Mach Translat (2014) 28:309–339
DOI 10.1007/s10590-014-9159-7

Online adaptation to post-edits for phrase-based statistical machine translation

Nicola Bertoldi · Patrick Simianer · Mauro Cettolo · Katharina Wäschle · Marcello Federico · Stefan Riezler

Received: 3 February 2014 / Accepted: 20 September 2014 / Published online: 20 November 2014

Abstract Recent research has shown that accuracy and speed of human translators can benefit from *post-editing* output of machine translation systems, with larger benefits for higher quality output. We present an efficient online learning framework for adapting all modules of a phrase-based statistical machine translation system to post-edited translations. We use a constrained search technique to extract new phrase-translations from post-edits without the need of re-alignments, and to extract phrase pair features for discriminative training without the need for surrogate references. In addition, a cache-based language model is built on n-grams extracted from post-edits. We present experimental results in a simulated post-editing scenario and on field-test data. Each individual module substantially improves translation quality. The modules can be implemented efficiently and allow for a straightforward stacking, yielding significant additive improvements on several translation directions and domains.

N. Bertoldi (✉) · M. Cettolo · M. Federico
FBK - Fondazione Bruno Kessler, via Sommarive 18, Povo, 38123 Trento, Italy
e-mail: bertoldi@fbk.eu

M. Cettolo
e-mail: cettolo@fbk.eu

M. Federico
e-mail: federico@fbk.eu

P. Simianer · K. Wäschle · S. Riezler
Department of Computational Linguistics, Heidelberg University, 69120 Heidelberg, Germany

P. Simianer
e-mail: simianer@cl.uni-heidelberg.de

K. Wäschle
e-mail: waeschle@cl.uni-heidelberg.de

S. Riezler
e-mail: riezler@cl.uni-heidelberg.de

Keywords Statistical machine translation · Post-editing · Online adaptation

1 Introduction

Worldwide demand of translation services has dramatically accelerated in the last decade, as an effect of market globalization and the growth of the Information Society. Computer assisted translation (CAT) tools are currently the dominant technology in the translation and localization market, and those including machine translation (MT) engines are on the increase. Although MT systems are not yet able to provide output that is suitable for publication without human intervention, recent achievements in the field have raised new expectations in the translation industry. Several empirical studies (Federico et al. 2012; Green et al. 2013; Läubli et al. 2013) have recently shown that significant productivity is gained when professional translators *post-edit* MT output instead of translating from scratch. So far, however, MT has focused on providing ready-to-use translations, rather than outputs that minimize the effort of a human translator. Our approaches focus on the latter. In fact, a very important issue for research and industry is how to effectively integrate machine translation within CAT software.

State-of-the-art statistical MT systems translate each sentence of an input text in isolation. While this reduces the complexity of translating large documents, it introduces the problem that information beyond the sentence level is lost. As mentioned in Tiedemann (2010), for instance, there are two types of important properties in natural language and translation that are often ignored in statistical models: consistency and repetitiveness. In the post-editing scenario additional information beyond the sentence level is available. After producing a system translation, user feedback in the form of a manual translation or of a user correction is received, which can be exploited to refine the next translations. From the viewpoint of professional translators, immediate refinement of the MT system in response to user post-editing is crucial in order to offer the experience of a system that learns from feedback and corrections. Online adaptation achieves this by increasing consistency of system translations with respect to the user translation of previously seen examples.

The post-editing scenario[1] fits well into an online learning protocol (Cesa-Bianchi and Lugosi 2006), where a stream of input data is revealed to the learner one by one. For each input example, the learner must make a prediction, after which the actual output is revealed, which the learner can use to refine the next prediction. Online learning can be applied to the framework of online adaptation by weighting new inputs heavier than older ones. It is important to stress that the potential of any online adaptation technique can be effectively exploited for texts, like for instance in technical documents, where repetition of content words is very common.

This paper compares approaches to online adaptation of phrase-based statistical MT systems (Koehn 2010) by building on and significantly extending recent works

[1] We will use the term post-editing instead of the more generic term CAT when our attention is focused on the interaction between human translator and MT system, disregarding the influence of other components available in a CAT tool, such as translation memories and terminology dictionaries.

by Wäschle et al. (2013) and Bertoldi et al. (2013), which were the first attempt to apply both generative and discriminative online adaptation methods in a post-editing setting.

First, we propose methods that augment the generative components of the MT system, translation and language model, straightforwardly by building cache-based local models of phrase pairs and n-grams from user feedback. Then, we present a discriminative method based on a structured perceptron to refine a feature-based re-ranking module applied to the k-best translations of the MT system. The generative and discriminative approaches are independent and can be straightforwardly cascaded.

A deep investigation and comparison of the proposed adaptation techniques have been conducted in three domains, namely information technology, legal documents, and patents, and for several language pairs, namely from English into Italian, Spanish, and German, and from German into English. In sum, the gains of the generative and discriminative approaches come to improvements up to 10 absolute BLEU points over a non-adaptive system.

The paper is organized as follows. Section 2 overviews previous research on caching techniques, online learning, and discriminative re-ranking in machine translation. Section 3 presents the post-editing workflow from an online learning perspective. Section 4 describes the proposed generative and discriminative approaches to online adaptation of a MT system, and provides details on their implementation as well. Section 5 provides details about the translation directions and tasks we considered in our experimental evaluation; Sect. 6 reports on the experiments that we conducted, including a discussion of their outcomes. Some final summarizing comments end the paper in Sect. 7.

2 Related research

The concept of caching arose in computer science in the '60s, when it was introduced to speed up the fetching of instructions and data and the virtual-to-physical address translation. In caching, results of computations are stored transparently so that future requests for them can be served faster. Caches exploit the locality of reference (principle of locality): the same value or related storage location is frequently accessed. This phenomenon does not only occur in computer science, but also in natural language, where the short-term shifts in word-use frequencies is empirically observed and was the rationale behind the introduction of the cache component in statistical language models by Kuhn and De Mori (1990). In this case, the argument was not efficiency like for computers but improving the prediction capability of the model; caching has also been used for time savings in the concrete implementation of language models (Federico et al. 2008).

The use of caching in MT was introduced by Nepveu et al. (2004), with the goal of improving the quality of both translation and language models in the framework of interactive MT; the approach includes automatic word alignment of source and post-edited target, namely the IBM model 2 Viterbi search. Tiedemann (2010) proposed to incrementally populate the translation model cache with the translation options used by the decoder to generate the final best translation; no addi-

tional alignment step is required here. Our cache-based translation model stands in between these two approaches: The cache is filled with phrase pairs from the previous post-edit session. An explicit, possibly partial, phrase-alignment is obtained via an efficient constrained search, fed by all translation options whose source side matches the sentence to translate. We propose a further enhancement of the basic caching mechanism for rewarding cached items related to the current sentence to translate.

Our work is also related to MT adaptation in general and online learning in particular. Online learning methods in statistical MT are found in the context of stochastic methods for discriminative training (Liang et al. 2006; Chiang et al. 2008), or streaming scenarios for incremental adaptation of the core components of MT (Levenberg et al. 2010, 2012). However, the online learning protocol is applied in these approaches to training data only, i.e., parameters are updated on a per-example basis on the training set, while testing is done by re-translating the full test set using the final model. In an online adaptation framework, an important aspect is the evaluation of a dynamic system, which should consider not only the overall average performance, but also its evolution over time. Bertoldi et al. (2012) proposed the *Percentage Slope* as an effective measure for the system learning capability. Further related work can be found in the application of incremental learning to domain adaptation in MT. Here a local and a global model have to be combined, either in a (log)-linear combination (Koehn and Schroeder 2007; Foster and Kuhn 2007), with a fill-up method (Bisazza et al. 2011), or via ultraconservative updating (Liu et al. 2012).

Various structured learning techniques have been applied to online discriminative re-ranking in a post-editing scenario, for example, by Cesa-Bianchi et al. (2008), Martínez-Gómez et al. (2012), or López-Salcedo et al. (2012). Incremental adaptations of the generative components of MT have been presented for a related scenario, interactive machine translation, where an MT component produces hypotheses based on partial translations of a sentence (Nepveu et al. 2004; Ortiz-Martínez et al. 2010). Our online learning protocol is similar, but operating on the sentence instead of word or phrase level.

Incremental adaptations have also been presented for larger batches of data (Bertoldi et al. 2012). In terms of granularity, our scenario is most similar to the work by Hardt and Elming (2010), where the phrase-based training procedure is employed to update the phrase table immediately after a reference becomes available. Our work, however, focuses on adapting both language and translation models with techniques that combine small adaptive local models with large static global models. This feature in fact nicely fits with the typical use of CAT tools, in which users use both a global shared translation memory and a local private translation memory.

In parallel to our work, Denkowski et al. (2014) developed an approach to learning from post-editing that allows independent adaptation of translation grammar, language model, and discriminative parameters of the statistical MT model. The results are not directly comparable because of the use of different corpora. However, while their approach differs also in the proposed techniques in several respects from our work, they achieve best results by stacking all adaptation techniques, confirming a central finding in our work.

Fig. 1 Online learning procedure for the post-editing workflow

Train global model M_g
For each document d of $|d|$ segments
  Reset local model $M_d = \emptyset$
  For each example $t = 1, \ldots, |d|$
    1. Combine M_g and M_d into M_{g+d}
    2. Receive input sentence x_t
    3. Output translation $\hat{y}_t$ from M_{g+d}
    4. Receive user translation y_t
    5. Refine M_d on pair (x_t, y_t)

3 Online learning from post-editing

In the CAT workflow, source documents are split into chunks, typically corresponding to sentences, called *segments*, that are in general translated sequentially. When the translator opens a segment, the CAT tool proposes possible translation suggestions, originating from the translation memory and/or from a machine translation engine. Depending on the quality of the suggestions, the translator decides whether to post-edit one of them or to translate the source segment from scratch. Completed segments represent a valuable source of knowledge which can be readily stored in the translation memory for future use. The advantage of post-edited translations over reference translations, which are created independently by a human translator, is that post-edits are closer to actual MT translation hypotheses in terms of edit distance and domain relevance. This work addresses this issue and presents several methods which successfully improve SMT performance over time.

3.1 Online learning protocol

From a machine learning perspective, the post-editing scenario perfectly fits the *online learning* paradigm (Cesa-Bianchi and Lugosi 2006), which assumes that every time a prediction is made, the correct target value of the input is discovered right after and used to improve future predictions. We conveniently transpose the above concept to our post-editing scenario as depicted in Fig. 1. The learning process starts from training a global model M_g on parallel data in the range of millions of sentence pairs. Then for each document d, consisting of a few tens up to a thousand segments, an empty local model M_d is initialized. For each example, first the static global model M_g and the current local model M_d are combined into a model M_{g+d} (step 1). Next the received input x_t (step 2) is translated into $\hat{y}_t$ using the model M_{g+d} (step 3). Then the user translation y_t is received after producing $\hat{y}_t$ (step 4). Finally the local model M_d is refined on the user feedback y_t (step 5).

This basic online learning protocol will be adapted to generative and discriminative learning components of phrase-based SMT, and extended from several directions, e.g. by adding an aging factor that scores more recent data more heavily than older data, thus accounting for online learning as online adaptation. In the following, we will use the terms online learning and online adaptation interchangeably. The evaluations reported in this paper take the local predictions $\hat{y}_t$ and compare them to the user translations y_t for each document, e.g. using $BLEU\{(\hat{y}_t, y_t)\}_{t=1}^{|d|}$ (Papineni et al. 2002). Note, that

this setup differs from the standard scenario, where the whole test set is re-translated using the learned model. However, the evaluation is still fair since only feedback from previous test set examples is used to update the current model.

3.2 Measuring the repetitiveness of a text

In Sect. 1, repetitiveness was mentioned as one of the phenomena occurring in texts that can highly affect the effectiveness of the online adaptation technique and the quality of automatic translation.

Bertoldi et al. (2013) introduced a way to measure repetitiveness inside a text, by looking at the rate of non-singleton n-grams types ($n = 1, \ldots, 4$) it contains. As shown in Bertoldi et al. (2013), this rate decays exponentially with n. For combining values with exponential decay, a reasonable scheme is to average their logarithms, or equivalently to compute their geometric mean. Furthermore, in order to make the measure comparable across differently sized documents, statistics are collected on a sliding window of 1,000 words, and properly averaged.

Formally, the repetition rate (RR) in a document can be expressed as:

$$RR = \left(\prod_{n=1}^{4} \frac{\sum_S \left(V(n) - V(n,1) \right)}{\sum_S V(n)} \right)^{1/4} \tag{1}$$

where S is the sliding window, $V(n,1)$ is the number of singleton n-grams types in S, and $V(n)$ is the total number of n-grams types in S. RR ranges between 0 and 1, where the extreme points are respectively reached when all n-grams observed in all text windows occur exactly once (RR $= 0$) and more than once (RR $= 1$). It is worth noting that using a sliding window, in addition to permiting the comparison of RR of texts of different sizes, allows to largely maintain the linguistic features of the original text, as opposed to what would happen if the sentences to be processed together were randomly sampled.

3.3 Measuring the learning capability of a dynamic system

The standard MT metrics, such as BLEU (Papineni et al. 2002) and TER (Snover et al. 2006), provide absolute performance of the system, but they do not fit well into an online adaptation scenario, in which the system evolves dynamically over time. As shown in Bertoldi et al. (2012), adapting systems can be effectively analyzed by means of the *Percentage Slope* (henceforth *Slope*), which measures their learning capability. This metric originates in the industrial environment to evaluate the efficiency gained when an activity is repeated. Slope expresses the rate of learning on a scale of 0–100 %. A 100 % Slope represents no learning at all, zero percentage reflects a theoretically infinite rate of learning. In practice, human operations hardly ever achieve a rate of learning faster than 70 % as measured on this scale.

The Percentage Slope is actually a meta-metric $Slope(metric)$, because it relies on an external metric measuring the efficiency of the activity in a range between 0 and 100, and fulfilling the constraint that "the lower, the better". More details on its

Segment	Source Text	Post-edit
#6	*Annex to the Technical Offer*	*Allegato all' Offerta Tecnica*
...	...	...
#39	*This Technical Offer adopts several graphic notations*	*Nella presente Offerta Tecnica vengono adottate alcune notazioni grafiche*

Fig. 2 Two segments occurring in different positions of a sample document containing the same fragment "*Technical Offer*"; the user post-edit, including its correct translation "*Offerta Tecnica*", is shown in the third column. These sentences are used throughout the paper to illustrate the proposed approaches

computation can be found in Bertoldi et al. (2012). From a practical point of view, as suggested by the authors, the sequence of scores are computed while the adapting system is being used; the learning curve which best matches the sequence is then found and eventually Slope is computed.

It is crucial to stress that the main assumption in the definition of the Percentage Slope metric is that the difficulty of the activity remains constant. Actually, this is not true in our scenario, because the difficulty of translating different portions of a text can vary a lot. Nevertheless, this metric is still useful to evaluate the learning capability of a dynamic system. It is sufficient to consider the performance difference between the dynamic system and the static (non-adaptive) system, taken as reference, to remove the effects of the intrinsic variable difficulty of the text. We hence computed a Percentage Slope on the difference of BLEU achieved by the dynamic and static systems, and named it $\Delta Slope$ to make such a difference clear. More precisely, as a metric is required which decreases when efficiency increases, we defined $\Delta Slope = Slope(100 - (BLEU(dynamic) - BLEU(static))$. The dynamic system has learning power if $\Delta Slope$ is below 100 %.

4 Online adaptation in SMT

We present several techniques for refinements of local MT models (step 5 in Fig. 1), namely adaptations of the generative components of translation model (TM) (Sects. 4.3.1 and 4.3.2) and language model (LM) (Sect. 4.4) and adaptation via discriminative re-ranking (Sect. 4.5). Different refinements result in different modes of combination of *global* and *local* models (step 1). Both generative and discriminative adaptation modes deploy a constrained search technique (Sect. 4.2) to extract information relevant for system refinement from the received user feedback (step 4). Translation (step 3) employs a standard phrase-based MT engine, briefly introduced in Sect. 4.1.

Our adaptation techniques are exemplified throughout this Section by the two segments #6 and #39 of a sample document in the Information Technology domain[2] reported in Fig. 2. In these two segments, the phrase "*Technical Offer*" occurs twice, and the user chooses a common translation "*Offerta Tecnica*" for it. As shown in Fig. 3, the baseline non-adaptive MT system repeatedly produces an incorrect translation; the user, who relies on the MT suggestions to post-edit the segments, is forced to correct

[2] The sample document is set0 of the English–Italian Information Technology task; see Sect. 5 for more details.

Segment	Source Text	MT output
#6	*Annex to the Technical Offer*	*Allegato a Technical offerta*
...	...	...
#39	*This Technical Offer adopts several graphic notations*	*Questa Technical offerta utilizza diversi notazioni grafica*

Fig. 3 The output of both sample segments generated by the baseline system contain the same error "*Technical offerta*" for the expression "*Technical Offer*"

the error twice. The final goal of our adaptation approaches is to learn the right translation from the user feedback on the first segment and suggest this in the second, in order to reduce the post-editing effort of the user.

4.1 Baseline system

The MT engine is built with the open source toolkit Moses (Koehn et al. 2007). The 4-score global translation and the 6-score lexicalized reordering models are estimated on parallel training data with default settings. The global 5-g LM is smoothed by the improved Kneser–Ney technique, and estimated on the target monolingual side of the parallel training data using the IRSTLM toolkit (Federico et al. 2008). Models are case-sensitive. Moreover, word and phrase penalties, and a distance-based distortion model are employed as features. The log-linear interpolation weights are optimized using the standard Minimum Error Rate Training (MERT) procedure (Och 2003) provided with the Moses toolkit. As suggested by Cettolo et al. (2011), weights are estimated averaging three distinct optimization runs to reduce the instability of MERT (Clark et al. 2011). The baseline system also provides a list of k-best translations, which are exploited by the online discriminative re-ranking module.

4.2 Constrained search for feedback exploitation

In order to extract information for system refinement from user feedback, source and user translation need to be aligned at the phrase-level. We use a constrained search technique described in Cettolo et al. (2010) to achieve this, which optimizes the coverage of both source and target sentences given a set of translation options.

The search produces exactly one phrase segmentation and alignment, and allows gaps such that some source and target words may be uncovered. Unambiguous gaps (i.e. one on the source and one on the target side) can then be aligned. It differs in this respect from forced decoding, which produces an alignment only when the target is fully reachable with the given models.

From the phrase alignment, three types of phrase pairs can be collected: (i) **new** phrase pairs by aligning unambiguous gaps; (ii) **known** phrase pairs already present in the given model; (iii) **full** phrase pairs consisting of the complete source sentence and its user translation. Since preliminary experiments showed that using cached phrases consisting only of function words had a negative impact on translation quality, we restricted the phrase extraction to phrases that contain at least one content word.[3] We

[3] We used language-specific stop word lists to filter for content words, making our approach slightly language-dependent.

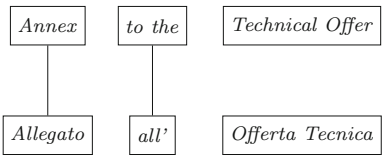

Fig. 4 Phrase segmentation and alignment obtained applying the constrained search algorithm to the sample segment #6

found that in our experimental data cached phrases tended to be between one and three words in length, so we limited the length of **new** and **known** phrases to four words to speed up the annotation. The full phrase pair, which can have any length, is added to mimic the behaviour of a translation memory.

From the alignment shown in Fig. 4, we extract the new phrase pair *Technical Offer* → *Offerta Tecnica*, the known phrase pairs *Annex* → *Allegato* and *to the* → *all'* and the full phrase *Annex to the Technical Offer* → *Allegato all' Offerta Tecnica*. Then, *to the* → *all'* is actually discarded, because it does not contain any content word.

4.3 TM adaptation

The growing collection of source sentences and corresponding user-approved translations enables the construction of a *local* translation model. The goal of this local model is to reward MT translations that are consistent with previous user translations as well as to integrate new translations learned from user corrections, in order to better translate the following sentences.

We compare two alternative approaches to include the growing set of phrase pairs into the translation models: the former is based on a cache external to the decoder, and exploits the Moses xml-input option; the latter relies on a new Moses feature implementing an internal cache.

4.3.1 TM adaptation with an external cache

From each sentence pair, all phrase pairs extracted with the constrained search technique described in Sect. 4.2 are inserted into a cache; a score for each phrase pair is estimated based on the relative frequency of the target phrase given the source phrase in the cache, as described in Eq. 2.

$$\begin{aligned} score(f,e) &= \frac{c_{all}(f,e)}{c_{all}(f)} \\ c_{all}(f,e) &= \lambda_{new} \cdot c_{new}(f,e) + \lambda_{known} \cdot c_{known}(f,e) + \lambda_{full} \cdot c_{full}(f,e) \\ c_{all}(f) &= \sum_{e} c_{all}(f,e) \end{aligned} \tag{2}$$

where $c_{new}(f,e)$, $c_{known}(f,e)$, and $c_{full}(f,e)$ are the frequencies of (f,e) in the corresponding groups of extracted phrase pairs; $c_{all}(f,e)$ is their linear combination with individual weights λ; $c_{all}(f)$ is the marginal weighted frequency of all translation options e for a given f. Cache and model are updated on a per-sentence basis as soon

#39 | *This* `<phrase translation="`*Offerta Tecnica*`||`*Proposta Tecnica*`" prob="0.75||0.25">`*Technical Offer*`</phrase>` *adopts several graphic notations*

Fig. 5 Annotation of the sample segment #39 when TM adaptation via external cache is applied and the cache contains two translation options "*Offerta Tecnica*" and "*Proposta Tecnica*" for "*Technical Offer*", with frequency 3 and 1, respectively

as source sentence and user translation become available. Details about the estimation of λ parameters are given in Sect. 6.1.

A fast way to integrate the constantly changing local model in the decoder at run-time is the Moses xml-input option.[4] Translation options can be passed to the decoder via xml-like markup. Through this modality, the options are temporarily added to the global translation model. They become entries of the model in effect, and hence at decoding time can be accessed like those already inside. Note that the temporary options are automatically deleted after the translation of the current sentence. In the global translation model each entry is usually associated with multiple feature scores (four in our setting); hence, the $score(f, e)$ (Eq. 2) assigned to each option passed via xml-like markup is split uniformly. Multiple translation options and their corresponding probabilities can be suggested for a specific source phrase.

Moses offers two ways to interact with this local phrase table. In *inclusive* mode, the given phrase translations compete with existing phrase table entries, as though they were temporarily added to the baseline phrase table. The decoder is instead forced to choose only from the given translations in *exclusive* mode.[5] During development, we found that the exclusive option is too strict in our scenario. Though most phrase pairs are correct and useful additions, for example spelling variants such as *S.p.A*→*SpA* or domain vocabulary such as *lease payment*→*canone*, some are restricted to a specific context, e.g. translation from singular to plural such as *service*→*servizi*, and some are actually incorrect. In inclusive mode, the global translation and language model can reject unlikely translations.

Since the xml-input option does not support overlapping phrases, sentences are annotated in a greedy way from left to right. For each phrase in the input sentence, the cache is checked for possible translations, starting from the complete sentence down to single words. In this way, translations for larger spans are preferred over word translations. We did not explore other setups, such as preferring newly learned phrases over older options from the cache, but instead opted to keep the implementation simple.

Figure 5 illustrates the annotation to pass translation options to the decoder through the xml-like markup. The scores 0.75 and 0.25 associated with the two options are computed according to their frequency in the cache.

[4] Details about the xml-input option can be found in the Moses official documentation (http://www.statmt.org/moses/manual/manual.pdf).

[5] The usage of either modes do not introduce new features.

```
<dlt cbtm="Annex to the Technical Offer . ||| Allegato all' Offerta Tecnica .
          |||| Technical Offer ||| Offerta Tecnica|||| Annex ||| Allegato"/>
```

Fig. 6 Annotation to update the local translation model when the internal cache is applied; in this example we feed the decoder with the phrase pairs extracted from segment #6

4.3.2 TM adaptation with an internal cache

In this second approach for generative adaptation, the local translation model is implemented as an additional 1-score phrase table,[6] i.e. an additional feature providing one score. This model dynamically changes over time in two respects: (i) new phrase pairs can be inserted, and (ii) scores of all entries are modified when new pairs are added.

All entries are associated with an *age*, corresponding to the time they were inserted, and scored accordingly. Each new insertion causes the ageing of the existing phrase pairs and hence their re-scoring; in case of re-insertion of a phrase pair, the old value is overwritten. Each phrase pair is scored according to its actual age in the cache, hence it varies over time; the score is computed with the following function[7]:

$$score(age) = 1 - exp\left(\frac{1}{age}\right) \tag{3}$$

From each sentence pair, phrase pairs are extracted with the procedure used in the previously described approach, and simultaneously added to the local translation model by feeding the decoder with an annotation illustrated in Fig. 6. All options in the example are inserted simultaneously in the cache and hence they are associated with the same age.

During decoding, translation alternatives are searched both in the global static phrase table and in the local cache-based (dynamic) phrase table, get a score from both tables, and compete in the creation of the best translation hypothesis.

4.4 LM adaptation

Similar to the local cache-based translation model described in Sect. 4.3.2, a *local* cache-based language model is built to reward the n-grams found in post-edited translations. This model is implemented as an additional feature of the log-linear model, which provides a score for each n-grams; the feature relies on a cache storing target n-grams.

The same policy employed in the modification of the local cache-based translation model is applied to the local cache-based language model. The score is also computed according to Eq. 3 and depends on the actual age of the n-grams in the cache.

[6] The source code of the local cache-based translation model and language model is available in the branch "dynamic-models" under the official GitHub repository of Moses toolkit, directly accessible from this URL: https://github.com/moses-smt/mosesdecoder/tree/dynamic-models.

[7] Other scoring functions have been tested in preliminary experiments, but no significant differences were observed. Details of additional scoring functions as well as usage instructions can be found in (Bertoldi 2014).

<dlt cblm="*Allegato all' Offerta Tecnica || Allegato all' Offerta || Allegato all' || Allegato || all' Offerta Tecnica || all' Offerta || Offerta Tecnica || Offerta || Tecnica*"/>

Fig. 7 Annotation to update the local language model; in this example we feed the decoder with the n-grams extracted from segment #6

Only the annotation slightly changes as shown in Fig. 7. As with the cache-based translation model, we found in preliminary experiments that discarding stopword phrases improved results, so for each user-approved translation y, all its n-grams ($n = 1, \ldots, 4$) containing at least one content word are extracted (for an example, see Fig. 7) and inserted in the cache.

At decoding time, the target side of each translation option fetched by the search algorithm is scored with the cached model according to the same policy applied for the local cache-based translation model. If the target is not found in the cache, it receives no reward. Note that n-grams crossing over contiguous translation options are not taken into account by this model.

It is worth emphasizing that, despite the name, the proposed additional feature is not a conventional language model, but rather a function rewarding approved high-quality word sequences in target sides of phrase pairs.

4.5 Online discriminative re-ranking

Our discriminative re-ranking approach[8] is based on the structured perceptron by Collins (2002), which fits nicely into the online scenario considered here: For each source sentence the baseline system is asked to generate a k-best list of hypotheses. This list is ranked according to the current linear re-ranking model and its prediction is returned. Then, the learner receives the user translation, which is used for parameter updating. Updates occur, if the prediction of the re-ranking differs from the user translation. More formally, given a feature representation $f(x, y)$ for a source-target pair (x, y), and a corresponding weight vector w, the perceptron update on a training example (x_t, y_t) where the prediction $\hat{y} = \arg\max_y \langle w, f(x_t, y)\rangle$ does not match the target y_t is defined as:

$$w = w + f(x_t, y_t) - f(x_t, \hat{y}) \tag{4}$$

We use lexicalized sparse features defined by the following feature templates: All phrase pairs used by the decoder (for system translations) or given by the constrained search (for the user translation) are used as features; in addition, we use target-side n-grams ($n = 1, \ldots, 4$) as features, extracted from the user translation or the system translation, respectively. All features are simple indicator functions, with feature values given by the number of source words for phrase pairs, or n for target-side n-grams. This way, more weight is put on phrases spanning longer parts of the source sentence and higher order n-grams. During development we found this to have a positive effect on BLEU results, but as shown in

[8] An implementation is available from https://github.com/pks/bold_reranking.

the experiments section, this may result in worsening of other metrics such as TER. Considering the example in Fig. 4, the following features are extracted: *Annex* → *Allegato* with a feature value of 1, *Offerta Tecnica* → *Technical Offer*[9] with feature value 2, and all n-grams found in the English translation. As for the TM adaptations we also only consider features that include at least one content word.

The advantage of using only the two above described feature templates in discriminative reranking are as follows. First, a major advantage of this approach is its simplicity as there is no need for interaction with the decoder. The decoder is only required to return a k-best list of translation hypotheses along with phrase segmentations. This way, a wide variety of decoders can be used with the re-ranking module, which may be beneficial for practical use. Second, the combination of decoder-independent features with the constrained search technique allows us to apply an update condition that can be categorized as *bold* in terms of Liang et al. (2006). That is, for the purpose of discriminative training, in our setup all references are effectively reachable since we can extract features from them and assign model scores.

4.6 Tuning of the dynamic systems

The described techniques for TM adaptation via internal cache and LM adaptation add one more feature each to the standard feature set of the baseline system. Optimization of the additional feature together with the standard ones is achieved by a modification of the standard MERT procedure provided by Moses. The translation of the input text is now performed sequentially instead of in parallel; practically, the batch translation of the standard MERT procedure is replaced by the online process introduced in Sect. 3 to update the dynamic models sentence after sentence. This enhanced MERT procedure permits to reliably tune the weight of the additional feature as well as those of the standard feature set.

The overall dynamic system has meta-parameters related to the constrained search step and to the selection of the features for updating the dynamic models. Tuning of these parameters is described in Sects. 6.1 and 6.2.

5 Data benchmarks

Experimental analysis of the systems was performed on several tasks on both proprietary and publicly available data, involving the translation of documents from three domains, namely information technology, law, and patents, and for several language pairs, namely from English into Italian, Spanish, and German, and from German into English. Experiments were carried out by simulating post-editing feedback with the available reference translations, as proposed by Hardt and Elming (2010). Note that, for two tasks (English–Italian, information technology and legal domains), the used

[9] This phrase pair feature can only be used if the re-ranking is combined with one of the TM adaptations, as it is a newly created phrase pair.

Table 1 Statistics for the training data for all tasks: number of segments, source and target running words

Task	Segments	Running words	
		Source	Target
English–Italian IT	1.2	18.8	19.2
English–Italian legal	2.3	55.7	57.5
English–Spanish legal	2.3	56.1	62.0
English–German patents	4.2	160.9	130.6

Figures (in millions) refer to tokenized text

reference translations correspond to actual post-edits made by professional translators working with a non-adaptive MT system.

5.1 Training data

For the English–Italian Information Technology task training data mostly consist of commercial data extracted from a translation memory built during a number of translation projects for several commercial companies; these data were provided by Translated srl, the industrial partner of the MateCat project, and were collected during the real use of CAT tools by professional translators. In addition, parallel texts from the OPUS corpus[10] were are also included (Tiedemann 2012).

For the other tasks, public data, which allow replicability and cross-assessment of our outcomes, were chosen in order to cover a wide range of linguistic conditions. For the English–Italian and English–Spanish Legal tasks training data are taken from version 3.0 of the JRC-Acquis[11] collection (Steinberger et al. 2006).

For the English–German and German–English Patents tasks training texts consist of patent text sampled from title, abstract and description sections from the PatTR[12] corpus (Wäschle and Riezler 2012).

Statistics for the training corpora are reported in Table 1.

5.2 Development and test data

In this paper we aim at comparing the proposed adaptation techniques across different domains, language pairs, text repetitiveness, reference types, and overall performance of the baseline system. To this purpose we collected several documents for each task employed for either development or testing. Main statistics are shown in Table 2: number of segments, number of source and target running words, and source and target repetition rate (RR) computed as explained in Sect. 3.2. Figures on the source side refer to the texts the users are requested to translate; figures on the target side refer to either the translations or the actual post-edited texts; all figures refer to tokenized texts. When references consist of post-edits, they are created by translators modifying the suggestions of a static SMT system.

[10] http://opus.lingfil.uu.se.

[11] http://langtech.jrc.it/JRC-Acquis.html.

[12] http://www.cl.uni-heidelberg.de/statnlpgroup/pattr.

Table 2 Statistics of the development and test data for all tasks: number of segments, running words and repetition rate for both source and target sides

Task	set	Segments	Running words		RR	
			Source	Target	Source	Target
English–Italian IT	set0	420	7,307	7,416	13.3	12.0
	set1	931	11,627	13,198	16.7	16.9
	set2	375	8,488	9,227	14.1	12.1
	set3	289	4,005	4,147	12.9	12.0
	set4	1,000	10,539	11,917	16.0	15.8
	set5	864	11,338	11,398	11.7	10.9
	set6A	160	2,988	3,248	9.4	8.4
	set6B	160	2,988	3,331	9.5	8.2
	set6C	160	2,988	3,252	9.7	10.4
	set6D	160	2,988	3,326	9.9	8.9
	set7A	176	3,062	3,389	14.0	13.7
	set7B	176	3,062	3,329	15.2	12.4
	set7C	176	3,062	3,367	13.8	12.7
	set7D	176	3,062	3,441	12.5	12.9
English–Italian legal	set0	551	28,974	31,337	11.6	12.5
	set1	514	27,546	29,831	11.8	12.8
	set2	571	17,798	17,087	18.2	16.7
	set3A	91	2,957	3,211	9.4	7.5
	set3B	91	2,957	3,237	8.5	7.9
	set3C	91	2,957	3,358	8.8	9.0
	set3D	91	2,957	3,218	9.7	7.4
English–Spanish legal	set0	551	29,472	34,345	11.8	14.8
	set1	514	27,928	32,665	11.9	14.9
	set2	573	17,798	18,800	18.2	18.1
English–German patents	set0	318	12,568	9,449	15.7	8.9
	set1	304	8,481	6,290	13.3	6.9
	set2	222	11,539	9,245	21.1	15.6
	set3	300	9,286	7,269	17.1	9.4
	set4	227	8,475	7,320	19.8	13.8
	set5	239	8,429	6,685	16.6	11.6
	set6	232	9,055	6,974	17.7	11.9
	set7	230	7,289	6,008	19.3	14.4
	set8	225	6,000	5,049	16.9	10.0
	set9	231	7,285	5,782	19.2	14.1

Figures refer to tokenized text

5.2.1 English–Italian IT

Six documents labeled set0–5 correspond to projects provided by Translated srl, where the references are user-approved translations. Documents set6 and set7 are taken from a software user manual. For each sentence, the actual user corrections by four different translators (A–D) were collected during a field test and used as references. We report the scores for all four translators, regarding each translator's post-edits as an independent document.

This choice has strong motivations in the online adaptation scenario. Each translator processed the sentences in his/her preferred order and provided a different reference; hence, the original baseline system evolves differently, and possibly achieves different performance. In addition, since sentence order and references differ among documents set6A–D and set7A–D, both source and target repetition rates vary, as explained in Sect. 3.2.

Weight optimization was performed on documents set0–2 for each system independently. During the tuning procedure of the dynamic systems, their caches were cleared at the beginning of each document in order to avoid possibly uncontrolled interactions among them.

5.2.2 English–Italian and English–Spanish legal

For both the English–Italian and English–Spanish task, three documents were selected by exploiting the labeling of JRC-Acquis documents in terms of Eurovoc[13] subject domain classes. We chose two classes including a not too large nor too small number of documents (around 100), and three documents were selected from each class.[14] For fairness, all other documents of those classes have been removed from the training data.

For English–Italian only, an additional document set3, taken from a recent motion for a European Parliament resolution published on the EUR-Lex platform, was translated by four different translators (A–D) during a field test, hence four independent post-edits were used as user feedback.

Document set0 was used for development, the remaining documents for testing. Note that small variations of the source side statistics of the English–Italian and English–Spanish data sets, are due to minor differences in sentence alignment.

5.2.3 English–German and German–English Patents

For the Patents task, we selected 10 patent documents each containing a title, an abstract and a description section with a total length of more than 200 sentences. All documents

[13] http://eurovoc.europa.eu.

[14] The selected Eurovoc codes, as reported in the original documents, are 1338 and 4040. The corresponding selected documents are 32005R0713 and 52005PC0110 from class 1338, and 52005PC0687 from class 4040.

were sampled from the same IPC[15] section `E` ('Fixed Constructions'), which can be viewed as technical subdomain. Data from these documents were excluded from the training corpus.

Patents set3–5 were used for development, the remaining sets for testing. The same data sets were used for both translation directions. All data is available for download from the PatTR website.[16]

6 Experiments

In this Section we describe the detailed experimental comparison of the online adaptation approaches proposed in Sects. 4.3–4.5. The approaches to TM and LM adaptation and the discriminative re-ranking module are autonomous and can be applied independently from the others. Consequently, 12 systems could be built employing the TM adaptation via either external (`+xmtm`) or internal cache (`+cbtm`), the LM adaptation (`+cblm`), and cascading the discriminative re-ranking module (`+rnk`). The baseline system (`bsln`) does not make use of any adaptation techniques.

The fine-grained tuning of the systems was conducted on the English–Italian IT task taking into account the BLEU score (Papineni et al. 2002), and only the best performing systems were tested and compared on all other tasks, namely English–Italian and English–Spanish Legal and the English–German and German–English Patents. In particular, the meta-parameters of the TM adaptation via external cache (the weights λ of Eq. 2) and LM adaptation (the order of n-grams) techniques were estimated in this condition.

Evaluation of systems was performed by means of BLEU and TER (Snover et al. 2006), both ranging from 0 to 100. Performance of the system `bsln` on all test documents and for all tasks are shown in Fig. 8. From these plots, we observe that there is a fairly high correlation[17] between BLEU and TER scores[18] and that performance among test documents of the same task can vary a lot.

The comparison among the systems is also performed by means of ΔSlope, introduced in Sect. 3.3, which reflects the learning capability of the dynamic systems.

6.1 TM adaptation

The generative approaches to online adaptation rely on the set of phrase pairs extracted from the user feedback by means of the constrained search. As explained in Sect. 4.2, three types of phrase pairs (**new**, **known**, and **full**) are collected. We carried out preliminary experiments comparing the translation performance of all three variants. Table 3 shows performance of `+xmtm` system in terms of BLEU, under different

[15] International Patent Classification is a hierarchical patent classification scheme. Details can be found here: http://www.wipo.int/classifications/ipc/en.

[16] http://www.cl.uni-heidelberg.de/statnlpgroup/pattr/online.tar.gz.

[17] The coefficient of determination R^2 of linear regression equals to 0.92.

[18] It is worth recalling that BLEU is an accuracy metric, i.e. "the higher, the better", whereas TER is an error metric, i.e. "the lower, the better".

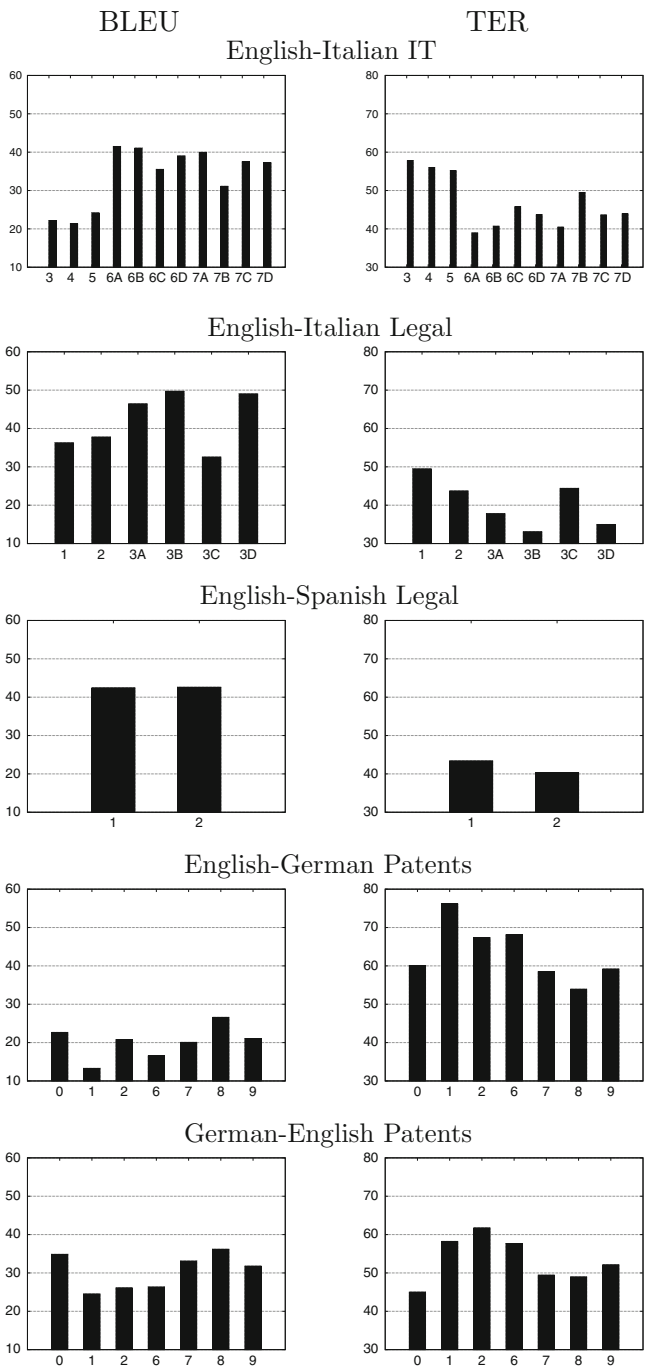

Fig. 8 Performance in terms of BLEU (*left*) and TER (*right*) of the baseline system for all tasks and all test documents

Table 3 Performance of the +xmtm system using new, known or full phrases and combinations of the three on set0–2 and set3–5 of English–Italian IT task

set	bsln	new	known	full	new +known	new +known +full
set0–2	22.59	23.11	23.73	24.22	24.33	**25.49**
		+0.52	+1.14	+1.63	+1.75	**+2.90**
		[0.57]	[0.70]	[1.73]	[0.80]	**[2.18]**
set3–5	21.49	21.64	22.24	23.07	22.42	**23.91**
		+0.15	+0.75	+1.58	+0.93	**+2.42**
		[0.06]	[0.15]	[0.91]	[0.19]	**[0.83]**

Figures reported are mean BLEU scores, mean difference and corrected sample standard deviation from baseline (in squared brackets).

conditions on set0–5 of the English–Italian IT task; the difference from the baseline are also reported. We report mean improvement over the baseline in small font size and give the corrected sample standard deviation of the improvements over all documents in the group in square brackets. Statistical significance is assessed using approximate randomization (Noreen 1989) with a significance level set to 0.05.

Each type of phrase pair in +xmtm yields individual improvements. **new** phrase pairs yield smaller improvements than **known** and **full**, which is explained by the fact that only few new phrase translations are extracted (recall, that only unambiguous gaps in the alignment are considered). The individual improvements add up to an overall statistically significant improvement of 2.90 (set0–2) and 2.42 (set3–5) BLEU points over the baseline for the combination of all conditions, namely **new + known + full**. For the combination we attempted to optimize the λ parameters in Eq. 2 using a simple grid search. However, no changes in BLEU score were observed for different weight settings, so the weights were kept uniform for all following experiments. This is in accordance with the additiveness of the different phrase pair types and an indicator for consistency of the documents: only a small number of translation options are observed for every source phrase, so the different conditions do not have to compete with each other. We use the combination of all types of phrase pairs for the second TM adaptation approach (+cbtm) as well, without individual experiments.

6.2 LM adaptation

Table 4 shows a comparison of different LM adaptation conditions applied as standalone (+cblm) and in combination with the best TM adaptation technique determined above in Sect. 6.1 (+xmtm+cblm).

The LM adaptation always outperforms the baseline regardless of the type of added n-grams; although the improvements seem consistent, they are not always statistically significant. The most important observation is that the gains achieved by the TM adaptation techniques via external cache and LM adaptation are definitely independent and additive, and the combination of the two techniques yields significant improvements.

Table 4 Performance of the +cblm system using 1-g, 4-g, and tm-*n*-g on top of the bsln and the +xmtm systems on set0–2 and set3–5 of English–Italian IT task

	bsln	+cblm			+xmtm	+xmtm+cblm		
		1-g	4-g	tm-*n*-g		1-g	4-g	tm-*n*-g
set0–2	22.59	24.06	24.38	24.18	25.49	27.32	**27.64**	26.17
		+1.47	+1.79	+1.59	+2.90	+4.73	**+5.05**	+3.58
		[1.00]	[1.48]	[0.14]	[2.18]	[1.74]	**[2.80]**	[1.43]
set3–5	21.49	22.03	22.57	22.89	23.91	24.62	**25.25**	24.73
		+0.53	+1.08	+1.40	+2.42	+3.13	**+3.76**	+3.25
		[0.91]	[1.08]	[0.37]	[0.83]	[1.92]	**[1.94]**	[1.46]

Figures reported are mean BLEU scores, mean difference and corrected sample standard deviation from baseline (in squared brackets)

Using *n*-grams up to order 4 (4-g) on top of the TM adaptation outperforms the baseline on both development sets by 5.05 and 3.76 BLEU points, corresponding to a relative improvement of around 20 %.

Rewarding only 1-g (1-g) gives the smallest improvements, indicating that more context is helpful. Using only those *n*-grams that are target sides of phrase pairs (tm-*n*-grams) added during the TM adaptation yields good improvements as stand-alone, but in combination with TM adaptation, the 4-g LM performs best. We therefore consider this the best system configuration and keep it fixed for the remaining investigation, in the case of the TM adaptation via internal cache as well.

6.3 Comparison of generative approaches

In this section we compare the five dynamic systems generated by different configuration of the TM and LM adaptation approaches. Figure 9 shows their difference in terms of BLEU and TER from the baseline system.

Apart from set6A–D, whose behavior is discussed later, similar observations can be drawn from performance of set3–5 and set7A–D. Each single adaptation approach is effective in improving baseline performance, even if gains vary a lot ranging from 1 to 8 BLEU points and from 2 to 7 TER points. Moreover, looking at Fig. 8, no correlation between the effectiveness of the online adaptation and the baseline performance can be observed. This is fortunate, as it shows that the proposed adaptation techniques are effective regardless of the absolute quality of the system to which they are applied.

When used alone, the LM adaptation technique +cblm seems less effective than both TM adaptation techniques +cbtm and +xmtm. Among the TM adaptation approaches, +cbtm outperforms +xmtm by more than 1 BLEU point if +cblm is not applied, otherwise the difference mostly vanishes; indeed, +cbtm alone achieves the best performance. The gain of the +xmtm and +cblm are instead partially additive. In our opinion, the larger effectiveness of the +cbtm approach with respect to the +xmlm approach is due to the larger freedom of the decoder in choosing the correct translation options. In the former case, the decoder is free to apply the suggested options (in the internal cache) to any source fragment; in the latter case the decoder can apply them only to the fragments greedily identified through the xml-like markup. We think that

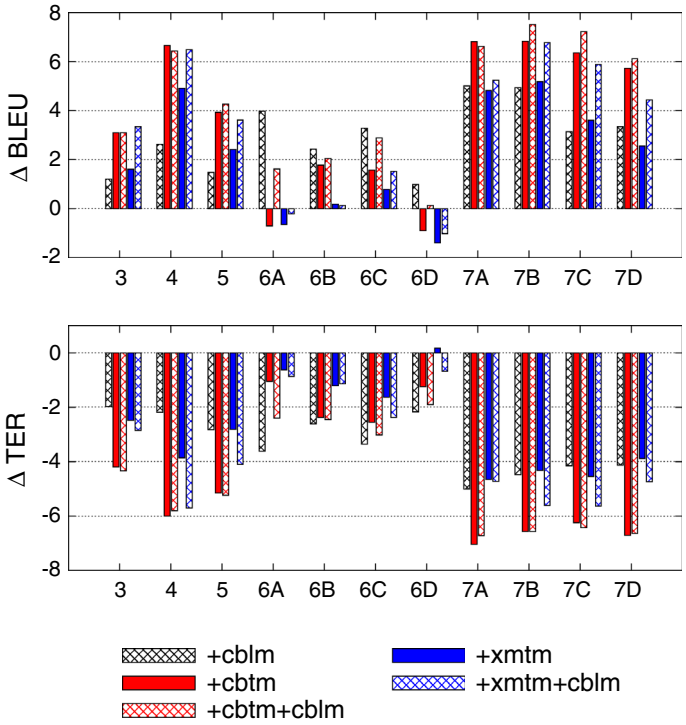

Fig. 9 Performance of the generative online adaptation approaches on all English–Italian IT documents, expressed as difference in BLEU (*left*) and TER (*right*) from the baseline system

the `+cblm` does not give an additive contribution to `+cbtm` because the latter system has already achieved the maximum possible gain. However, `+cblm` can help `+xmtm` to reach this maximum.

In summary, we find that the TM adaptation techniques outperform the LM adaptation technique with the best results achieved by their combination. Among the TM adaptation techniques, there is a slight preference for the internal-cache (`+cbtm`) over the external cache (`+xmtm`).

Concerning documents set6A–D, the LM-adapted systems clearly outperform their corresponding systems without `+cblm`, while both TM adaptation techniques fail. The main reason is likely related to the low Repetition Rate of those documents. Figure 10 is the scatter plot of Repetition Rate and BLEU performance gain from the baseline system achieved by the TM adaptation techniques (`+cbtm` and `+xmtm`) on all English–Italian IT documents; the trend lines – computed according to simple linear regression and having a coefficient of determination R^2 of 0.79 and 0.86, respectively – are evidence of a high correlation between the two measures. We conclude that it is unlikely to get any improvement by the TM-adapted system on texts like documents set6A–D, which are scarcely repetitive.

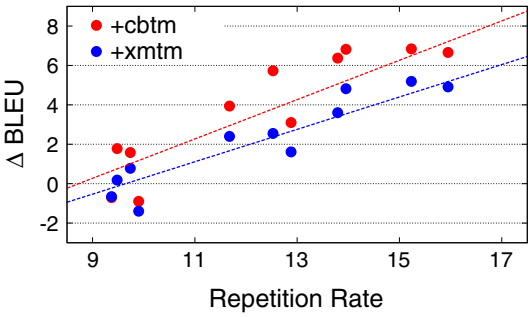

Fig. 10 English–Italian IT domain: trend lines of Repetition Rate vs. ΔBLEU between TM-adapted (`+cbtm` and `+xmtm`) and baseline systems. Corresponding trend lines have R^2 of 0.79 and 0.86, respectively

6.4 Impact of the discriminative re-ranking module

The discriminative re-ranking module described in Sect. 4.5 can be independently combined with any of the previous systems, including the baseline. For our experiments we precomputed 200-best lists for all systems and for all data sets. The features of the hypotheses could then be readily extracted from the translations and the phrase alignment. The **new** phrase pairs from adapted systems are also communicated to the re-ranking module to have complete feature representations for all hypotheses. The development for the re-ranking module was carried out with the baseline system on the set0–2 of the English–Italian IT data and settings were carried over to the other data sets and combination with other systems without any further optimization.

In Fig. 11 we directly compare the performance of the re-ranking module applied alone (`+rnk`) against the `+cbtm+cblm` and `+xmtm+cblm` systems, clearly proving that its effectiveness is lower, but reasonably consistent. This is not surprising, as the re-ranking uses a narrow search space and is not able to affect decoding in any way.

Results for the stacking of the re-ranking on top of all systems in terms of BLEU and TER differences from the baseline for the English–Italian IT data sets are shown as a scatter plot in Fig. 12. Favorable results lie in the 4th quadrant, with an increase in BLEU and a decrease in TER.

The re-ranking module worsens BLEU scores in several cases (18 out of 66), but 17 of them refer to test documents set6A–D. We already discussed this in Sect. 6.3, stressing that they do not fit very well into an online adaptation scenario because of their low repetitiveness.

By excluding these sets, the module increases the BLEU score in the vast majority of cases (41 out of 42); however, TER still worsens in more cases (10 out of 42). This can be attributed to the fixation on n-grams matches enforced by the re-ranking step, which promotes higher order n-grams matches through its feature values. See Fig. 13 for two examples taken from the outputs of the baseline system and the baseline stacked with the re-ranking module demonstrating this issue. In both cases, the baseline system has fewer or lower n-grams matches than the hypothesis of the re-ranker, but it would take fewer operations to transform the baseline translations into the references: exactly one swap in the first example, and three operations in the second example.

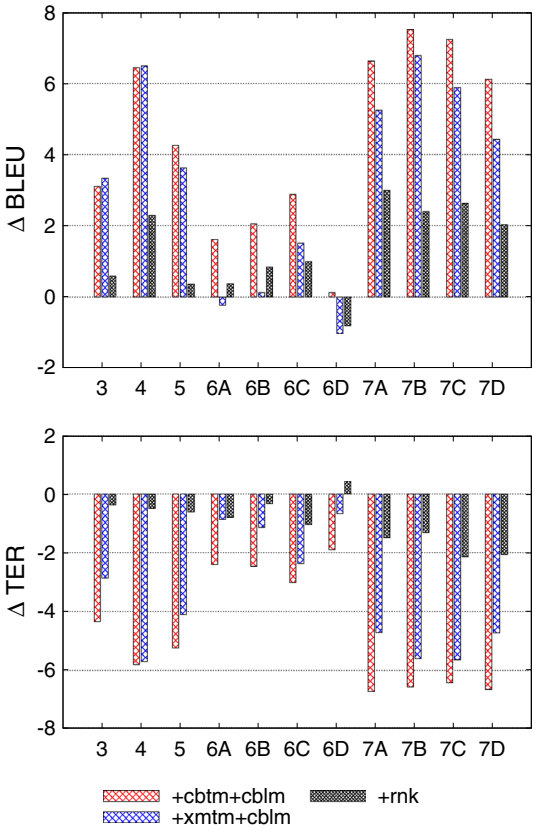

Fig. 11 Performance of the discriminative re-ranking module (`+rnk`) on all English–Italian IT documents, expressed as difference in BLEU (*left*) and TER (*right*) from the baseline system. As reference the performance of the `+cbtm+cblm` and `+xmtm+cblm` systems are also reported

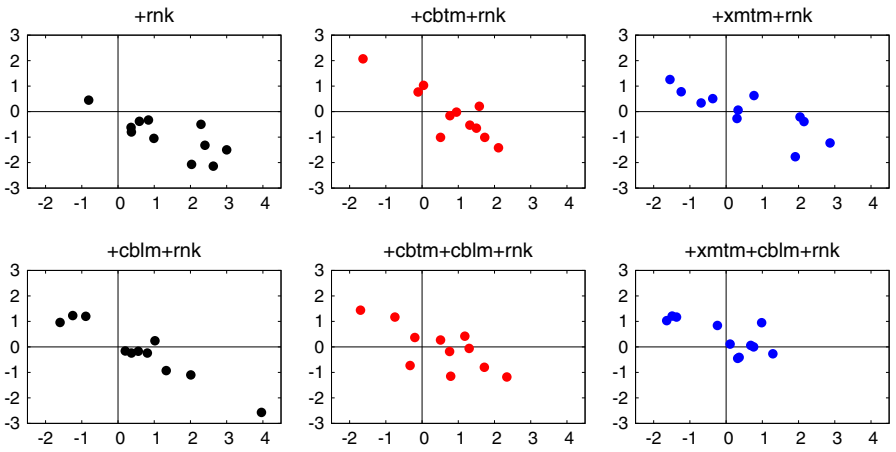

Fig. 12 Test set results in terms of BLEU (x-axis) and TER (y-axis) differences from the baseline system for the re-ranking module stacked with all systems on the English–Italian IT data sets

Example: English-Italian IT, set4	
	#741
source	*Table 7-8-1 Middleware Support Services Baseline*
bsln	***Tabella 7-8-1 Middleware*** *Services Support Baseline*
+rnk	***Tabella 7-8-1 Middleware Support*** *Baseline Servizi*
reference	*Tabella 7-8-1 Middleware Support Services Baseline*
	...
	#782
source	*Restore the backup file upon Customer request*
bsln	*Ripristina* ***il file di*** *salvataggio* ***su richiesta del*** *cliente*
+rnk	*Ripristina il salvataggio* ***su richiesta del Cliente*** *file*
reference	*Ripristinare il file di backup su richiesta del Cliente*

Fig. 13 Two examples showing a mismatch between BLEU and TER. In the first example, the baseline has fewer n-grams matches than the re-ranked hypothesis, but fewer edits are needed to reach the reference. In the second example, the count of the n-grams matches is identical, but the re-ranked translation has a 4-g match, while the baseline system does not. Still, fewer edits are needed to make the baseline match the reference. Highest order n-grams matches with the reference are in *bold font*

Finally, we examined positively and negatively weighted features that explain how re-ranking can help the system recover from errors by re-weighting translations. For example, our re-ranking model captures the contextual difference of translating the English *and* into the Italian *e* before a consonant or *ed* before a vowel by assigning high positive weight to n-grams such as *DLI ed IBM* and *ed IBM* and a high negative weight to n-grams such as *DLI e IBM* and *e IBM*. Due to the frequent use of title case in the IT data, the system also learned to prefer phrase pairs with matching case (*Life* → *Vita*, *machine* → *macchina*) over pairs with case mismatch (*Customer* → *clienti*).

6.5 Performance of the full-fledged system

According to the results reported above, the best dynamic system employs both TM (with internal cache) and LM adaptation techniques and the discriminative re-ranking module. We therefore apply this system combination (`+cbtm+cblm+rnk`) to the other tasks, namely English–Italian and English–Spanish Legal and English–German and German–English Patents. Its performance is reported in Fig. 14.

The full-fledged adapted system outperforms the baseline system across almost all documents of all tasks, except for documents set6A–D of English–Italian IT and set3A–D of the English–Italian Legal. A reasonable explanation for this bad performance was given in Sect. 6.3, and it is related to the relatively low Repetition Rates of those documents, as reported in Table 2. Therefore, we exclude these sets from any further analysis. On the other documents improvements vary a lot and range from $+1$ to $+10$ BLEU points and -1 to -8 TER points.

One goal of the proposed online adaptation approaches is the improvement of the system over time. Therefore, the considered systems are evaluated not only in terms of their average BLEU and TER performance, but also in terms of Percentage Slope, a measure of their learning capability as explained in Sect. 3.3. According to the aforementioned discussion, where we assessed the importance of the gain rather than the absolute value of the metric, in Fig. 14 we also plot the ΔSlope difference of the full-fledged system from the baseline system.

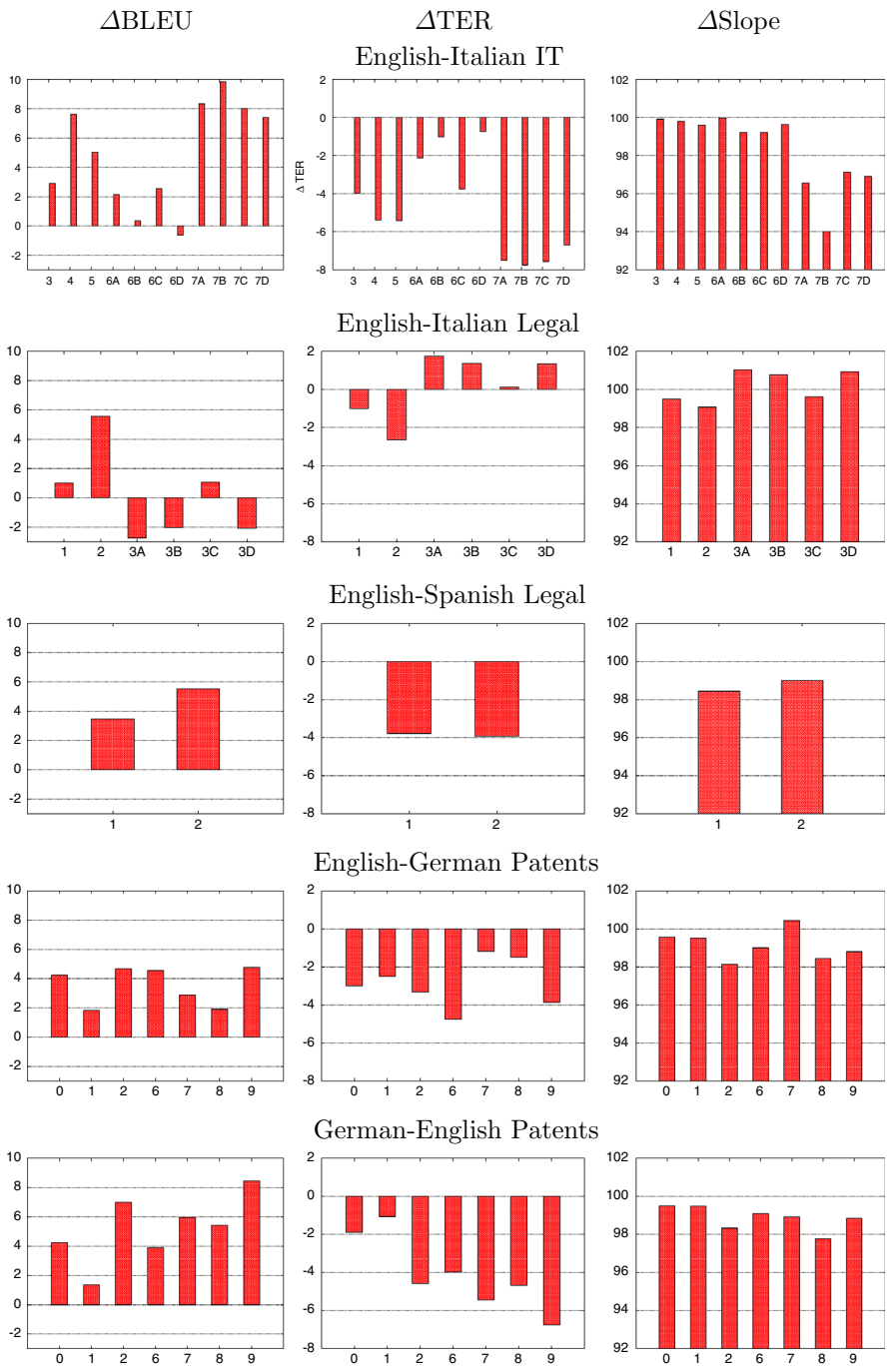

Fig. 14 Performance of the full-fledged system on all documents of all tasks, expressed as difference in BLEU (*left*) and TER (*middle*) from the baseline system, and ΔSlope (*right*)

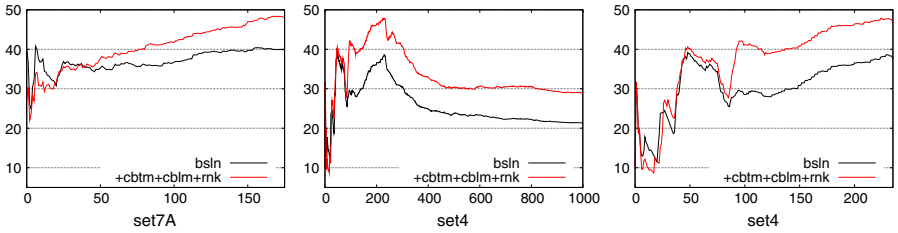

Fig. 15 Performance in terms of BLEU of the baseline and full-fledged systems on increasing portions of set7A (*left*) and set4 (*middle*) of the English–Italian IT task. Plot on the right zooms in on the first 235 sentences of set4

The plots indicate that the learning capability of the full-fledged system is definitely strong only on test documents set7A–D of the English–Italian IT task, where the ΔSlope ranges from 94 to 97 points. For the remaining test sets the gain is much lower or even slightly negative.

A reasonable explanation for this very different behavior can be inferred by considering how the baseline and the full-fledged system perform over time. Figure 15 reports the incremental BLEU achieved by the two systems on two typical sets of the English–Italian IT task, namely set4 and set7A. Apart from the first part of the document (up to segment #50) for which BLEU is not reliable enough due to the small amount of text, performance of the compared system on set7A is definitely regular; the full-fledged system constantly improves over the baseline as the divergent curve proves.

The behavior on set4 is much less consistent; the big drop from sentence 235 is due to a substantial change in the intrinsic difficulty of the source text. Indeed, by analyzing the text we observed that up to that segment the software manual mainly contains table of contents and indexed items, whereas a more descriptive and verbose language is used afterwards. By focusing on the first 235 segments (see Fig. 15 right), the learning capability of the full-fledged system is observable again, and it is confirmed by the ΔSlope decreasing to 96.05 from the 99.81 computed on whole set4.

This analysis suggests a related additional explanation for the low ΔSlope figures. Considering a document as a very tiny sub-domain, most of the document-specific information has been learned at some point; after that the learning curve necessarily flattens, and the assumption of an almost constant learning ratio, which the *Slope* metric relies on, no longer holds. Nevertheless, the reliability of the metric remains valid if computed over the first part. But to confirm this observation, a deep and specialized investigation into how the ΔSlope varies over time should be performed.

In Fig. 16, we give some insights about the translations actually produced by the baseline and dynamic systems, in order to highlight the main pros and cons of the proposed online MT adaptation method. We give three examples from the the English–Italian Information Technology task, but similar cases can be easily found in the other tasks.

Example 1 shows that the adaptive system was able to capture from the user feedback (Sentence #35) the preferred translation of "*consistency*" ("*di congruenza*") and to properly use it when the phrase appears again in sentence #45. In Example 2 user feedback is immediately exploited to correct not only the lexical error ("*Customer*" vs

Example 1: English-Italian IT, set7A

	#35
source	*they do not have to belong to a* ***consistency*** *group .*
bsln	*tuttavia , essi non devono appartenere a un gruppo* ***di coerenza*** *.*
+cbtm+cblm	*tuttavia , essi non devono appartenere a un gruppo* ***di coerenza*** *.*
post-edit	*possono anche non appartenere ad alcun gruppo* ***di congruenza*** *.*
	...
	#45
source	*... can be added to the* ***consistency*** *group .*
bsln	*... può essere aggiunto al gruppo* ***di coerenza*** *.*
+cbtm+cblm	*... può essere aggiunto al gruppo* ***di congruenza*** *.*
post-edit	*... possono essere aggiunte al gruppo* ***di congruenza*** *.*

Example 2: English-Italian IT, set4

	#70
source	*11* ***Customer*** *responsibilities*
bsln	*11* ***Customer*** *responsabilità*
+cbtm+cblm	*11* ***Customer*** *responsabilità*
post-edit	*11 Responsabilità del* ***Cliente***
	...
	#71
source	*11.1* ***Customer*** *responsibilities*
bsln	*11.1* ***Customer*** *responsabilità*
+cbtm+cblm	*11.1 Responsabilità del* ***Cliente***
post-edit	*11.1 Responsabilità del* ***Cliente***

Example 3: English-Italian IT, set4

	#269
source	*1* ***Pre-configuration*** *services*
bsln	*1* ***Pre-configuration*** *servizi*
+cbtm+cblm	*1 I servizi* ***Pre-configuration***
+cbtm+cblm+rnk	*I servizi di 1 .* ***Pre-configuration***
post-edit	*1 Servizi di* ***Pre-configurazione***
	...
	#283
source	***Pre-configuration*** *services in* ***Italy***
bsln	***Pre-configuration*** *servizi in* ***Italy***
+cbtm+cblm	***Pre-configurazione*** *servizi in* ***Italy***
+cbtm+cblm+rnk	*I servizi di* ***Pre-configurazione*** *in* ***Italia***
post-edit	*Servizi di* ***Pre-Configurazione*** *in* ***Italia***

Fig. 16 Some examples of the translations produced by the baseline and dynamic systems showing typical errors and improvements. Source input and post-edits are also reported

"*Cliente*") but also the word order error. In Example 3 the term "*Pre-configuration*", occurring for the first time in sentence #269, is erroneously translated because it has not been seen before in the training data. Thanks to user feedback the correct translation "*Pre-configurazione*" is added to the local translation model, which is able to provide the correct translation at the next occurrence in sentence #283. Nevertheless, the system `+cbtm+cblm` still contains word-reordering ("*Pre-configurazione servizi*") and lexical ("*Italy*") errors; both errors are fixed by the re-ranking module. This also

indicates that system `+cbtm+cblm` can produce the correct translation, but this does not necessarily have the highest score.

Example 3 also reveals a limitation of our current online adaptation approaches: they strongly rely on the quality and coherence of the user feedback. The English terms "*Pre-configuration*" is inconsistently translated with different word casing, causing an actual, though minor, error in sentence #283. In fact, we have to consider that the feedback exploited here is not a true post-edit and was produced independently from the system. We expect that in a real usage scenario the translator will be influenced by the MT suggestion and consequently will produce more coherent translations.

7 Conclusion

We presented the application of an online learning protocol that fits well with the typical post-editing workflow and achieves a tighter integration of human and machine translation. The protocol offers immediate feedback provided by the human after each translation output, and allows an MT system to learn from this feedback for future translations. Assuming coherent texts, the obvious advantages of this scenario are the possibilities to provide more consistent MT suggestions, to reduce the post-editing effort, and last but not least to enhance the user experience. Our adaptation techniques generalise in some sense the behaviour of some translation memory systems, which perform real-time updates: the MT system is adapted as soon as a segment is post-edited, so that future outputs will reflect the recent translation preferences of the user. The generalization lies in the fact that the MT system learns user preferences both at the sentence level (like a translation memory) and at the phrasal level: i.e. from single words to groups of words. The crucial steps in this learning process are the extraction of relevant parallel and target phrases from the source and post-edited segments, and the assignment of proper scores to such phrases. While we use a constrained search procedure for the first step, we investigated two approaches for assigning scores to the extracted parallel phrases and monolingual target phrases: (i) methods that augment the generative components of the MT system, translation and language models, by building local models based on internal or external caches; (ii) a discriminative method based on a structured perceptron that refines a feature-based re-ranking module applied to the k-best translations of the MT system. The proposed generative and discriminative approaches are independent and can be straightforwardly cascaded.

A deep investigation and comparison of the proposed adaptation techniques have been conducted on three domains and four translation directions. Evaluations have been carried out by using both reference and post-edited translations. We also related the Repetition Rate capturing the amount of phrase repetitiveness inside a text to effectiveness of our adaptation methods on the different domains we tested.

The main outcomes of our experiments can be summarized as follows: (i) adaptation is highly affected by the level of repetitiveness of the text; (ii) bilingual features are more effective than monolingual features; (iii) the internal cache model is the most effective adaptation method; (iv) generative and discriminative methods are to some extent additive.

Our work also raised interesting issues related to the development of MT systems that learn from user feedback, which have only be addressed partially. First, our cache-based adaptation method shows performance correlating with the repetitiveness rate of the input. As the input text is available in advance and its repetitiveness may locally vary in a significant way, it would be interesting to refine the Repetition Rate measure in such a way as to predict which portions of texts can mostly benefit from the cache models. Another issue which deserves further investigations is the feature extraction step that is applied on the source and target sides of each post-edited segment. In particular, its effectiveness (precision and recall) could be improved especially when one or more translation pairs including never observed words occur. Finally, work is in progress to actually integrate the discussed adaptation methods in a CAT tool and to field test them with professional translators. Relevant aspects we are focusing on are the latency of the adaptation step, which should not delay the human workflow, and the concurrent use of cache models by multiple users translating similar documents.

Acknowledgments FBK researchers were supported by the MateCat project, funded by the EC under FP7; researchers at Heidelberg University by DFG grant "Cross-language Learning-to-Rank for Patent Retrieval".

References

Bertoldi N (2014) Dynamic models in Moses for online adaptation. Prague Bull Math Linguist 101:7–28

Bertoldi N, Cettolo M, Federico M, Buck C (2012) Evaluating the learning curve of domain adaptive statistical machine translation systems. In: Proceedings of the seventh workshop on statistical machine translation. Montréal, Canada, pp 433–441

Bertoldi N, Cettolo M, Federico M (2013) Cache-based online adaptation for machine translation enhanced computer assisted translation. In: Proceedings of the MT summit XIV, Nice, France, pp 35–42

Bisazza A, Ruiz N, Federico M (2011) Fill-up versus interpolation methods for phrase-based SMT adaptation. In: Proceedings of the international workshop on spoken language translation (IWSLT). San Francisco, California, USA, pp 136–143

Cesa-Bianchi N, Lugosi G (2006) Prediction, learning, and games. Cambridge University Press, Cambridge

Cesa-Bianchi N, Reverberi G, Szedmak S (2008) Online learning algorithms for computer-assisted translation. Technical report, SMART (www.smart-project.eu)

Cettolo M, Federico M, Bertoldi N (2010) Mining parallel fragments from comparable texts. In: Proceedings of the international workshop on spoken language translation (IWSLT), Paris, France, pp 227–234

Cettolo M, Bertoldi N, Federico M (2011) Methods for smoothing the optimizer instability in SMT. In: Proceedings of the MT summit XIII, Xiamen, China, pp 32–39

Chiang D, Marton Y, Resnik P (2008) Online large-margin training of syntactic and structural translation features. In: Proceedings of the conference on empirical methods in natural language processing (EMNLP), Honolulu, Hawaii, USA, pp 224–233

Clark JH, Dyer C, Lavie A, Smith NA (2011) Better hypothesis testing for statistical machine translation: controlling for optimizer instability. In: Proceedings of the 49th annual meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT): short papers, vol 2, Portland, Oregon, USA, pp 176–181

Collins M (2002) 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), Philadelphia, Pennsylvania, USA, pp 1–8

Denkowski M, Dyer C, Lavie A (2014) Learning from post-editing: online model adaptation for statistical machine translation. Proceedings of the 14th conference of the European chapter of the Association for Computational Linguistics (EACL'14), Gothenburg, Sweden, pp 395–404

Federico M, Bertoldi N, Cettolo M (2008) IRSTLM: an open source toolkit for handling large scale language models. In: Proceedings of interspeech, Brisbane, Australia, pp 1618–1621

Federico M, Cattelan A, Trombetti M (2012) Measuring user productivity in machine translation enhanced computer assisted translation. In: Proceedings of the tenth conference of the Association for Machine Translation in the Americas (AMTA), San Diego, California, USA

Foster G, Kuhn R (2007) Mixture-model adaptation for SMT. In: Proceedings of the second workshop on statistical machine translation. Prague, Czech Republic, pp 128–135

Green S, Heer J, Manning C (2013) The efficacy of human post-editing for language translation. In: Proceedings of the SIGCHI conference on human factors in computing systems. Paris, France, pp 439–448

Hardt D, Elming J (2010) Incremental re-training for post-editing SMT. In: Proceedings of the conference of the Association for Machine Translation in the Americas (AMTA), Denver, Colorado, USA, pp 217–237

Koehn P (2010) Statistical machine translation. Cambridge University Press, Cambridge

Koehn P, Schroeder J (2007) Experiments in domain adaptation for statistical machine translation. In: Proceedings of the second workshop on statistical machine translation. Prague, Czech Republic, pp 224–227

Koehn P, Hoang H, Birch A, Callison-Burch C, Federico M, Bertoldi N, Cowan B, Shen W, Moran C, Zens R, Dyer C, Bojar O, Constantin A, Herbst E (2007) Moses: open source toolkit for statistical machine translation. In: Proceedings of the 45th annual meeting of the Association for Computational Linguistics companion volume proceedings of the demo and poster sessions. Prague, Czech Republic, pp 177–180

Kuhn R, De Mori R (1990) A cache-based natural language model for speech recognition. IEEE Trans Pattern Anal Machine Intell 12(6):570–582

Läubli S, Fishel M, Massey G, Ehrensberger-Dow M, Volk M (2013) Assessing post-editing efficiency in a realistic translation environment. In: Proceedings of the MT summit XIV workshop on post-editing technology and practice, Nice, France, pp 83–91

Levenberg A, Callison-Burch C, Osborne M (2010) Stream-based translation models for statistical machine translation. In: Proceedings of the 2010 annual conference of the North American chapter of the Association for Computational Linguistics (HLT-NAACL), Los Angeles, California, USA, pp 394–402

Levenberg A, Dyer C, Blunsom P (2012) A Bayesian model for learning SCFGs with discontiguous rules. In: Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL), Jeju Island, Korea, pp 223–232

Liang P, Bouchard-Côté A, Klein D, Taskar B (2006) An end-to-end discriminative approach to machine translation. In: Proceedings of the 21st international conference on computational linguistics and 44th annual meeting of the Association for Computational Linguistics, Sydney, Australia, pp 761–768

Liu L, Cao H, Watanabe T, Zhao T, Yu M, Zhu C (2012) Locally training the log-linear model for SMT. In: Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL), Jeju Island, Korea, pp 402–411

López-Salcedo FJ, Sanchis-Trilles G, Casacuberta F (2012) Online learning of log-linear weights in interactive machine translation. In: Proceedings of Iber speech, Madrid, Spain, pp 277–286

Martínez-Gómez P, Sanchis-Trilles G, Casacuberta F (2012) Online adaptation strategies for statistical machine translation in post-editing scenarios. Pattern Recognit 45(9):3193–3202

Nepveu L, Lapalme G, Langlais P, Foster G (2004) Adaptive language and translation models for interactive machine translation. In: Proceedings of the conference on empirical methods in natural language processing (EMNLP), Barcelona, Spain, pp 190–197

Noreen EW (1989) Computer intensive methods for testing hypotheses: an introduction. Wiley Interscience, New York

Och FJ (2003) Minimum error rate training in statistical machine translation. In: Proceedings of the 41st annual meeting of the Association for Computational Linguistics. Sapporo, Japan, pp 160–167

Ortiz-Martínez D, García-Varea I, Casacuberta F (2010) Online learning for interactive statistical machine translation. In: Proceedings of the 2010 annual conference of the North American chapter of the Association of Computational Linguistics (HLT-NAACL), Los Angeles, pp 546–554

Papineni K, Roukos S, Ward T, Zhu WJ (2002) BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the Association of Computational Linguistics (ACL). Philadelphia, Pennsylvania, USA, pp 311–318

Snover M, Dorr B, Schwartz R, Micciulla L, Makhoul J (2006) A study of translation edit rate with targeted human annotation. In: Proceedings of the 5th conference of the Association for Machine Translation in the Americas (AMTA). Cambridge, Massachusetts, USA, pp 223–231

Steinberger R, Pouliquen B, Widiger A, Ignat C, Erjavec T, Tufiş D, Varga D (2006) The JRC-Acquis: a multilingual aligned parallel corpus with 20+ languages. In: Proceedings of the 5th international conference on language resources and evaluation (LREC). Genoa, Italy, pp 2142–2147

Tiedemann J (2010) Context adaptation in statistical machine translation using models with exponentially decaying cache. In: Proceedings of the 2010 ACL workshop on domain adaptation for natural language processing, Uppsala, Sweden, pp 8–15

Tiedemann J (2012) Parallel data, tools and interfaces in OPUS. Proceedings of the 8th international conference on language resources and evaluation (LREC), Istanbul, Turkey, pp 2214–2218

Wäschle K, Riezler S (2012) Analyzing parallelism and domain similarities in the MAREC patent corpus. Proceedings of the 5th information retrieval facility conference (IRFC), Vienna, Austria, pp 12–27.

Wäschle K, Simianer P, Bertoldi N, Riezler S, Federico M (2013) Generative and discriminative methods for online adaptation in SMT. In: Proceedings of the MT summit XIV, Nice, France, pp 11–18