learning paradigm, where the model parameters $\boldsymbol{\theta}$ are treated as (hidden) random variables governed by some kind of a priori distribution $p(\boldsymbol{\theta})$. This distribution represents our prior knowledge about what values for $\boldsymbol{\theta}$ should be good estimates. Estimating $p(\boldsymbol{\theta})$ by using a sufficiently large collection of data $\mathcal{T}$ allows us to obtain a canonical model with parameters $\boldsymbol{\theta}_{\mathcal{T}}$, and it can be assumed that such estimation is a robust estimation. As further evidence arrives in form of adaptation data $\mathcal{A}$, that such estimations are revised so that they reflect the newly arrived data. Considering $\mathcal{A}$ within the Bayesian predictive distribution leads precisely to a scenario in which the decision regarding the output sentence includes a bias towards $\mathcal{A}$, but is still guided by $p(\boldsymbol{\theta}_{\mathcal{T}})$ (i.e., the prior distribution given $\mathcal{T}$). Hence, under the *Bayesian predictive adaptation* (BPA) framework, the final translation is not computed by considering only the topic-specific data (i.e., $\mathcal{A}$), which could lead to over-trained estimations of $\boldsymbol{\theta}$: if the amount of data available is small, the parameter prior $p(\boldsymbol{\theta})$ will compensate this, providing robustness [5]. However, the effect of this prior knowledge fades when incorporating further evidence, until a point in which the contribution of the parameter prior towards the complete model distribution is negligible. In addition, the Bayesian learning paradigm does not attempt to obtain a single best point estimate of $\boldsymbol{\theta}$, but rather relies on considering all possible parameter values, allowing uncertainty regarding what the best estimations of such parameters might be. In this paper, we focus on the Bayesian adaptation of the weights of the log-linear combination of features present in state-of-the-art SMT systems. Even though these weights are not very numerous (generally in the range of 10 or 20), providing the system with appropriate estimates for these weights is critical [6].

The rest of this paper is structured as follows: the related literature is reviewed in Section 2. The formal derivation of Bayesian predictive adaptation for SMT is presented in Section 3. Since the equation obtained is very costly to apply in practise, different sampling strategies are presented in Section 4. The experiments performed are detailed together with their results and the related analysis in Section 5. Finally, conclusions are presented in Section 6.

2. Related work

Adaptation in SMT is a research field that has been receiving increasing attention. Following the ideas in [1], one of the first works was performed in [7], where the authors added cache language and translation models to an interactive machine translation system. In [3], different ways to combine the available data belonging to two different sources were studied. The work in

[8] explores alignment model mixtures as a way of performing topic adaptation. Other authors [9, 10], have proposed the use of information retrieval and clustering techniques in order to extract the sub-domains of a large corpus, and [11, 12] proposed to select as training data only those sentences which can be considered topic-specific. Corpus weighting strategies were analysed in [13], and instance weighting techniques were applied in [14] in order to weight out-of-domain phrase pairs. Recently, sequential Bayesian methods were applied with the purpose of adapting the word alignments present in most state-of-the-art SMT systems [15]. In such work, the authors confront the problem of adapting the probabilities of the single-word models that are used for phrase extraction. In contrast, in this work we attempt to adapt the final translation model directly. Note that none of these works confront the problem of adapting the log-linear weights $\boldsymbol{\lambda}$ of the SMT system, but rather attempt to adapt either the underlying word alignments or the final translation model features $\mathbf{h}$, and comparison with such strategies is not suitable. Hence, re-estimating $\boldsymbol{\lambda}$ from scratch is, to the best of our knowledge, the most common approach when adapting $\boldsymbol{\lambda}$. This work intends to fill this gap, and can be seen as complementary to the adaptation approaches cited above.

Although only recently applied to SMT, Bayesian adaptation has been successfully applied in other natural language processing areas, such as speech recognition [2]. In fact, work done in this direction is very broad, covering both batch [16] and online adaptation [17]. Variational Bayes approaches have also been studied [18], which attempt to find a lower bound to approximate the intractable marginal likelihood, yielding point estimates of the model parameters. Alternatively, BPA attempts to approximate the marginal likelihood directly by sampling from the posterior distribution, and usually leads to more robust estimates [16].

With respect to BPA in SMT, to our knowledge the only work published as of yet in this direction is [19]. In that article, only the idea was introduced, together with preliminary experiments. Here, such preliminary work is widely extended both in depth and in range:

- Bayesian predictive adaptation is presented as an appropriate formal framework for model adaptation in SMT.
- Positive results concerning the adaptation of scaling factors are presented, for a standard adaptation task [20] with four different domains.
- Comparison with different $\boldsymbol{\lambda}$ re-estimation strategies, such as Minimum Error Rate Training (MERT) [21], batch Margin Infused Relaxed Algorithm (MIRA) [22] and Pairwise Ranking Optimisation (PRO) [23].
- Different sampling strategies are compared within BPA.
- Computational cost comparison among the methods presented.
- As a derived contribution, we also perform an in-depth analysis of the stability of the most common optimisation algorithms used in SMT, as a

function of development set size. To the best of our knowledge, no such study has been published as of yet.

3. Bayesian predictive adaptation for SMT

We first introduce the typical formulation of SMT [4]. In state-of-the-art SMT systems, it is quite common to have a log-linear combination of features $\mathbf{h}$, weighted by scaling factors $\boldsymbol{\lambda}$. Then, the probability of the output sentence $\mathbf{e}$ given the input sentence $\mathbf{f}$ is computed as[1]

$$p(\mathbf{e} \mid \mathbf{f}) = \frac{\exp \sum_m \lambda_m h_m(\mathbf{f}, \mathbf{e})}{\sum_{\mathbf{e}'} \exp \sum_m \lambda_m h_m(\mathbf{f}, \mathbf{e}')}, \tag{1}$$

and the decision rule is given by the expression

$$\hat{\mathbf{e}} = \operatorname*{argmax}_{\mathbf{e}} \sum_m \lambda_m h_m(\mathbf{f}, \mathbf{e}). \tag{2}$$

Typically, the weights $\boldsymbol{\lambda}$ of the log-linear combination are estimated on a development set by means of error-driven algorithms such as MERT [21] or MIRA [22], which have proven to provide good estimates if the amount of data available is sufficient and the characteristics of the data to be translated match approximately those of the development set. However, if either of these two premises are not fulfilled, over-fitting to the specific characteristics of the development set occurs and such algorithms fail to provide appropriate estimates [11, 6].

In this article, we propose to reformulate the decision rule whenever the following conditions are met:

- A development set for a given "*old*" domain is available, or a canonical estimation of $\boldsymbol{\lambda}$ is readily available.
- The text to be translated belongs to a different "*new*" domain.
- A small set of development data is available for the *new* domain, but such set is insufficient for a proper estimation of $\boldsymbol{\lambda}$.

The meaning of "sufficient" in this context depends mainly on the target domain, but also on the estimation method to be used. This will be analysed more in depth in the experiments section.

Under the circumstances described above, we consider *adapting* $\boldsymbol{\lambda}$, instead of performing a full re-estimation. For this purpose, we propose the use of the Bayesian paradigm [24], in which parameters are viewed as random variables with some kind of underlying distribution. Considering $\mathcal{T}$ as the training data,

[1] For readability purposes, we directly instantiate the model parameters $\boldsymbol{\theta}$ to the parameters we intend to adapt, i.e., $\boldsymbol{\lambda}$.

and $\mathcal{A}$ as an additional adaptation set, Equation 1 is rewritten by means of the predictive distribution as

$$p(\mathbf{e} \mid \mathbf{f}; \mathcal{T}, \mathcal{A}) = \int p(\mathbf{e}, \boldsymbol{\lambda} \mid \mathbf{f}; \mathcal{T}, \mathcal{A}) d\boldsymbol{\lambda} \quad (3)$$

$$\approx \int p(\boldsymbol{\lambda} \mid \mathcal{T}, \mathcal{A})\ p(\mathbf{e} \mid \mathbf{f}, \boldsymbol{\lambda}) d\boldsymbol{\lambda}. \quad (4)$$

From Equation 3 to Equation 4 it has been assumed that the probability of the output sentence $\mathbf{e}$ does not depend on $\mathcal{A}$ and $\mathcal{T}$, whenever the model parameters $\boldsymbol{\lambda}$ are known. It has also been assumed that $\boldsymbol{\lambda}$ is independent from the actual input sentence $\mathbf{f}$. Such simplifications lead to a decomposition of the integral in two parts: the first one, $p(\boldsymbol{\lambda} \mid \mathcal{T}, \mathcal{A})$, will assess how good the model parameters are, and the second one, $p(\mathbf{e} \mid \mathbf{f}, \boldsymbol{\lambda})$, will account for the quality of the translation $\mathbf{e}$ given $\boldsymbol{\lambda}$. The integral will force the model to take into account all possible parameter values, although the parameter prior will bias the final distribution towards our prior knowledge.

Operating with the probability of $\boldsymbol{\lambda}$, we obtain:

$$p(\boldsymbol{\lambda} \mid \mathcal{T}, \mathcal{A}) = \frac{p(\mathcal{A} \mid \boldsymbol{\lambda}; \mathcal{T})\ p(\boldsymbol{\lambda} \mid \mathcal{T})}{\int p(\mathcal{A} \mid \boldsymbol{\lambda}'; \mathcal{T})\ p(\boldsymbol{\lambda}' \mid \mathcal{T})\ d\boldsymbol{\lambda}'}. \quad (5)$$

In order to simplify Equation 5, and focusing on the probability of the adaptation data $\mathcal{A}$, of a given size $|\mathcal{A}|$, we obtain:

$$p(\mathcal{A} \mid \boldsymbol{\lambda}; \mathcal{T}) \approx p(\mathcal{A} \mid \boldsymbol{\lambda}) = \prod_{a=1}^{|\mathcal{A}|} p(\mathbf{f}_a \mid \boldsymbol{\lambda})\ p(\mathbf{e}_a \mid \mathbf{f}_a, \boldsymbol{\lambda}), \quad (6)$$

where the probability of $\mathcal{A}$ has been assumed to be independent of $\mathcal{T}$, given that $\boldsymbol{\lambda}$ is known, and has been modelled as the probability of each bilingual sample $(\mathbf{f}_a, \mathbf{e}_a) \in \mathcal{A}$ being generated independently by a given translation model.

For modelling the prior over the model parameters, i.e., $p(\boldsymbol{\lambda} \mid \mathcal{T})$, we will assume that $\boldsymbol{\lambda}$ follows a normal distribution centred on $\boldsymbol{\lambda}_{\mathcal{T}}$, i.e., the parameter values estimated on the training data ($\mathcal{T}$), and with a diagonal covariance matrix $I \cdot \sigma_{\mathcal{T}}$ with variance σ_T bounded for all parameters, yielding

$$p(\mathbf{e}|\mathbf{f}; \mathcal{T}, \mathcal{A}) \approx \mathcal{Z} \int p(\mathcal{A} \mid \boldsymbol{\lambda}; \mathcal{T})\ p(\boldsymbol{\lambda} \mid \mathcal{T})\ p(\mathbf{e} \mid \mathbf{f}, \boldsymbol{\lambda})\ d\boldsymbol{\lambda}$$

$$\approx \mathcal{Z} \int \prod_{a=1}^{|\mathcal{A}|} p(\mathbf{e}_a|\mathbf{f}_a, \boldsymbol{\lambda}) \mathcal{N}(\boldsymbol{\lambda}; \boldsymbol{\lambda}_{\mathcal{T}}, I \cdot \sigma_{\mathcal{T}}) p(\mathbf{e}|\mathbf{f}, \boldsymbol{\lambda})\ d\boldsymbol{\lambda}. \quad (7)$$

$\mathcal{Z}$ is the normalisation constant ensuring that $p(\mathbf{e} \mid \mathbf{f}; \mathcal{T}, \mathcal{A})$ defines a probability distribution. The term $p(\mathbf{f}_a \mid \boldsymbol{\lambda})$ present in Equation 6 can be simplified if $p(\mathcal{A} \mid \boldsymbol{\lambda}; \mathcal{T})$ is plugged into Equation 5 and if $\mathbf{f}_a$ can be assumed independent of $\boldsymbol{\lambda}$.

Plugging in the log-linear model described in Equation 1:

$$p(\mathbf{e} \mid \mathbf{f}; \mathcal{T}, \mathcal{A}) \propto \mathcal{Z} \int \prod_{a=1}^{|\mathcal{A}|} \frac{\exp \sum_m \lambda_m h_m(\mathbf{f}_a, \mathbf{e}_a)}{\sum_{\mathbf{e}'} \exp \sum_m \lambda_m h_m(\mathbf{f}_a, \mathbf{e}')} \exp\left\{-\frac{||\boldsymbol{\lambda} - \boldsymbol{\lambda}_{\mathcal{T}}||^2}{2\sigma_{\mathcal{T}}}\right\} \frac{\exp \sum_m \lambda_m h_m(\mathbf{f}, \mathbf{e})}{\sum_{\mathbf{e}'} \exp \sum_m \lambda_m h_m(\mathbf{f}, \mathbf{e}')} \, d\boldsymbol{\lambda}, \tag{8}$$

and, finally, the decision rule will be given by the maximisation of the previous equation, i.e.,

$$\hat{\mathbf{e}} = \underset{\mathbf{e}}{\operatorname{argmax}} \, p(\mathbf{e} \mid \mathbf{f}; \mathcal{T}, \mathcal{A}). \tag{9}$$

Note that, in this case, the denominators in Equation 8 cannot be easily neglected, as was the case in Equation 2, since they are affected by the integral.

In the next section, different strategies for approximating the integral in Equation 8 are presented. However, before carrying on with the presentation of these strategies, there are several approximations that need to be performed so that the predictive distribution can be computed. Firstly, $p(\mathcal{A} \mid \boldsymbol{\lambda}; \mathcal{T})$ and $p(\mathbf{e} \mid \mathbf{f}, \boldsymbol{\lambda})$ contain in their denominator sums over all possible sentences of the target language, which is not computable. For this reason, $\sum_{\mathbf{e}'}$ is approximated as the sum over all the hypotheses within the n-best list generated during the regular decoding process. Coherently, instead of performing a full search of the best possible translation we will only consider eligible the ones present in such n-best list, leading to

$$p(\mathbf{e} \mid \mathbf{f}; \mathcal{T}, \mathcal{A}) \propto \mathcal{Z} \int \prod_{a=1}^{|\mathcal{A}|} \frac{\exp \sum_m \lambda_m h_m(\mathbf{f}_a, \mathbf{e}_a)}{\sum_{\mathbf{e}' \in \Gamma_n(\mathbf{f}_a)} \exp \sum_m \lambda_m h_m(\mathbf{f}_a, \mathbf{e}')} \exp\left\{-\frac{||\boldsymbol{\lambda} - \boldsymbol{\lambda}_{\mathcal{T}}||^2}{2\sigma_{\mathcal{T}}}\right\} \frac{\exp \sum_m \lambda_m h_m(\mathbf{f}, \mathbf{e})}{\sum_{\mathbf{e}' \in \Gamma_n(\mathbf{f})} \exp \sum_m \lambda_m h_m(\mathbf{f}, \mathbf{e}')} \, d\boldsymbol{\lambda}, \tag{10}$$

where $\Gamma_n(\mathbf{f})$ represents the set of n best translation hypotheses that can be generated for sentence $\mathbf{f}$.

In addition, typical state-of-the-art SMT systems do not guarantee complete coverage of all possible sentence pairs due to the great number of heuristic decisions involved, and out-of-vocabulary words may imply that the SMT model is unable to explain a certain bilingual sentence completely. Hence, instead of using the true reference present in $\mathcal{A}$, we will use the best possible translation $\mathbf{e}^*$

generated during the decoding process, i.e., the best translation within $\Gamma_n(\mathbf{f}_a)$. $\mathbf{e}^*$ is often referred to as oracle derivation in related work [22].

The formulation presented would also allow considering as model parameters the feature functions $\mathbf{h}(\cdot,\cdot)$. However, in the present article we will only consider the adaptation of $\boldsymbol{\lambda}$, since adapting $\mathbf{h}$ is much more costly and is left as future work.

4. Sampling methods

Computing the integral over the complete parametric space, as described in Equation 8, is often computationally unfeasible. Moreover, the function to be integrated might not even be integrable. Hence, it is often approximated by a discrete sum over a sampling of such parameters. For simplicity $\mathcal{S}(\boldsymbol{\lambda}_\mathcal{T})$ will denote a specific sampling of $\boldsymbol{\lambda}$, starting from $\boldsymbol{\lambda}_\mathcal{T}$.

4.1. Heuristic sampling

As a first approach, the close neighbourhood of $\boldsymbol{\lambda}_\mathcal{T}$ was explored. For doing this, each one of the components of $\boldsymbol{\lambda}$ was perturbed by a random amount, successively:

Input $\boldsymbol{\lambda}_\mathcal{T}$, the parameter mean vector of size Q
Output $\mathcal{S}(\boldsymbol{\lambda}_\mathcal{T})$, a (pseudo-)random sampling of $\boldsymbol{\lambda}_\mathcal{T}$
Initialise $\mathcal{S}(\boldsymbol{\lambda}_\mathcal{T}) = \{\boldsymbol{\lambda}_\mathcal{T}\}$
For s in $\{1, \ldots, N_S\}$ do
    $\boldsymbol{\lambda}_s = \boldsymbol{\lambda}_\mathcal{T}$
    $k = s \bmod Q$
    $\lambda_{s,k} = \lambda_{s,k} + \text{rand}(-0.5, 0.5)$
    $\boldsymbol{\lambda}_s = \boldsymbol{\lambda}_s / \sum_k |\lambda_{s,k}|$
    $\mathcal{S}(\boldsymbol{\lambda}_\mathcal{T}) = \mathcal{S}(\boldsymbol{\lambda}_\mathcal{T}) \cup \{\boldsymbol{\lambda}_s\}$

---

where N_S is the amount of sampled $\boldsymbol{\lambda}$ desired, $\boldsymbol{\lambda}_s = [\lambda_{s,1}, \ldots, \lambda_{s,M}]^T$ is a single one of those samples, $|c|$ denotes the absolute value of c, and $\text{rand}(a, b)$ is a random value obtained from a uniform distribution in the interval $[a, b]$. Bear in mind that, although each λ_k may not be on the same scale, typical state-of-the-art systems such as Moses [25] use normalised weights (i.e., $\sum_k |\boldsymbol{\lambda}_k| = 1$). Such normalisation was found to be an important step during the experimental phase, since it implies that the range $[-0.5, 0.5]$ is large enough so as to provide variability to the samples obtained.

Although this algorithm involves a series of heuristic decisions, it has one main advantage: the sample $\mathcal{S}(\boldsymbol{\lambda}_\mathcal{T})$ produced is not a function of $\mathcal{A}$. This means that most terms in Equation 7 can be precomputed, except for $p(\mathbf{e} \mid \mathbf{f}, \boldsymbol{\lambda})$, and $\mathcal{S}(\boldsymbol{\lambda}_\mathcal{T})$ does not need to be recomputed for every $\mathcal{A}$, which would be far too costly when applying BPA in an online scenario. In next section, we present a sampling strategy more theoretically sound, but which does need to recompute $\mathcal{S}(\boldsymbol{\lambda}_\mathcal{T})$ for each $\mathcal{A}$.

However, the approximation provided by this heuristic algorithm is sensible to normalisation. This can be seen e.g. in Equation 8: dropping the normalisation constant $\mathcal{Z}$ leads to a product of probabilities, which implies that larger amounts of adaptation data will lead to smaller numeric values, and the relative importance of $p(\mathcal{A} \mid \boldsymbol{\lambda}; \mathcal{T})$ will fade when more evidence arrives. To compensate this fact, after replacing the integral by a finite summation, Equation 7 is complemented with a leveraging factor δ, such that

$$p(\mathbf{e} \mid \mathbf{f}; \mathcal{T}, \mathcal{A}) \approx \sum_{\boldsymbol{\lambda} \in \mathcal{S}(\boldsymbol{\lambda}_{\mathcal{T}})} (p(\mathcal{A} \mid \boldsymbol{\lambda}; \mathcal{T}) p(\mathbf{e} \mid \mathbf{f}, \boldsymbol{\lambda}))^{\frac{1}{\delta}} p(\boldsymbol{\lambda} \mid \mathcal{T}). \tag{11}$$

4.2. Markov chain Monte Carlo

Markov chain Monte Carlo (MCMC) methods [24] obtain samples $\mathcal{S}(\boldsymbol{\lambda}_{\mathcal{T}})$ of a variable (in this case $\boldsymbol{\lambda}$) assumed to follow a certain distribution, i.e., $p(\boldsymbol{\lambda} \mid \mathcal{T}, \mathcal{A})$. MCMC methods are specially suited for sampling distributions where the normalisation constant cannot be evaluated [24]. For doing this, a (first order) Markov chain is established, where each new sample $\boldsymbol{\lambda}^*$ depends on the previous sample $\boldsymbol{\lambda}'$. Specifically, in this article we will be using the Metropolis-Hastings (MH) algorithm [26], which consists in drawing a sample $\boldsymbol{\lambda}^*$ from a given proposal distribution $q(\boldsymbol{\lambda} \mid \boldsymbol{\lambda}')$. Then, $\boldsymbol{\lambda}^*$ is accepted with probability

$$A(\boldsymbol{\lambda}^*, \boldsymbol{\lambda}') = \min\left(1, \frac{\tilde{p}(\boldsymbol{\lambda}^*) q(\boldsymbol{\lambda}' \mid \boldsymbol{\lambda}^*)}{\tilde{p}(\boldsymbol{\lambda}') q(\boldsymbol{\lambda}^* \mid \boldsymbol{\lambda}')}\right), \tag{12}$$

with $p(\boldsymbol{\lambda}) = \tilde{p}(\boldsymbol{\lambda}) / \mathcal{Z}_p$ being the distribution from which we intend to sample, and $\mathcal{Z}_p$ being the normalisation term for $p(\boldsymbol{\lambda})$. In Equation 12, $\tilde{p}(\boldsymbol{\lambda})$ can be safely used instead of $p(\boldsymbol{\lambda})$, since $\mathcal{Z}_p$ would be simplified. If the proposal distribution is symmetric, terms $q(\cdot \mid \cdot)$ can also be simplified.

The proposal distribution is often set as a normal distribution $\mathcal{N}(\boldsymbol{\lambda}; \boldsymbol{\lambda}', I \cdot \sigma_o)$, with mean vector $\boldsymbol{\lambda}'$ and covariance matrix a diagonal matrix with main diagonal σ_o. Establishing σ_o is critical, since too small values will lead to a high rejection rate and the sampling chain will most likely get stuck at a local maximum, while too big values will lead to a chaotic chain which will not sample the density function appropriately.

Another aspect that needs to be taken into account when building a MCMC chain is the burn-in phase [24], which is the number of samples that need to be drawn in order to assume independence from the initial state of the Markov chain.

Once $\mathcal{S}(\boldsymbol{\lambda}_{\mathcal{T}})$ has been obtained from $p(\boldsymbol{\lambda} \mid \mathcal{T}, \mathcal{A})$ (or, dropping the normalisation constant in Equation 5, from $p(\mathcal{A} \mid \boldsymbol{\lambda}; \mathcal{T}) p(\boldsymbol{\lambda} \mid \mathcal{T})$), Equation 7 is approximated, again according to the Strong Law of Large Numbers [24], as

$$p(\mathbf{e} \mid \mathbf{f}; \mathcal{T}, \mathcal{A}) \approx \mathcal{Z}' \sum_{\boldsymbol{\lambda} \in \mathcal{S}(\boldsymbol{\lambda}_{\mathcal{T}})} p(\mathbf{e} \mid \mathbf{f}, \boldsymbol{\lambda}), \tag{13}$$

where δ is not required either. Although the right hand of Equation 13 seems independent from $\mathcal{T}$ and $\mathcal{A}$, this is only a notation issue, since such dependency

is hidden within $\mathcal{S}(\boldsymbol{\lambda}_\mathcal{T})$, and $\mathcal{S}(\boldsymbol{\lambda}_\mathcal{T})$ must be recomputed for every adaptation set $\mathcal{A}$.

5. Experiments

Experiments were performed by means of the open-source MT toolkit Moses [25][2] (version 0.91) in its default non-monotonic configuration, which includes 5 translation models (direct- and inverse- translation and lexicalised models and the phrase-penalty), 7 re-ordering models (an exponential model and the six models included within the `msd-reordering-fe` model [27]), the word-penalty and a word-based language model, i.e., $|\boldsymbol{\lambda}| = 14$. The language model used was a 5-gram with modified Kneser-Ney smoothing [28], built with the SRILM toolkit [29].

Translation quality will be assessed mainly by means of single-reference TER [30]. TER (*Translation Error Rate*) is an error metric (i.e., the lower the better) that computes the minimum number of edits required to modify the system hypotheses so that they match the reference. Possible edits include insertion, deletion and substitution of single words, as well as shifts of word sequences. Some results will also be presented in terms of BLEU [31], with the purpose of assessing whether the improvements in TER also correspond to improvements in BLEU. BLEU (*BiLingual Evaluation Understudy*) is a precision metric (i.e., the higher the better) that measures n-gram ($n \leq 4$) coverage of the system hypotheses with respect to the reference, with a penalty for sentences that are too short. For computing the best possible hypothesis $\mathbf{e}^*$ as described in Section 3, TER will be used, since BLEU is not always well defined at the sentence level, given that it implements a geometrical average which is zero whenever there is no common 4-gram between hypothesis and reference, e.g., a 3-word sentence. Selecting $\mathbf{e}^*$ with smoothed versions of BLEU is planned as future work.

The points in the plots presented in this section display the average of ten experiments, in which $\mathcal{A}$ was re-drawn each time with replacement. Unless stated otherwise, the x-axis will always be in logarithmic scale and display $|\mathcal{S}(\boldsymbol{\lambda}_\mathcal{T})|$. The scale of the y-axis will be linear whenever the plot displays translation quality, and logarithmic in the case of the confidence interval sizes (in TER points unless stated otherwise). These confidence intervals present the 95% confidence level and were computed as 2σ, where σ is the empirical standard deviation observed in the 10 repetitions. Note that the full confidence interval would be 4σ, i.e., $\pm 2\sigma$. Confidence intervals are displayed in different plots, instead of using error bars, because otherwise the translation quality plots would present vertical lines across the complete plot, rendering it unreadable. For analysing the effect of the different meta-parameters in BPA, we performed experiments on all the available domains (explained in next section). We only present here the clearest plot, for readability purposes. For simplicity purposes, the size n

[2] Available from http://www.statmt.org/moses/

of the n-best lists in Equation 10 was the same for both $\Gamma_n(\mathbf{f})$ and $\Gamma_n(\mathbf{f}_a)$ in all experiments, and was set to 500 in all experiments, unless stated otherwise.

In the following subsections, we will first present the corpora used (Section 5.1). Then, we will analyse the effect of the different meta-parameters involved in the BPA samplings (Section 5.3 for the heuristic sampling, Section 5.4 for the MCMC sampling). This being done, we compare both sampling approaches in terms of final translation quality (Section 5.5). Next, in Section 5.6, we compare the best performing BPA approach, namely MCMC, with other re-estimation approaches present in the literature, and a final analysis of the results is performed in Section 5.7.

5.1. Corpora

The experiments conducted in this paper were carried out on five different corpora, belonging to different domains, all of them stemming from the domain adaptation Summer workshop carried out at the John Hopkins University in 2012 [20]. In this workshop, the task was to adapt French→English translation models. The out-of-domain corpus provided originated in the parliamentary domain (Canadian Hansards), and the in-domain corpora included the medical domain (henceforth referred to as EMEA), the general news domain (henceforth referred to as NEWS), the press domain (NRC), and the subtitle domain (SUBS). Statistics of the out-of-domain corpus are provided in Table 1, and statistics of the in-domain corpora are provided in Table 2. Even though it might seem odd that the SUBS in-domain training data is larger than the Hansards out-of-domain training data, this is how the task was designed. Note that this task was intended for an adaptation problem involving many more parameters, i.e., feature function adaptation, which requires much more data than the problem of scaling factor adaptation. We will only make use of much less data, since it is common knowledge that scaling factors are already well estimated with about 2000 sentences. However, the amount of data available allows us to perform several random extractions of the adaptation set $\mathcal{A}$.

		Training ($\mathcal{T}$)		Development ($\mathcal{D}$)	
		French	English	French	English
Hansards	Sentences	8.1M		2000	
	Run. words	163M	144M	40.1k	35.8k
	Vocab/OoV	191.4k	186.8k	15	26
	Avg. length	19.9	17.8	20.1	17.9

Table 1: Main figures of the out-of-domain corpus. *OoV* stands for Out-of-Vocabulary words (types) with respect to $\mathcal{T}$.

The standard features $\mathbf{h}$ were estimated on the training partition of the Hansards corpus $\mathcal{T}$, whereas a canonical $\boldsymbol{\lambda}_{\mathcal{T}}$ was estimated on the development subset $\mathcal{D}$ (i.e., a held-out subset of $\mathcal{T}$ used to estimate $\mathbf{h}$) by means of the default MERT implementation within Moses. Translation quality will be estimated on the different in-domain test subsets.

		Training ($\mathcal{A}$)		Test	
		French	English	French	English
EMEA	Sentences	472k		2045	
	Run. words	6.5M	5.9M	29k	25k
	Vocab/OoV	35k	30k	1115	1073
	Avg. length	13.9	12.5	14.2	12.0
NEWS	Sentences	136k		2489	
	Run. words	3.9M	3.3M	69k	62k
	Vocab/OoV	63k	53k	1098	1040
	Avg. length	28.9	24.6	27.9	24.8
NRC	Sentences	66k		1982	
	Run. words	2.2M	1.7M	65k	52k
	Vocab/OoV	79k	78k	2587	2869
	Avg. length	32.7	26.3	32.8	26.4
SUBS	Sentences	19M		3306	
	Run. words	155M	174M	32k	36k
	Vocab/OoV	362k	293k	599	385
	Avg. length	8.1	9.1	9.7	10.9

Table 2: Main figures of the in-domain corpora. *OoV* stands for Out-of-Vocabulary words (types) with respect to $\mathcal{T}$.

5.2. Experimental setup

Whenever reporting translation quality of a BPA experiment, the (full) experimental setup involves:

1. Obtain n-best lists from the corresponding test set, with $\mathbf{h}_{\mathcal{T}}$ estimated on the training partition of the Hansards corpus and the $\boldsymbol{\lambda}_{\mathcal{D}}$ estimated on the development subset of the Hansards corpus.
2. Obtain n-best lists from the adaptation data, similarly as done for the test.
3. Compute sentence-level TER scores for each one of the n-best hypotheses for both test and adaptation.
4. Rerank the n-best lists obtained according to Equation 11 or Equation 13, accordingly.

Whenever reporting translation quality of a re-estimation experiment, the (full) experimental setup involves:

1. Initialize the $\boldsymbol{\lambda}$ estimation procedure with the $\boldsymbol{\lambda}_{\mathcal{D}}$ estimated on the development subset of the Hansards corpus, $\mathbf{h}_{\mathcal{T}}$ estimated on $\mathcal{T}$.
2. Run the corresponding $\boldsymbol{\lambda}$ estimation algorithm on the adaptation data, obtaining $\boldsymbol{\lambda}_{\mathcal{A}}$.
3. Decode the final test set using $\boldsymbol{\lambda}_{\mathcal{A}}$.

In both cases, these will also be the steps considered when reporting adaptation times (Section 5.7), without including the time taken by the initial $\mathbf{h}_{\mathcal{T}}$ and $\boldsymbol{\lambda}_{\mathcal{D}}$ estimation procedures, since these are common steps.

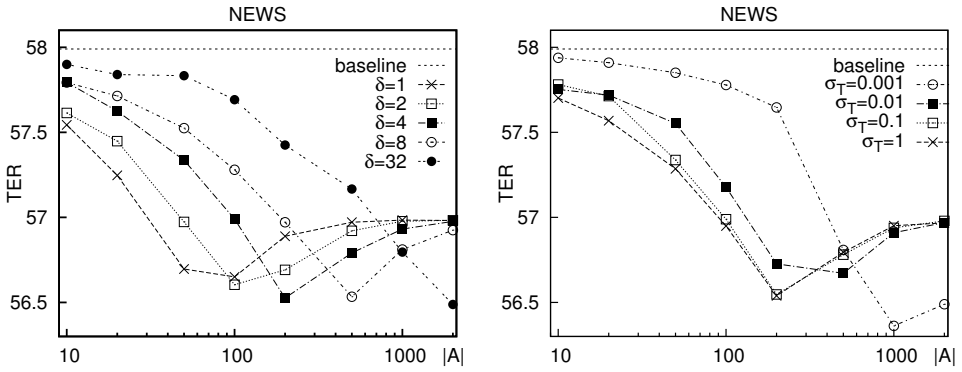

Figure 1: Effect of the δ leveraging factor (left) and of the prior variance $\sigma_{\mathcal{T}}$ (right) on translation quality (TER) in BPA with heuristic sampling. In the left plot, $\sigma_{\mathcal{T}} = 0.1$ was used. In the right plot, $\delta = 4$.

5.3. Heuristic sampling experiments

Results for this kind of sampling are shown in Figure 1, for different values of the δ leveraging factor and different values for the prior distribution variance $\sigma_{\mathcal{T}}$ in Equation 8. As shown, the BPA approach is able to improve over the unadapted system from the very beginning. The results show that smaller values of δ lead to a slight degradation in translation quality when the size of $\mathcal{A}$ becomes larger. The reason for this can be explained by looking at Equation 11. Since $p(\mathcal{A} \mid \boldsymbol{\lambda}; \mathcal{T})$ is implemented as a product of probabilities, the more adaptation samples the smaller becomes $p(\mathcal{A} \mid \boldsymbol{\lambda}; \mathcal{T})$, and a higher value of δ is needed to compensate this fact. The differences between different δ values were, although small, found to be coherent in all the other experiments conducted.

As for the effect of the prior distribution variance $\sigma_{\mathcal{T}}$, it was observed to have a very similar effect than δ. On the one hand, smaller values of $\sigma_{\mathcal{T}}$ entail that the adaptation procedure is practically de-activated for a small amount of adaptation samples. Since $\mathcal{N}(\boldsymbol{\lambda}; \boldsymbol{\lambda}_{\mathcal{T}}, I \cdot \sigma_{\mathcal{T}})$ is the more "peaky" the smaller $\sigma_{\mathcal{T}}$ is, less variability for the different $\boldsymbol{\lambda} \in \mathcal{S}(\boldsymbol{\lambda}_{\mathcal{T}})$ is allowed for small values of $\sigma_{\mathcal{T}}$. However, this implies that a certain σ will be "assigned" by the prior a numerically much smaller probability even if it is not very far away from $\sigma_{\mathcal{T}}$. This explains the fact that smaller values of $\sigma_{\mathcal{T}}$ behave better when the amount of adaptation data increases, since, as seen for the leveraging factor δ, $p(\mathcal{A} \mid \boldsymbol{\lambda}; \mathcal{T})$ also has a numerically smaller value in those cases.

Although not shown here for space reasons, increasing the number of sampled parameters $|\mathcal{S}(\boldsymbol{\lambda}_{\mathcal{T}})|$, did not have any effect on the average translation quality, as expected. However, it did provide further robustness to the results, and confidence intervals tended to be smaller for larger values of $|\mathcal{S}(\boldsymbol{\lambda}_{\mathcal{T}})|$, specially when the size of $\mathcal{A}$ was small. When increasing $|\mathcal{S}(\boldsymbol{\lambda}_{\mathcal{T}})|$ from 1000 to 2000, the improvements in stability were already very scarce, and were probably not worth the computational overhead.

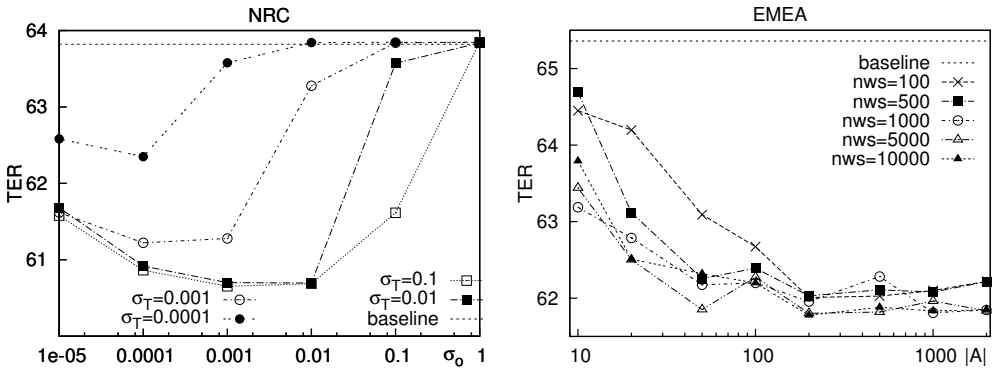

Figure 2: Effect of the different prior variance $\sigma_{\mathcal{T}}$ and proposal variance σ_o (left) and number of weight samples (`nws` in the right plot) on translation quality (TER) in BPA with MCMC sampling.

5.4. MCMC sampling

First, appropriate values for the variance of the prior distribution $\sigma_{\mathcal{T}}$ and of the proposal distribution σ_o were analysed (Figure 2 left). Note that σ_o is tightly related to $\sigma_{\mathcal{T}}$, since they both control how much variation is introduced into the predictive distribution of BPA. The best values for σ_o seemed to appear when setting $\sigma_o \approx 0.1 \cdot \sigma_{\mathcal{T}}$, and considering $\sigma_o > \sigma_{\mathcal{T}}$ seemed to lead to non-adaptive systems. In the rest this Section, $\sigma_{\mathcal{T}} = 0.1$ and $\sigma_o = 0.01$ were adopted.

Different $\mathcal{S}(\boldsymbol{\lambda}_{\mathcal{T}})$ sizes, i.e., different MCMC chain lengths, had an important effect on the confidence intervals, and also to some extent on average translation quality (Figure 2 right). As shown, the improvements achieved when moving from $|\mathcal{S}(\boldsymbol{\lambda}_{\mathcal{T}})| = 1000$ to $|\mathcal{S}(\boldsymbol{\lambda}_{\mathcal{T}})| = 5000$ might not be worth the computational overhead (let alone values larger than 5000). Although not reported here in order to avoid including too many plots, the duration of the burn-in phase was not found to be specially critical. A slight gain in stability was observed for small $\mathcal{S}(\boldsymbol{\lambda}_{\mathcal{T}})$ sizes, although no real conclusion could be drawn. Nevertheless, burn-in length was set to 500 in the rest of the experiments involving MCMC.

5.5. Sampling comparison

Once the effect of the different meta-parameters of the two different sampling approaches have been analysed, we now pursue to compare these two approaches among each other. In addition, with the purpose of analysing the effect of considering the integral (or finite sum in practise) over the complete parametric space (or parameter sampling in practise), instead of just a point estimate of the parameters, we performed a series of experiments by replacing the integral by an `argmax` operation. This means that, instead of computing the probability of a given output sentence as the complete integral, we only consider that specific $\hat{\boldsymbol{\lambda}}$ that yields the highest probability according to $p(\mathcal{A} \mid \boldsymbol{\lambda}; \mathcal{T}) p(\boldsymbol{\lambda} \mid \mathcal{T}) p(\mathbf{e} \mid \mathbf{f}, \boldsymbol{\lambda})$, following a Viterbi-like approach. The results are shown in Figure 3. As

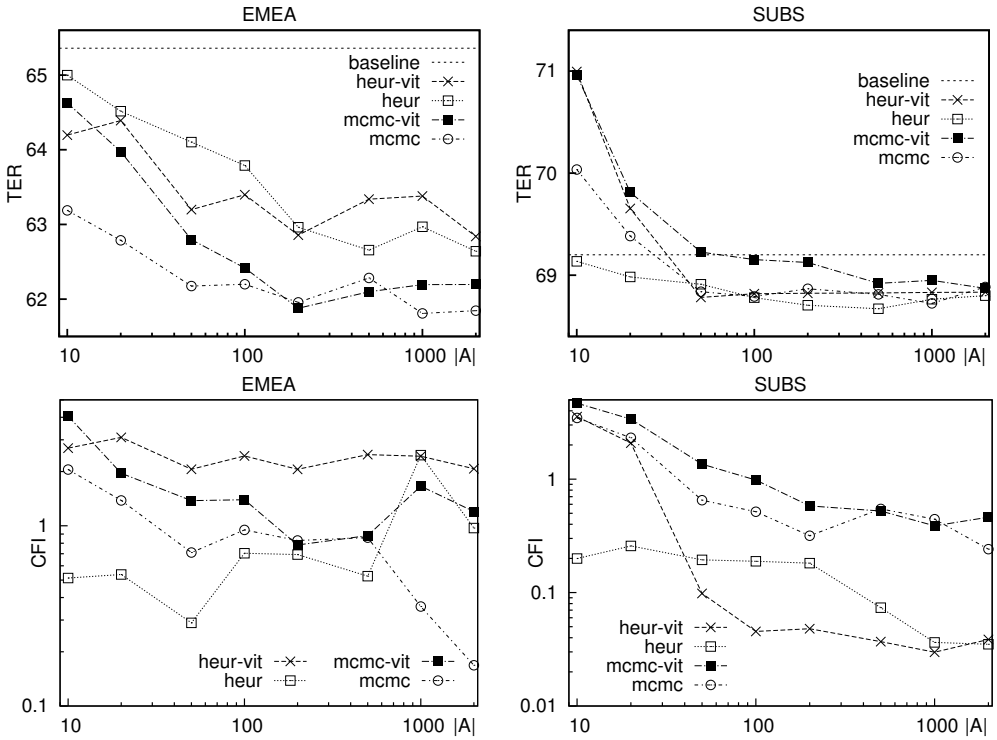

Figure 3: Batch adaptation with the Viterbi approach and different amount of adaptation set sizes. The top two plots display translation quality, while the lower two display confidence interval size.

shown, the MCMC sampling seems to perform better for the EMEA corpus, but worse for the SUBS corpus. Because the way in which the samples are extracted, `heur` relies much more on the parameter prior than `MCMC`, and it must be concluded that the test set chosen presents a different composition than the adaptation data. MCMC also presented a better behaviour for the NRC and NEWS corpora (not shown here for space reasons). On the other hand, the `heur-vit` setup proved to be extremely stable for $|\mathcal{A}| > 100$ for the SUBS, NRC and NEWS corpora. In addition, when looking at the translation quality of the SUBS corpus, it stands out that both `vit` curves behave specially poorly for small amounts of adaptation data. This elucidates precisely the effect of the integral: as already discussed above, the test set of the SUBS corpus seems to be quite different from the adaptation data, which means that, when $|\mathcal{A}|$ is small, over-training towards the adaptation data is more prone to occur. However, considering the integral over the complete parametric space tends to compensate this, providing robustness in case that adaptation and test data do not match. Hence, it can be said that, although `heur-vit` is more stable in some situations, its behaviour is less predictable. Lastly the `mcmc-vit` setting

is not really noteworthy, since it simply performs worse than the `mcmc` setting, as could be expected.

5.6. Comparison between BPA and parameter re-estimation

To synthesise the different strategies, the different SMT systems compared in this section when adapting $\boldsymbol{\lambda}$ are:

- Baseline system: Phrase-pairs extracted from the Hansards training corpus, i.e., $\mathbf{h}$ estimated on $\mathcal{T}$. Scaling factors $\boldsymbol{\lambda}_{\mathcal{T}}$ estimated on $\mathcal{D}$ by means of MERT.
- MCMC: Initial setup identical to the baseline system. Adaptation samples $\mathcal{A}$ were randomly extracted from the training partitions of the in-domain corpora (i.e., EMEA, NEWS, NRC or SUBS). $\boldsymbol{\lambda}_{\mathcal{T}}$ within $p(\boldsymbol{\lambda} \mid \mathcal{T})$ estimated on $\mathcal{D}$ with MERT. The sampling strategy used within BPA was MCMC.
- MERT: Initial setup identical to the baseline system. The adaptation samples $\mathcal{A}$ described above were used for estimating a new set of scaling factors using MERT.
- MIRA: As MERT, but scaling factors are estimated by means of MIRA. Note that MIRA is an incremental estimation strategy, and $\boldsymbol{\lambda}_{\mathcal{T}}$ were provided for the MIRA starting point. Hence, MIRA could also be considered a sort of adaptation strategy (i.e., not full re-estimation).
- PRO: As MERT, but scaling factors are estimated by means of pairwise optimisation.
- MERT+: Initial setup identical to the baseline system. Then, $\boldsymbol{\lambda}$ was re-estimated on $\mathcal{A}$ and $\mathcal{D}$, concatenated.

In addition, we also conducted experiments by concatenating $\mathcal{A}$ and $\mathcal{D}$, and using the result for estimating $\boldsymbol{\lambda}$ by means of MERT. However, such strategy performed consistently worse in terms of TER than the other re-estimation strategies analysed here. For this reason, this setup was removed from the final comparison in order to avoid clogging the plots with too many curves.

It must be emphasised, however, that the re-estimation strategies are not really a fair comparison, since they are all by far much more costly than BPA (computational cost is analysed in Section 5.7). In addition, they involve several translation steps, each of which re-computes the n-best list, and have better chances to obtain better hypotheses, whereas the BPA strategies implemented rely on a pre-computed n-best list of fixed size (in this Section, n-best size was set to 500).

Results of such comparison can be seen in Figure 4. There are several things that should be noted:

- For small amounts of adaptation data, BPA is the strategy that performs the best in all cases except for the NRC corpus.

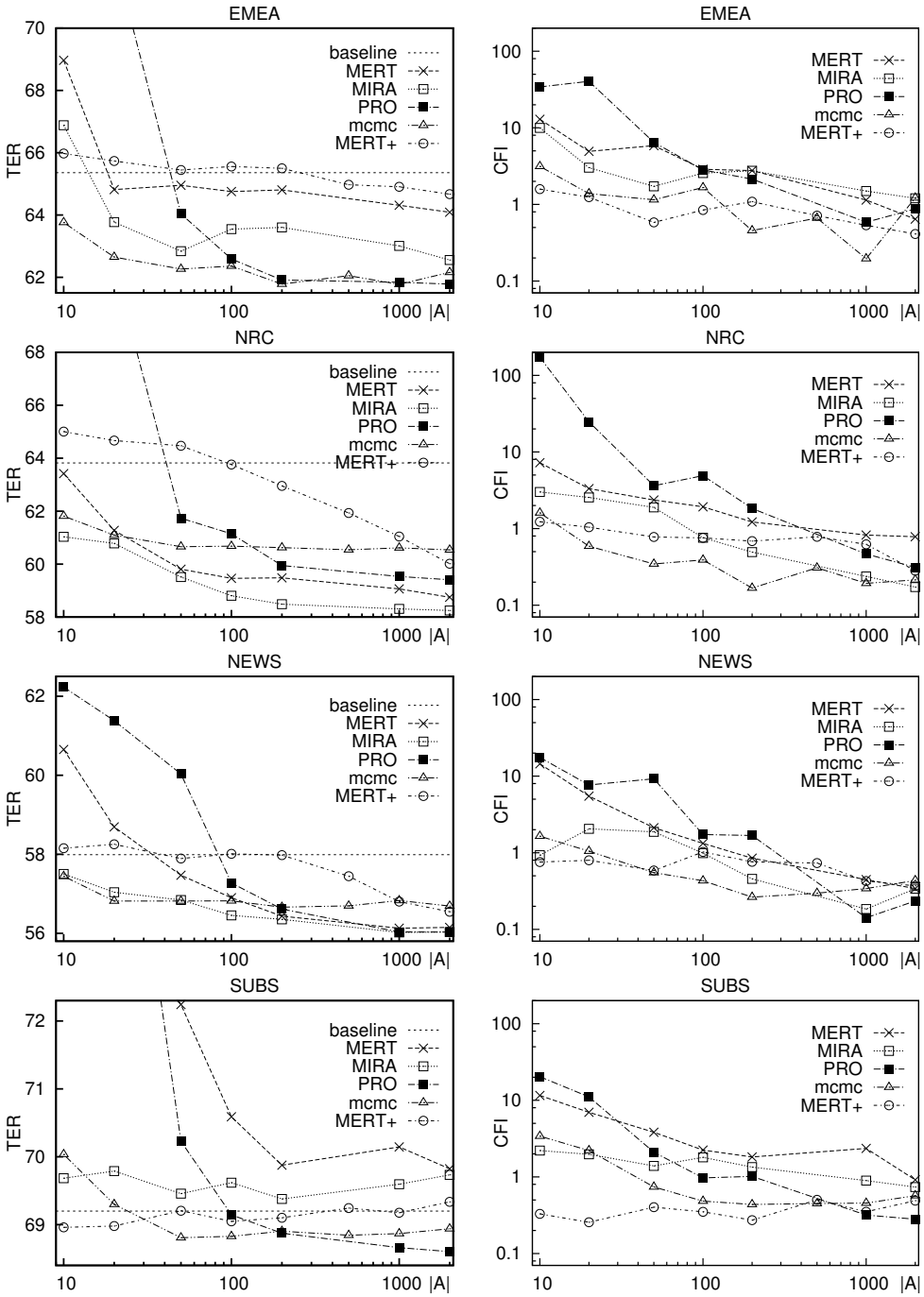

Figure 4: Performance comparison across the different corpora analysed and with the different $\boldsymbol{\lambda}$ estimation strategies. The four plots on the left display TER, while the four plots on the right display the size of the confidence intervals.

- MERT and PRO display a very unstable behaviour for small $|\mathcal{S}(\boldsymbol{\lambda}_{\mathcal{T}})|$ sizes, and MIRA seems to exhibit better performance. This is not surprising, since MIRA can also be seen as an adaptation strategy, as discussed above.

- The SUBS corpus appears to be a specially difficult corpus. In the first place, TER scores are specially high. In addition, neither the MERT and MIRA strategies are able to improve over the unadapted baseline.

- For small amounts of adaptation data, the confidence intervals of PRO, MIRA and MERT may be as big as 10-20 TER points. When taking into account a baseline of about 65 TER points, 10 points may well imply the difference between the output being useful or being completely useless. In contrast, BPA seldom yields confidence intervals of more than 2 TER points.

- MERT+ yields estimations which are as stable as BPA, but yields worse performance in most settings (i.e., except for the SUBS corpus). We consider this point important, since the stability achieved by BPA reveals precisely that BPA is an appropriate adaptation strategy for this problem: the size of the confidence intervals can be seen as a measure of how prone is a given algorithm to over-training, and BPA proves to be able to provide quite stable estimations even for very small amounts of data.

From these observations, it can be concluded that BPA is an effective adaptation strategy. An adaptation strategy only is useful when the amount of adaptation data is small, and BPA proves to perform well under such circumstances. If the amount of adaptation data is larger (> 100), BPA still yields an acceptable behaviour, although the pure re-estimation strategies do yield better estimates of $\boldsymbol{\lambda}$.

Since BLEU is a more standard evaluation metric than TER among the SMT community, we also report some BLEU results in Figure 5. The overall analysis varies little with respect to the one done with TER. However, two things are noteworthy: firstly, BPA tends to perform slightly worse in terms of BLEU. This is actually expected, since the best possible hypothesis hypothesis $\mathbf{e}^*$ is selected according to TER, and BLEU includes a brevity penalty witch TER does not take into account. Second, the size of the confidence intervals obtained is nearly the same as in the case of TER, and BPA tends to achieve smaller confidence intervals, specially in the case of small amounts of adaptation data. It is also interesting to see that the re-estimation approaches need more than 100 adaptation samples in order to achieve the performance that the heuristic version of BPA achieves with only 10 samples. As in the case of the experiments involving TER, the results for NEWS and NRC were similar to those obtained with the EMEA corpus, and SUBS seems to be specially difficult.

5.7. Analysis

Given that adapting $\boldsymbol{\lambda}$ is a rather coarse-grained adaptation strategy, it is important to analyse where the improvements come from, and whether such

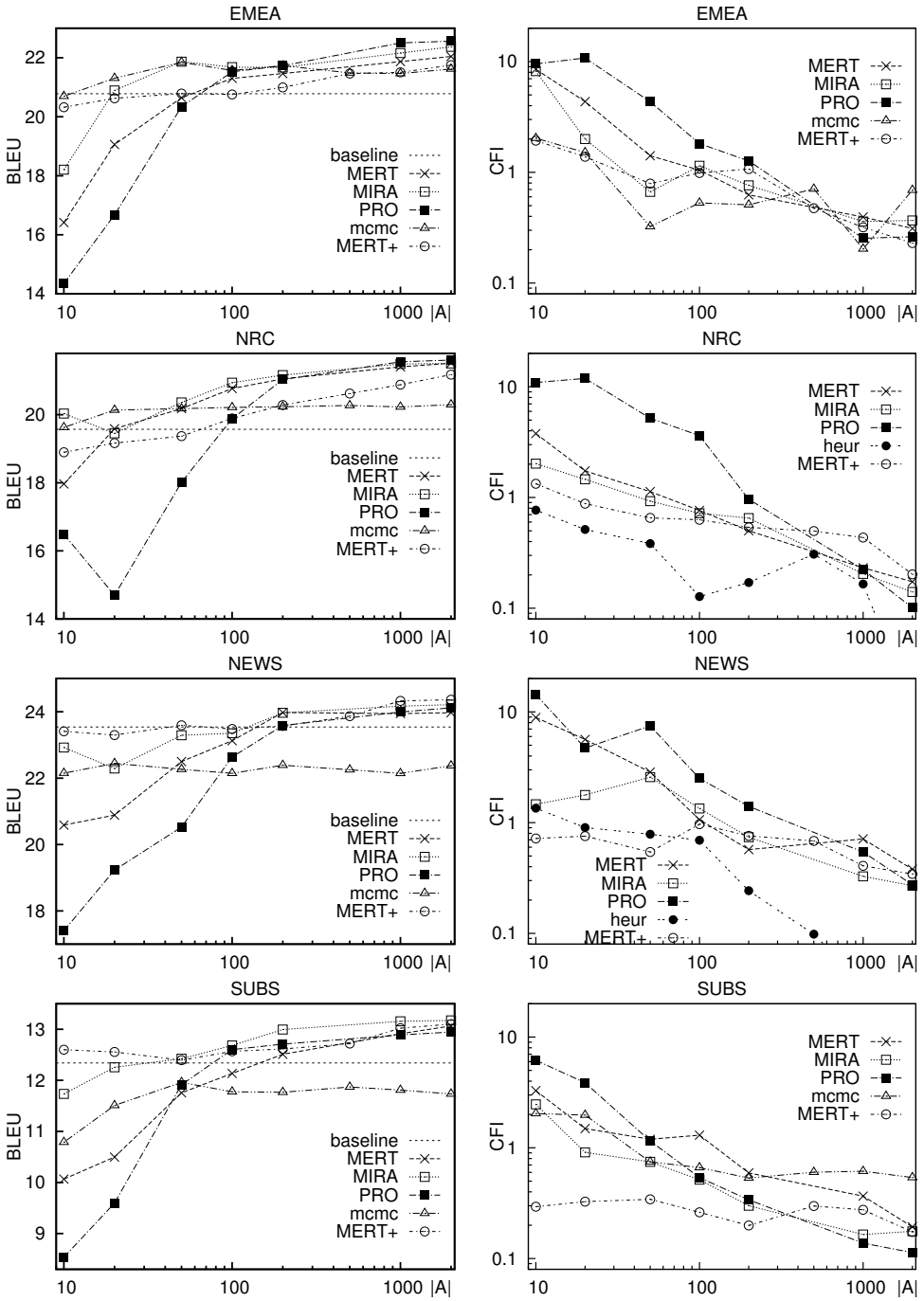

Figure 5: Performance comparison across the different corpora analysed and with the different $\boldsymbol{\lambda}$ estimation strategies. The four plots on the left display BLEU, while four plots on the right display the size of the confidence intervals.

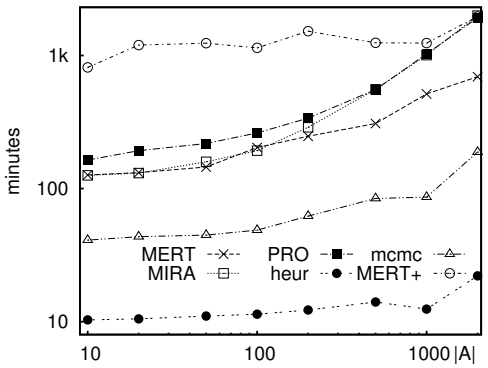

Figure 6: Time consumed by the different approaches compared. In the case of BPA, $|\mathcal{S}(\boldsymbol{\lambda}_{\mathcal{T}})| = 1000$ and $|n-\text{best}| = 500$. NRC corpus considered.

improvements may be due only to a re-adjustment in sentence length. For this reason, we analysed the n-gram precision and the brevity penalty implemented within BLEU. For a certain n, n-gram precision is computed as the proportion of n-grams that match between the candidate hypotheses and the references. The brevity penalty is defined as $\min(1, r)$, being r the ratio between hypothesis and reference lengths. In Table 3, average n-gram precision for all 10 repetitions is shown for two of the corpora analysed. Since we are interested in analysing the behaviour in conditions where there is a very small amount of data available, and when there is more data available, we studied n-gram precision with 10 and with 100 adaptation samples. Interestingly, it is observed that, as a matter of fact, sentence length is actually penalising rather heavily the BPA approach: for the EMEA corpus, BPA is the only strategy to be penalised by the brevity penalty, meaning that it is the one that performs worst when only taking into account sentence length. This is actually expected, since BPA selects the best hypothesis according to TER. However, in terms of n-gram precision, it clearly outperforms the other approaches for 10 adaptation samples, and also for 100 adaptation samples in the lower order n-grams. In the higher order n-grams, BPA seems to perform a bit worse. When considering the SUBS corpus, two facts stand out: on the first place, that MCMC and MIRA yield the best n-gram precision for $|\mathcal{A}| = 10$, but also suffer a very heavy penalisation in terms of sentence length. When considering $|\mathcal{A}| = 100$, all four methods are very heavily penalised by the brevity penalty. This points towards a big mismatch between the SUBS adaptation data and the SUBS test data, which explains as well the fact that all methods studied yield pretty poor results. Such mismatch can already be observed when looking at the average sentence length of the sentences, shown in Table 2, where the SUBS corpus is the one that presents the most mismatch between $\mathcal{A}$ and test. All in all, n-gram precision seems to signal that improvements obtained by BPA are due to a better lexical choice of the phrases involved, and not to a side-effect of adjusting the output sentence length.

EMEA	base-line	10 adaptation samples				100 adaptation samples			
		MCMC	MERT	MIRA	PRO	MCMC	MERT	MIRA	PRO
1-gram	54.6	56.1	50.5	53.5	46.9	56.6	55.3	56.0	55.8
2-gram	25.9	25.8	17.4	25.0	16.8	27.1	26.6	27.2	27.4
3-gram	14.8	14.5	7.3	14.4	7.6	15.6	15.5	16.0	16.2
4-gram	8.9	8.6	3.3	8.8	3.9	9.4	9.6	9.9	9.9
BP	1.00	0.99	1.00	1.00	1.00	0.99	1.00	1.00	1.00
SUBS	base-line	10 adaptation samples				100 adaptation samples			
		MCMC	MERT	MIRA	PRO	MCMC	MERT	MIRA	PRO
1-gram	54.6	50.5	41.3	51.4	38.2	51.0	50.6	49.4	49.1
2-gram	25.9	19.5	12.2	18.5	12.0	20.7	19.7	20.3	20.0
3-gram	14.8	9.6	4.8	8.6	4.7	10.5	9.7	10.4	10.1
4-gram	8.9	4.8	1.9	4.3	1.8	5.5	5.1	5.5	5.3
BP	1.00	0.74	0.93	0.73	0.97	0.75	0.73	0.84	0.86

Table 3: n-gram precision and brevity penalty (BP) for $|\mathcal{A}| = 10$ and $|\mathcal{A}| = 100$, MCMC sampling for BPA and the different re-estimation strategies.

Regarding computational time, Figure 6 reports the time consumed by each one of the approaches reported in Figure 4. Note that, in this case, both axes are plotted in logarithmic scale. Computational time was measured in single-threaded runs of the algorithms presented on 64 bit machines with Intel Xeon CPUs at 2.50GHz with 6MB cache. In the case of BPA, the number of sampled weights was 1000, and the time reported also includes the time required for generating the n-best lists of $\mathcal{A}$ and the sentence-level TER counts. In fact, the time taken only by BPA ranges from 10 to 20 minutes. As shown, the heuristic BPA approach is the fastest one, and both BPA approaches are about one order of magnitude faster than `MERT`, `MIRA` and `PRO`. Although somewhat hidden by the logarithmic scale, it must be noted that `MIRA` and `PRO` present very steep curves for $|\mathcal{A}| = 1000$ and $|\mathcal{A}| = 2000$, and the difference in computational cost between these two and `MERT` is noteworthy.

Lastly we also analysed the effect of varying the size of the n-best list considered (Figure 7). Again, BPA is able to cope well with additional input information, and additional hypotheses in the n-best list imply that BPA is able to select better hypothesis without incurring into over-trained solutions.

6. Conclusions and future work

In this paper, Bayesian predictive adaptation has been thoroughly analysed for its application to log-linear weight adaptation in statistical machine translation. On the one hand, the theoretical framework for adapting the scaling factors present in most state-of-the-art SMT systems has been developed. On the other hand, experimental results analysing the effectiveness of such adaptation procedures have been reported.

Results show that BPA is able to provide consistent improvements in translation quality over the baseline systems, as measured by TER, with as few as

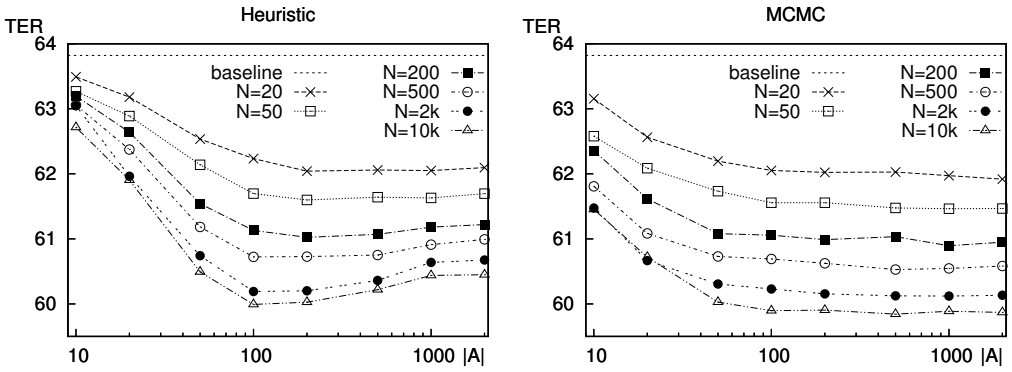

Figure 7: Translation quality with heuristic and MCMC sampling with different sizes of n-best, represented as `N` in the plot. NRC corpus considered.

10 adaptation samples, and up to an amount of adaptation data that allows a complete re-estimation of the model parameters. In addition, BPA proves to be more stable than most re-estimation strategies, which rely heavily on the amount of adaptation data. It should be emphasised that an adaptation technique, by nature, is only useful whenever the amount of adaptation data is low, and BPA proves to behave well in such context. Whenever the amount of adaptation data is high, the best thing that one can do is to re-estimate the model parameters from scratch, although such re-estimation is often very costly. From a computational point of view, the Bayesian adaptation technique presented does not imply a significant computational overhead, and most terms can be precomputed in the case of heuristic sampling. Hence, we consider that it could be easily implemented within the decoder itself without a significant increase in computational complexity. Nevertheless, it must be taken into account that the search space explored by a given n-best list is much more restrained than the one that the decoder will take into account. This means that, if BPA is to be implemented within the decoder (instead of by re-scoring n-best lists), the number of n-best considered by BPA in the term $p(\mathcal{A} \mid \boldsymbol{\lambda}; \mathcal{T})$ must be sufficiently large. We plan to explore this in future work.

Different parameter sampling strategies have been studied when applying BPA to the adaptation of the scaling factors, such as the theoretically sound Markov chain Monte Carlo and an ad-hoc heuristic sampling strategy and the Viterbi approach. It emerges that the heuristic sampling strategy performs slightly worse than MCMC, but is computationally less expensive and most terms can be precomputed. In addition, MCMC yields slightly larger confidence intervals.

As future work, we plan to analyse the possibility of adapting the log-linear features of the translation model, and to extend the current BPA implementation so that it is able to deal with more feature-rich SMT models.

Acknowledgement

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement Nr. 287576 (CasMaCat). Also funded by the Generalitat Valenciana under grant Prometeo/2009/014.

References

[1] R. Kuhn, R. De Mori, A cache-based natural language model for speech recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (6) (1990) 570–583.

[2] Q. Huo, C. Chan, C.-H. Lee, Bayesian adaptive learning of the parameters of hidden markov model for speech recognition, IEEE Transactions on Speech and Audio Processing 3 (5) (1995) 334–345.

[3] P. Koehn, J. Schroeder, Experiments in domain adaptation for statistical machine translation, in: Proc. of the ACL 2nd workshop on SMT, 2007, pp. 136–158.

[4] P. Koehn, Statistical Machine Translation, Cambridge University Press, 2010.

[5] R. Duda, P. Hart, D. Stork, Pattern Classification, Wiley-Interscience, 2001.

[6] J. H. Clark, C. Dyer, A. Lavie, N. A. Smith, Better hypothesis testing for statistical machine translation: Controlling for optimizer instability, in: Proc. of the annual meeting of the ACL, 2011, pp. 176–181.

[7] L. Nepveu, G. Lapalme, P. Langlais, G. Foster, Adaptive language and translation models for interactive machine translation, in: Proc. of EMNLP, 2004, pp. 190–197.

[8] J. Civera, A. Juan, Domain adaptation in statistical machine translation with mixture modelling, in: Proc. of the ACL 2nd workshop on SMT, 2007, pp. 136–158.

[9] B. Zhao, M. Eck, S. Vogel, Language model adaptation for statistical machine translation with structured query models, in: Proc. of CoLing, 2004, pp. 411–417.

[10] G. Sanchis-Trilles, M. Cettolo, N. Bertoldi, M. Federico, Online Language Model Adaptation for Spoken Dialog Translation, in: Proc. of the international Workshop on SLT, 2009, pp. 160–167.

[11] G. Gascó, V. Alabau, J. Andrés-Ferrer, J. González-Rubio, M.-A. Rocha, G. Sanchis-Trilles, F. Casacuberta, J. González, J.-A. Sánchez, Iti-upv system description for iwslt 2010, in: Proc. of the international Workshop on SLT, 2010, pp. 85–92.

[12] A. Axelrod, X. He, J. Gao, Domain adaptation via pseudo in-domain data selection, in: Proc. of EMNLP, 2011, pp. 355–362.

[13] S. Matsoukas, A.-V. I. Rosti, B. Zhang, Discriminative corpus weight estimation for machine translation, in: Proc. of EMNLP, 2009, pp. 708–717.

[14] G. Foster, C. Goutte, R. Kuhn, Discriminative instance weighting for domain adaptation in statistical machine translation, in: Proc. of EMNLP, 2010, pp. 451–459.

[15] K. Duh, K. Sudoh, T. Iwata, H. Tsukada, Alignment inference and bayesian adaptation for machine translation, in: Proc. of the MT Summit XIII, 2011, pp. 19–23.

[16] K. Yu, M. J. Gales, Bayesian adaptation and adaptively trained systems, in: Proc. of 2005 IEEE Workshop on ASRU, 2005, pp. 209–214.

[17] K. Yu, M. Gales, Incremental adaptation using bayesian inference, in: Proc. of ICASSP, 2006, pp. 217–220.

[18] F. Valente, C. J. Wellekens, Variational bayesian adaptation for speaker clustering, in: Proc. of ICASSP, 2005, pp. 965–968.

[19] G. Sanchis-Trilles, F. Casacuberta, Log-linear weight optimisation via bayesian adaptation in statistical machine translation, in: Proc. of CoLing, 2010, pp. 1077–1085.

[20] M. Carpuat et al., Domain adaptation in machine translation: Final report, in: 2012 Johns Hopkins Summer Workshop Final Report, 2012.
URL `http://hal3.name/damt/`

[21] F. Och, Minimum error rate training for statistical machine translation, in: Proc. of the 41st Annual Meeting of the ACL, 2003, pp. 160–167.

[22] C. Cherry, G. Foster, Batch tuning strategies for statistical machine translation, in: Proc. of NAACL, 2012, pp. 427–436.

[23] M. Hopkins, J. May, Tuning as ranking, in: Proc. of EMNLP, 2011, pp. 1352–1362.

[24] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[25] P. Koehn et al., Moses: Open source toolkit for statistical machine translation, in: Proc. of the annual meeting of the ACL: Demo and Poster Sessions, 2007, pp. 177–180.

[26] W. Hastings, Monte carlo sampling methods using markov chains and their applications, Biometrika 57 (1) (1970) 19–109.

[27] P. Koehn, A. Axelrod, A. B. Mayne, C. Callison-Burch, M. Osborne, D. Talbot, Edinburgh system description for the 2005 iwslt speech translation evaluation, in: Proc. of the international Workshop on SLT, 2005.

[28] R. Kneser, H. Ney, Improved backing-off for m-gram language modeling., in: Proc. of ICASSP, 1995, pp. 181–184.

[29] A. Stolcke, SRILM – an extensible language modeling toolkit, in: Proc. of the 7th Intl. Conf. on Spoken Language Processing, 2002, pp. 901–904.

[30] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, J. Makhoul, A study of translation edit rate with targeted human annotation, in: Proc. of conf. of the AMTA, 2006, pp. 223–231.

[31] K. Papineni, A. Kishore, S. Roukos, T. Ward, W. J. Zhu, Bleu: A method for automatic evaluation of machine translation, in: Technical Report RC22176 (W0109-022), 2001.