# Cache-based Online Adaptation for
# Machine Translation Enhanced Computer Assisted Translation

**Nicola Bertoldi** **Mauro Cettolo** **Marcello Federico**

FBK - Fondazione Bruno Kessler
via Sommarive 18
38123 Povo, Trento, Italy
{bertoldi,cettolo,federico}@fbk.eu

## Abstract

The integration of machine translation in the human translation work flow rises intriguing and challenging research issues. One of them, addressed in this work, is how to dynamically adapt phrase-based statistical MT from user post-editing. By casting the problem in the online machine learning paradigm, we propose a cache-based adaptation technique method that dynamically stores target n-gram and phrase-pair features used by the translator. For the sake of adaptation, during decoding not only recency of the features stored in the cache is rewarded but also their occurrence in similar already translated sentences in the document. Our experimental results show the effectiveness of the devised method both on standard benchmarks and on documents post-edited by professional translators through the real use of the MateCat tool.

## 1 Introduction

Worldwide demand of translation services has dramatically accelerated in the last decade, as an effect of the 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. Recent achievements by the so-called statistical MT (SMT) approach have raised new expectations in the translation industry. Several empirical studies have recently given evidence of significant productivity gains when human translators post-edit machine translation output rather than translate from scratch. So far, however, SMT has focused on providing ready-to-use translations, rather than outputs that minimize the effort of a human translator. Ignoring this need is not motivated.

In fact, nowadays a very hot issue for research and industry is how to effectively integrate machine translation within computer assisted translation software. This is exactly the goal of the MateCat[1] and CasmaCat[2] European projects, which are jointly developing a new generation CAT tool integrating novel interaction modalities and MT functions.

In this paper we deal with SMT models which dynamically learn from the user feedback by means of a caching mechanism. The main idea behind cache-based models is to mix a large global (static) model with a small local (dynamic) model estimated from recent items observed in the history of the input stream. In (Kuhn and De Mori, 1990), the caching mechanism was applied to language models, and only very later its use was extended to translation models (Nepveu et al., 2004). In lab tests, cache-based language and translation models have already proven to be effective in interactive SMT (Nepveu et al., 2004); in adapting generic SMT models, (Tiedemann, 2010) obtained interesting but not definitive gains, the "main obstacle" being "the invalid assumption that initial translations are correct". In a real CAT framework like ours, the cached items are correct by definition.

The road map of our work is defined through a list of research questions addressed in this paper. Clearly, the first questions we are interested in is:
**Q1** - Is the effectiveness of cache-based adaptation confirmed in a real CAT environment?

As mentioned in (Tiedemann, 2010), there are two types of important properties in natural language and translation that are often ignored in statistical models: repetition and consistency. Repetition of content words is very common, espe-

[1] www.matecat.com
[2] www.casmacat.eu

Sima'an, K., Forcada, M.L., Grasmick, D., Depraetere, H., Way, A. (eds.)
*Proceedings of the XIV Machine Translation Summit (Nice, September 2–6, 2013), p. 35–42.*

cially in "technical" documents. Consistency in translation has to do with ambiguity, which is typically handled by working in specific domains and contexts such that ambiguous items have a well-defined and consistent meaning. Assuming that cache based-models are effective in handling these two types of phenomena (Q1), we ask:

**Q2** - How do cache-based adaptive models perform when the translated documents are less repetitive and translation inconsistencies occur?

In fact, the caching approach could fail due to the risks of adding noise and corrupting local dependencies. The last issue arises another question:

**Q3** - Given a document, is it possible to predict if cache-based adaptation will properly work or not? Moreover, assuming that the effectiveness of cache models depends on some linguistic features of the document itself (Q2), which are those features?

Cache models take into account recency by means of a decaying factor (Clarkson and Robinson, 1997), which has been shown to be effective on average, since repetitions frequently occur in contiguous portions of the text. Anyway, it might also happen that items cached in far epochs which are penalized by the decay indeed occur in the current sentence to translate. Then, we ask:

**Q4** - Is it worth to re-emphasize translations cached in far epochs that are predicted to be useful for the current translation context?

In this paper, we try to answer to all these questions by running experiments on proprietary and public evaluation sets, including texts post-edited by professional translators through a CAT tool.

The paper is organized as follows. After a brief overview of related works in Section 2, the adaptation protocol is sketched in Section 3 which provides details on its implementation as well. Section 4 reports on experiments, including an analysis of data supporting our proposals and a discussion of results. Some final summarizing comments end the paper.

## 2 Related Research

The concept of cache arose in computer science in the '60s, when it was introduced to speed-up the fetch of instructions and data, and the virtual-to-physical address translation. Generally speaking, it is a component that transparently stores items so that future requests for them can be served faster. Caches exploit the locality of reference, also known as principle of locality: the same value, or related storage location, is frequently accessed. This phenomenon does not occur only 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 statical language models by Kuhn and De Mori (1990). In this case, the argument was not the efficiency like for computers but the improvement of prediction capability of the model; caching has also been used for time saving in the concrete implementation of language models (Federico et al., 2008).

The use of cache in SMT 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 a further processing for the automatic alignment of source and post-edition words, namely the IBM model 2 Viterbi search. Tiedermann (2010) proposed to incrementally populate the translation model cache with the translation options used by the decoder to generate the final best translation; no additional alignment step is required here. Our cache-based translation model stands in between these two works: cache is filled with phrase pairs from the previous post-edition; 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. Moreover, we propose a further enhancement of the basic caching mechanism for rewarding cached items related to the current sentence to translate.

Our work also deals with MT adaptation in general, and online learning more specifically. A thorough overview of the literature on these topics can be found in the companion paper (Wäschle et al., 2013). Here, we just list some of the most representative papers: (Levenberg et al., 2010; Levenberg et al., 2011) for online learning methods in streaming scenarios; (Koehn and Schroeder, 2007; Foster and Kuhn, 2007; Bisazza et al., 2011; Liu et al., 2012; Bertoldi et al., 2012) for incremental adaptation.

## 3 Online Adaptation in SMT

In the well established CAT work flow, source documents are split into chunks, typically corresponding to sentences and called *segments*, that are in general translated sequentially. When the translator opens a segment, the CAT tool tries to propose 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 indeed a valuable source of knowledge, which is in fact readily stored in the translation memory for future use. In fact, given the compositional nature of SMT, it comes natural to believe that also text fragments observed in the post-edited or translated segments can be potentially useful to improve SMT output of future segments.

From a machine learning perspective, the CAT scenario perfectly fits the *online learning* paradigm, 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 CAT scenario as follows. We assume an initial SMT engine $M_1$ and a source document $x_1, \ldots, x_D$. For $t = 1, \ldots, D$ the following steps are performed:

**1**. the source segment $x_t$ is received

**2**. ...`[optional step]`...

**3**. a translation $\hat{y}_t$ is computed with $M_t$

**4**. a post-edited translation $y_t$ is received

**5**. a new system $M_{t+1}$ is created by adapting $M_t$ with features extracted from $(x_t, y_t)$

Step 5 is implemented via a caching mechanism that allows us to define and dynamically adapt local (with respect to the document) models that are combined during decoding with the global SMT models estimated on the training data. In the following, we present how the local cache-based models are defined and adapted under the well-known phrase-based SMT setting of the Moses toolkit (Koehn et al., 2007). They will be included in a future release of the toolkit.

The optional step 2 is additional with respect to the basic online learning procedure and comprises the updating of $M_t$ with context features, as described in more detail in Section 3.3.

### 3.1 TM Adaptation

The pair $(x, y)$ composed of a source segment and its post-edited translation, is exploited to update a *local* translation model. This model is intended for integrating new translation alternatives suggested by the user and for rewarding those approved, with the ultimate goal of translating the successive segments more consistently with the user preferences.

The local translation model is implemented as an additional phrase table 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 actually inserted, and scored accordingly. Each new insertion causes the ageing of the existing phrase pairs and hence their rescoring; in case of re-insertion of a phrase pair, the old value is overwritten. Phrase pairs are scored based on a negative exponential decaying function.

For each segment pair, a set of phrase pairs are extracted from the partial alignment provided by the constrained search algorithm described by Cettolo et al. (2010). The procedure, detailed in (Wäschle et al., 2013), extracts both already "known" and "new" pairs; the latter can provide translation options for OOVs and phrases including OOVs. All the extracted phrase pairs are simultaneously added to the local translation model by feeding the decoder with an XML-tag like:

> `<dlt cbtm=`“*The crude face of domination .|||Le visage rustre de la domination .||||crude|||rustre||||···||||domination |||la domination||||face|||visage*”`/>`

The pair $(x, y)$ consisting of the whole segments is also added to mimic the behaviour of a translation memory.

During decoding, translation alternatives are searched both in the global static and in the local dynamic phrase tables, and scored accordingly.

### 3.2 LM Adaptation

Similarly to the local translation model, a *local* language model is built to reward the $n$-grams found in post-edited translation. This model is implemented as an additional feature of the log-linear model, which provides a score for each translation option, based on a cache storing target $n$-grams.

For each user-approved translation $y$, all its $n$-grams containing at least one content word are extracted, associated with an age and scored by a negative exponential decaying function of the age. The same policy for modifying the local translation model is applied to the local language model; only the XML-tag input slightly changes:

> `<dlt cblm=`”*Le visage||visage rustre|| rustre de||la domination||domination .||visage||rustre||domination*”`/>`

At decoding time, the target side of each translation option fetched by the search algorithm is scored with the cache model. If it is not found, it does not receive any reward. Notice that $n$-grams crossing over contiguous translation options are not taken into account by this model.

It is worth emphasizing that, although the chosen name, the proposed additional feature is not a conventional language model, but rather a function

| task | lang. pair | source of data | segm | src tok | trg tok |
|---|---|---|---|---|---|
| IT | en→it | commercial | 1.9 | 27.8 | 29.0 |
| LGL | en→it | JRC | 1.5 | 47.6 | 49.3 |
| TED | ar→en | WIT$^3$ | 0.138 | 2.5 | 2.7 |
|  | en→fr |  | 0.141 | 2.8 | 2.9 |

Table 1: Language pairs, origin of texts and statistics of parallel data of the IT, Legal and TED domains used for training purposes. Counts of source and target words refer to tokenized texts after duplicates removal. Numbers are in millions.

rewarding approved high-quality word sequences.

### 3.3 Context Reward

As mentioned before, the score of the cached items decays with their age thus rewarding recency of observed feature. To mimic the way translation memory works, we also decide to reward target $n$-gram and phrase-pair features observed in similar and already translated segments of the document. This implies adding the following *lazy learning* step to the previous on-line learning algorithm:

2. $M_t$ is updated with features extracted from the pair $(x, y)$ in $\{(x_1, y_1), \dots, (x_{t-1}, y_{t-1})\}$ such that $x$ is most similar to $x_t$.

Practically, we equip our SMT engine with a translation memory that stores each translation pair $(x_t, y_t)$ together with all features extracted from it. At step 2, the SMT engine queries such translation memory to retrieve the most similar segment pair, and resets inside the LM and TM cache models the age of the features associated to the retrieved pair. In our experiments, the translation memory is realized by means of an information retrieval engine implemented with the open-source software `Lucene` (McCandless et al., 2010).

## 4 Experimental Evaluation

### 4.1 Overview on Data

Evaluations were performed on both proprietary and publicly available data, involving the translation of documents from three domains, namely Information Technology (IT), Legal (LGL) and TED talks,[3] and for different language pairs. Table 1 summarizes the experimental framework in terms of data type.

Commercial data were provided by the industrial partner of the MateCat project and were collected during the real use of CAT tools by professional translators. Public data, which allow to replicate and cross-assess our outcomes, were chosen to cover a wide range of linguistic conditions. For the Legal domain, we worked with the JRC-Acquis collection (Steinberger et al., 2006), on the English to Italian direction. The same direction is also investigated for the IT domain. As concerns TED, the two official MT tasks of the IWSLT 2012 evaluation campaign (Federico et al., 2012) were considered, i.e. the translation of talks from Arabic to English and from English to French.

#### 4.1.1 Training Data

The training data for the IT domain mostly consist of data extracted from a translation memory built during the execution of translation projects commissioned by several commercial companies. In addition, parallel texts from the OPUS corpus (Tiedemann, 2012) were are also included.

Training data of the Legal domain come from Version 3.0 of the JRC-Acquis collection (Steinberger et al., 2006), while TED training data are those released through the WIT$^3$ (Cettolo et al., 2012) website for the IWSLT 2012 evaluation campaign (Federico et al., 2012). The reader may again refer to Table 1 for statistics on the actual corpora employed for training.

#### 4.1.2 Evaluation Data

Evaluation on IT and Legal domains has been performed on data collected in a two-day field test. During the first day (D0), MT suggestions came from baseline SMT systems (Section 4.4); during the second day (D1), they came from static SMT systems adapted to the post-edits of the first day as described in (Cettolo et al., 2013).

For the IT domain, the text to be translated was taken from a software user manual. Concerning the Legal domain, the text was taken from a recent motion for a European Parliament resolution published on the EUR-Lex platform.

Statistics on the test documents translated during the field test are reported in Table 2 (rows D0 and D1); they refer to tokenized texts. Figures on the source side (English) refer to the texts the users are requested to translate; figures on the target side (Italian) refer to the references, whatever they are actual post-edited texts or new translations.

The additional much larger test set of the Legal domain consists of eight documents from the JRC-Acquis corpus, that were selected according to their Eurovoc[4] subject domain labels as follows: We chose eight classes including a not too large nor too small number of documents (around 100) in the corpus. Within each of the eight subsets, one

[3] www.ted.com

[4] eurovoc.europa.eu

| task | set | segm | src tok | trg tok |
|---|---|---|---|---|
| IT | D0 | 177 | 3,332 | 3,544 |
| | D1 | 176 | 3,066 | 3,336 |
| LGL | D0 | 91 | 2,960 | 3,202 |
| | D1 | 90 | 3,007 | 3,421 |
| | test | 861 | 25.4k | 26.3k |
| TED$_{\mathbf{ar}\rightarrow\mathbf{en}}$ | test | 1,664 | 29.3k | 32.0k |
| TED$_{\mathbf{en}\rightarrow\mathbf{fr}}$ | test | 1,664 | 32.0k | 33.8k |

Table 2: Overall statistics on the test sets of our experiments. D0 and D1 references are actual post-edits, because produced during real field tests; *test* references are instead translations from scratch, not produced through a post-editing session.

document was picked having a size adequate for evaluation purposes (few thousand words in order to reach the typical overall size of 20-30 thousand words).[5] For fairness, we removed from the training data all documents of the chosen eight classes.

Each of the eight chosen documents was split into two blocks. The union of first blocks is used for development, that of second blocks for evaluation. The entry LGL-`test` in Table 2 provides some overall statistics on the latter set.

With regard to the TED task, we run our experiments with the development and evaluation sets (`dev2010` and `tst2010`) of the IWSLT 2010 edition. Statistics of `tst2010` sets for the two language pairs are included in Table 2 as well. It is worth mentioning that `dev2010` and `tst2010` consist of 8 and 11 talks, respectively.

### 4.2 Measuring the Repetitiveness of a Text

In Section 1, repetitiveness was mentioned as one of the phenomena occurring in texts that can highly affect the quality of automatic translation.

One way to measure repetitiveness inside a text is to look at the rate of non-singleton $n$-grams it contains. Hence, we computed the rate of non-singleton $n$-grams ($n$=1. . .4) in each of our test sets and, inspired by the way BLEU score is computed, took their geometric mean. In order to make the rates comparable across different sized corpora, statistics over test sets were collected on a sliding window of one thousand words, and properly averaged. Formally, the repetition rate in a document can be expressed as:

$$RR = \left( \prod_{n=1}^{4} \frac{\sum_S n_{1+} - n_1}{\sum_S n_{1+}} \right)^{1/4} \qquad (1)$$

| task | set | % non-singletons | | | | RR |
|---|---|---|---|---|---|---|
| | | 1-gr | 2-gr | 3-gr | 4-gr | |
| IT | D0 | 38.76 | 15.09 | 7.25 | 4.71 | 11.89 |
| | D1 | 42.92 | 18.05 | 9.59 | 6.60 | 14.88 |
| LGL | D0 | 31.56 | 11.63 | 5.66 | 2.99 | 8.88 |
| | D1 | 29.47 | 10.96 | 4.92 | 2.29 | 7.77 |
| | test | 36.64 | 17.14 | 8.87 | 5.68 | 13.34 |
| TED | test | 36.36 | 12.20 | 3.2 | 1.27 | 6.67 |
| News | NIST | 22.14 | 5.95 | 1.15 | 0.24 | 2.45 |
| | WMT | 18.13 | 3.94 | 0.40 | 0.08 | 1.24 |

Table 3: Rate of non-singleton $n$-grams ($n$=1. . .4) and repetition rate (RR) of various sets in different domains. The English side is considered.

where `S` is the sliding subsample, $n_r$ represents the number of different $n$-grams occurring exactly `r` times in `S`, $n_{1+}$ the total number of different $n$-grams in `S`.

### 4.3 Analysis of Data

Table 3 collects the non-singleton rate of $n$-grams and their geometric mean, i.e. the repetition rate, for different sets and domains. The English side is always considered to avoid any language bias. For the sake of comparison, the table also includes average figures of news texts that were computed on NIST MT-08 and MT-09 evaluation data sets (newswire) and on the development sets of the WMT 2011 MT shared task.

The rate of repetition in IT documents is larger than in any other considered domain. The repetition rate is significantly lower in the legal domain, and even lower in the TED talks. At the end of the scale are the news texts, which present a 2-3 lower rate than TED talks. It is worth noticing that significant differences among domains regard longer $n$-grams, aspect well captured by the geometric mean. Finally, in experiment not reported here, similar relative differences were observed also when the only content words were considered. Given these empirical outcomes, it is expected that any possible gain due to cache-based models should be larger on IT documents and gradually decrease on Legal and TED texts.

In order to forecast the effectiveness of the context-reward variant discussed in Section 3.3, we also analyzed the retrieved segments, in two respects: On one side, we looked at the retrieval scores[6] of the top-ranked segments; this suggests how similar the retrieved segment is to the current sentence to translate. On the other side, we measured the "age difference" between the current sen-

[5] The selected Eurovoc codes, as reported in the original documents, are: 1338, 1937, 2560, 3466, 4040, 4692, 5237, 5343. The corresponding selected documents are: 32000R2313, 21998A0421_01,32000D0428,22006A0718_01,32003R1210, 52006AE0582, 52006PC0188_02, 32005R2076.

[6] The retrieval score is computed through the Lucene's Practical Scoring Function as defined at lucene.apache.org.

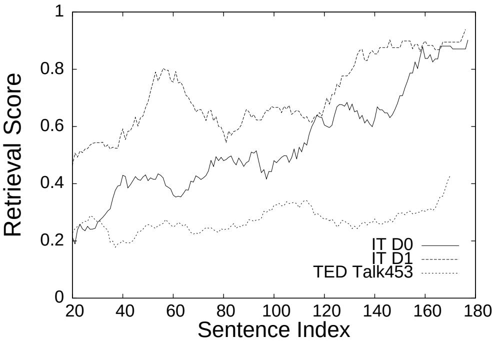

Figure 1: Moving average Lucene scores computed for IT documents and for one TED talk.

tence to translate and the retrieved segment, that is how many sentences occur in between the two sentences; this suggests the added value of the age-resetting with respect to the standard decay. We focused our analysis to IT documents and to the eight TED talks included in `ar-en dev2010` set.

First of all, we noted an important difference between the retrieval score of the top-ranked segments for IT-D0 and -D1 and for TED talks. When the most similar segment is searched among the previously edited sentence pairs of D0, the average score is 0.55; 0.71 in the case of D1.[7] For TED talks, the score is definitely lower ranging in 0.15-0.37, 0.29 on average. Figure 1 plots the moving average retrieval score (20-points window) for D0, D1 and one TED talk. Clearly, higher values (i.e. more similar segments) are found as their number increases, at a greater extent for IT documents than for TED talks. Hence, all our empirical observations suggest that hints from the most similar post-edited sentence in IT domain should be more useful than in the TED domain.

Concerning the age difference, first let us note that in a text with $N$ sentences, assuming no locality effect (Section 2), the expected number of sentences in between any sentence to translate and the most similar among previous sentences is about $N/4$: in fact, for any sentence at position $i$, the mean among the previous segment positions is $i/2$. By taking the average over the whole document:

$$\frac{1}{N}\sum_{i=1}^{N}\frac{i}{2} = \frac{1}{2N} \times \frac{N(N+1)}{2} \simeq \frac{N}{4}$$

The observed values seem consistent with those expected without locality effect, both in IT documents and in TED talks. For instance, in D0 the observed difference is 38 on average (44 expected), for the shortest TED talk it is 8 (9 expected), 38 (47 expected) for the longest.

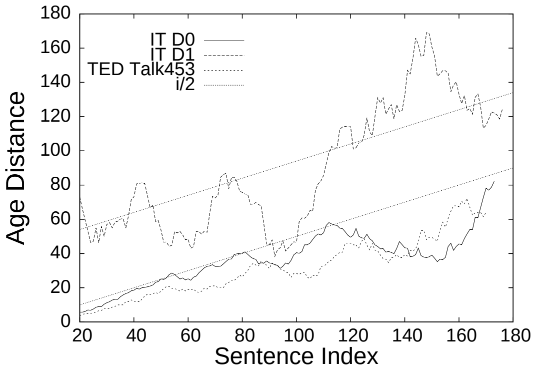

Figure 2: Moving average age-distance of the most similar sentence for IT docs and for one TED talk. The expected age difference curves of these documents are plotted as a reference.

Figure 2 plots the moving average age-distance (20-points window) for D0, D1 and one mid-length TED talk. The two straight lines are the expected age difference for IT-D0 and TED (the lower line) and for IT-D1 (the upper line).

Although not really linearly, the age of the most similar sentence tends to grow with the index of the sentence to translate, whatever the repetition rate: this means that the locality effect, if any, is at least smoothed. The outcome empirically support our idea of going beyond the basic cache mechanism with the exponential decaying age function by searching for and resetting the age of cached items potentially useful for the current context.

### 4.4 SMT Systems

The SMT systems were developed with the Moses toolkit (Koehn et al., 2007). Translation and lexicalized reordering models were trained on the parallel training data; 6-gram (for IT and Legal systems) and 5-gram (for TED systems) LMs with improved Kneser-Ney smoothing (Chen and Goodman, 1999) were estimated on the target side of the training parallel data with the IRSTLM toolkit (Federico et al., 2008). The weights of the log-linear interpolation model were optimized via the MERT procedure provided with Moses. Performance are provided in terms of BLEU and TER, computed by means of the `MultEval` script (Clark et al., 2011) that also provides the standard deviation $\sigma$, and of GTM[8]

Here some specific features of systems developed for each task:

`IT/Legal baselines`: different weights were used in the two days of the field test,

[7] The most similar segments for D1 are retrieved in D0 segments as well.

[8] nlp.cs.nyu.edu/GTM

| task | system | D0 | | | D1 | | |
|---|---|---|---|---|---|---|---|
| | | BLEU ($\sigma$) | TER ($\sigma$) | GTM | BLEU ($\sigma$) | TER ($\sigma$) | GTM |
| IT | baseline | 44.69 (1.67) | 36.34 (1.30) | 74.59 | 41.06 (1.57) | 39.44 (1.31) | 73.17 |
| | +cache | 47.25 (1.81) | 34.83 (1.34) | 76.36 | 44.61 (1.72) | 36.01 (1.38) | 76.21 |
| | +context | 47.89 (1.82) | 34.61 (1.44) | 76.99 | 48.04 (1.93) | 32.33 (1.44) | 79.31 |
| LGL | baseline | 47.66 (2.13) | 34.69 (1.71) | 73.88 | 47.42 (2.11) | 34.62 (1.78) | 74.84 |
| | +cache | 47.69 (2.05) | 33.94 (1.63) | 75.04 | 47.57 (2.09) | 34.43 (1.75) | 74.94 |
| | +context | 46.58 (2.05) | 34.74 (1.65) | 74.49 | 48.07 (2.16) | 33.98 (1.78) | 75.28 |

Table 4: Performance on field test evaluation sets with and w/o online adaptation procedures applied to fair setups of IT and Legal systems.

estimated respectively on sets selected from the training data for D0, and on D0 documents for D1.

`TED baselines`: their training is in all respects identical to the baselines mentioned in (Federico et al., 2012), and in fact their performance are pretty similar.

### 4.5 Results and Discussion

Automatic translation of the test sets mentioned in Section 4.1.2 were performed with the above described SMT. Results are collected in Tables 4 and 5.

By comparing the translation scores with the repetition rates in Table 3, we can summarize the main experimental outcomes and answer the four research questions listed in Section 1:

The `+cache` vs. `baseline` results on the IT and Legal field test suggest that the cache-based adaptation is useful to improve the quality of the automatic MT suggestions, although the gain are not always significant. However, in order to see whether the professional translators would really benefit from this improvement real field tests with the online adaptive system need to be run; in fact we think that even small changes, having negligible impact on the automatic measures, could positively affect their productivity. (Answer to **Q1**).

Although the repetition rate in the TED talks is higher than that of generic news texts, it is not high enough to observe a performance improvement for this task by the online-adapted SMT system. Anyway, it is worth noticing that in such a case, caching and context-reward methods do not negatively impact on SMT performance. (Answer to **Q2**).

In general, the analysis of the `+cache` vs. `baseline` results show a correspondence between the effectiveness of the cache-based adaptation and the repetition rate of the text to translate. Whatever the nature of the user-approved translations is (post-edits or new translations), the repetition rate seems a good and simple measure to predict whether our proposed adaptation scheme will help or not. (Answer to **Q3**). Anyway, we are going to further investigate this issue by comparing variants/alternatives to our repetition rate on correlation measurements.

| task | system | test | | |
|---|---|---|---|---|
| | | BLEU ($\sigma$) | TER ($\sigma$) | GTM |
| LGL | baseline | 40.38 (0.97) | 44.26 (0.91) | 67.95 |
| | +cache | 41.11 (0.98) | 43.71 (0.92) | 68.67 |
| | +context | 41.42 (1.00) | 43.53 (0.94) | 69.25 |
| TED ar−en | baseline | 23.62 (0.45) | 57.22 (0.51) | 57.49 |
| | +cache | 23.83 (0.45) | 57.03 (0.50) | 57.86 |
| | +context | 23.95 (0.46) | 57.08 (0.50) | 57.92 |
| TED en−fr | baseline | 28.71 (0.47) | 51.30 (0.47) | 60.35 |
| | +cache | 28.85 (0.47) | 51.17 (0.47) | 60.50 |
| | +context | 28.79 (0.47) | 51.20 (0.46) | 60.49 |

Table 5: Performance with and without online adaptation procedures applied to fair setups for IT, Legal, and TED tasks.

The context-reward strategy of Section 3.3 yields a further effective exploitation of cached-items, as clearly showed by the `+context` vs. `+cache` results on IT and Legal documents. Also in this case, the improvements on IT-D1 are particularly remarkable. (Answer to **Q4**).

## 5 Conclusions

In this paper we have faced a hot research issue concerning the integration of human and machine translation: how to quickly adapt a statistical MT system on user feedback. We have (i) described a caching mechanism to implement online learning in phrase-based SMT, (ii) introduced a novel lazy-learning method for refreshing cached items expected to be useful for the current translation, and (iii) proposed a repetition rate measure for predicting the utility of cache models in any given text.

As experimental set-up, both conventional MT references and post-edited outputs from static SMT engines have been employed. A summary of the main outcomes is:

- cache-based adaptation is very effective with repetitive texts, but does not hurt with less repetitive texts
- MT accuracy gains are somehow correlated with the document's level of repetitiveness
- the context-reward strategy increases the effectiveness of cache models.

Current work is devoted to fully integrate the proposed methods in the MateCat tool and to run field tests with professional translators in order to measure the actual impact on productivity of online adaptation.

## Acknowledgments

This work was supported by the MateCat project, which is funded by the EC under the $7^{th}$ Framework Programme.

## References

Bertoldi, N., M. Cettolo, M. Federico, and C. Buck. 2012. Evaluating the learning curve of domain adaptive statistical machine translation systems. In *WMT*, Montréal, Canada.

Bisazza, A., N. Ruiz, and M. Federico. 2011. Fill-up versus interpolation methods for phrase-based SMT adaptation. In *IWSLT*, San Francisco, US-CA.

Cettolo, M., M. Federico, and N. Bertoldi. 2010. Mining parallel fragments from comparable texts. In *IWSLT*, Paris, France.

Cettolo, M., C. Girardi, and M. Federico. 2012. WIT$^3$: Web inventory of transcribed and translated talks. In *EAMT*, Trento, Italy.

Cettolo, Mauro, Nicola Bertoldi, and Marcello Federico. 2013. Project adaptation for mt-enhanced computer assisted translation. In *Proceedings of the MT Summit XIV*, Nice, France, September.

Chen, S. F. and J. Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 4(13):359–393.

Clark, J., C. Dyer, A. Lavie, and N. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *ACL*, Portland, US-OR.

Clarkson, P. and A. Robinson. 1997. Language model adaptation using mixtures and an exponentially decaying cache. In *ICASSP*, Munich, Germany.

Federico, M., N. Bertoldi, and M. Cettolo. 2008. IRSTLM: an open source toolkit for handling large scale language models. In *Interspeech*, Melbourne, Australia.

Federico, M., M. Cettolo, L. Bentivogli, M. Paul, and S. Stüker. 2012. Overview of the IWSLT 2012 evaluation campaign. In *IWSLT*, Hong Kong, China.

Foster, G. and R. Kuhn. 2007. Mixture-model adaptation for SMT. In *WMT*, Prague, Czech Republic.

Koehn, P. and J. Schroeder. 2007. Experiments in domain adaptation for statistical machine translation. In *WMT*, Prague, Czech Republic.

Koehn, P., H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *ACL Companion Volume Proc. of the Demo and Poster Sessions*, Prague, Czech Republic.

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

Levenberg, A., C. Callison-Burch, and M. Osborne. 2010. Stream-based translation models for statistical machine translation. In *HLT-NAACL*, Los Angeles, US-CA.

Levenberg, A., M. Osborne, and D. Matthews. 2011. Multiple-stream language models for statistical machine translation. In *WMT*, Edinburgh, UK.

Liu, L., H. Cao, T. Watanabe, T. Zhao, M. Yu, and C. Zhu. 2012. Locally training the log-linear model for SMT. In *EMNLP*, Jeju, Korea.

McCandless, M., E. Hatcher, and O. Gospodnetić. 2010. *Lucene in action*. Manning Publications Co.

Nepveu, L., G. Lapalme, P. Langlais, and G. Foster. 2004. Adaptive language and translation models for interactive machine translation. In *EMNLP*, Barcelona, Spain.

Steinberger, R., B. Pouliquen, A. Widiger, C. Ignat, T. Erjavec, D. Tufi, and D. Varga. 2006. The JRC-Acquis: a multilingual aligned parallel corpus with 20+ languages. In *LREC*, Genoa, Italy.

Tiedemann, J. 2010. Context adaptation in statistical machine translation using models with exponentially decaying cache. In *the ACL Workshop on Domain Adaptation for Natural Language Processing*, Uppsala, Sweden.

Tiedemann, J. 2012. Parallel data, tools and interfaces in OPUS. In *LREC*, Istanbul, Turkey.

Wäschle, K., P. Simianer, N. Bertoldi, S. Riezler, and M. Federico. 2013. Generative and discriminative methods for online adaptation in SMT. In *MT Summit*, Nice, France.