# The Repetition Rate of Text as a Predictor of the Effectiveness of Machine Translation Adaptation

**Mauro Cettolo** cettolo@fbk.eu
**Nicola Bertoldi** bertoldi@fbk.eu
**Marcello Federico** federico@fbk.eu
FBK, Fondazione Bruno Kessler, 38123 Povo, Trento, Italy

**Abstract**

Since the effectiveness of MT adaptation relies on the text repetitiveness, the question on how to measure repetitions in a text naturally arises. This work deals with the issue of looking for and evaluating text features that might help the prediction of the impact of MT adaptation on translation quality. In particular, the repetition rate metric, we recently proposed, is compared to other features employed in very related NLP tasks. The comparison is carried out through a regression analysis between feature values and MT performance gains by dynamically adapted versus non-adapted MT engines, on five different translation tasks. The main outcome of experiments is that the repetition rate correlates better than any other considered feature with the MT gains yielded by the online adaptation, although using all features jointly results in better predictions than with any single feature.

## 1 Introduction

Language and content repetitiveness[1] is a key factor for the successful deployment of translation memories (TMs) (Somers, 2003) as well as statistical machine translation (MT) (Koehn, 2010). The capability of a TM to provide useful translation suggestions for a text segment relies on the chance of finding segments with very similar content – i.e. with a significant percentage of overlapping words – inside a large repository of already translated texts. On the other hand, statistical MT also relies on the assumption that the segment to be translated shares with the training data a significant amount of patterns, from single words to groups of words.

Advances on the integration of human post-editing into MT have recently revealed the potential of incremental and online MT adaptation. While doing their job, post-editors are also generating fresh in-domain training data. Adding these data to the MT engine would for instance mean letting MT focus on the lexical choices of the post-editor or even avoid future translation mistakes. Given that the *internal knowledge* of statistical MT is mainly represented by translation patterns, the potential impact of MT adaptation strictly depends on how much such type of patterns will re-occur in future sentences. We can hence claim that text repetitiveness is a precondition for the effectiveness of MT adaptation in general, whether this is performed on sentence batches as for incremental adaptation, or single sentence as in online adaptation. Thus, the question becomes how to measure repetitions in a text in a way to help the prediction[2] of the impact of MT adaptation on translation quality for that text. Such a prediction

[1] In this paper the word *repetitiveness* is **not** used with a negative meaning, e.g. boring, unpleasant.

[2] Although sometimes they are given slightly different meaning, in this work we consider *prediction* and *forecast* as synonyms.

could of course be useful to avoid the cost and even damage of applying adaptation in case the text results unfit or guide the choice among alternative adaptation methods.

This work deals with the issue of identifying and evaluating source text features that do significantly predict the performance of MT adaptation. Actually, the analysis of text and the design of features for modeling text characteristics are issues investigated in various NLP topics. As a consequence, many features have been already proposed and investigated by the scientific community which we can draw inspiration from; nevertheless, none of them was specifically designed for capturing the repetitiveness of text. Indeed, in the case of MT related tasks, like the quality estimation of MT output, in addition to features computed on the source text, features have been proposed which involve the translated/target text or even the MT models: although they can be really effective, we focus our investigation to the source side only, since we are interested in deciding what kind of MT system is most suitable for translating a given text before having any MT engine at disposal.

In this paper we experimentally assess the repetition rate, that we recently proposed in (Bertoldi et al., 2013) where no support to its effectiveness was provided, as a single light measure to characterize a full document to be translated. Roughly, the repetition rate computes the rate of event types (single words and $n$-grams) that occur more than once in a text; for making this statistics independent from the size of the document, it is computed on a fixed-size sliding window. We measured the prediction power of the repetition rate on several MT adaptation tasks and compared it against other text features that were proposed for very related NLP tasks. The comparison was carried out through a regression analysis between feature values and MT performance gains by an online adapting MT engine versus a static, non-adapted MT engine. Five different experimental tasks were considered, defined over two domains and three language pairs.

The main outcome of experiments is that the repetition rate correlates better than any other considered feature with the MT gains yielded by the online adaptation, although using all features jointly results in better predictions than with any single feature. Therefore, it seems feasible to decide in advance whether to activate or not the online adaptation procedure, just looking at the values of very few features, even just the repetition rate, of the text to be translated.

The remainder of the paper is organized as follows. First, an overview of related works with particular attention to text features is provided in Section 2. The repetition rate is formally described in Section 3, while Section 4 lists the other features we have selected for comparison purposes. Data, their analysis, the experimental setup and results are presented and commented in Section 5. A summary and the list of future works end the paper.

## 2 Related Work

To our knowledge this is the first work that deals with the problem of predicting the effectiveness of MT adaptation by means of features of the input text. On the contrary, the identification of text properties has been an essential problem in many NLP tasks, like text categorization, readability assessment, text comprehension, text complexity evaluation, automatic text-plagiarism detection, source and translation classification, information retrieval. Moreover, a number of features have been used in previous work for quality (also referred to as "confidence") estimation of MT output (Blatz et al., 2003), the task closer to the problem we are dealing with.

Aware of not being exhaustive, we survey a list of features used in some of those tasks that in principle could be useful for our purposes.

Blatz et al. (2003) describe 91 sentence-level confidence features and some additional word level features used in experiments. We cannot borrow most of them because they either are dependent on the MT models or involve the target text, and as stated in the introduction we are not interested in features with such dependency; concerning the measures on the source side, we think that the source length is not relevant for us, the log-probability and the perplexity of the source sentence are somehow included in our investigation, while the twelve quartile range measures of the source $n$-gram frequency deserve a more thorough discussion. They are defined as follows: Each list of distinct $n$-grams in the training corpus is first ordered by frequency and then split into four parts containing approximately an equivalent number of elements (quartiles); for each source sentence, the percentage of 1-, 2-, and 3-grams in each of the four frequency quartile ranges is then computed, for a total of 12 values. Since they characterize single sentences, the 12 values cannot be used as they are to model a whole text; therefore, we will not consider them in our experiments. Nevertheless, in the definition of our repetition rate, the rationale behind those quartiles is somehow considered: in fact, as we will see, we partition the $n$-grams into two groups, depending whether they occur once or more times.

In text categorization (Sebastiani, 2002), features such as single tokens or stems are mostly used. In the typically employed bag-of-words representation, information about dependencies and the relative position of tokens are not used. Anyway, they can be introduced at some extent through phrasal features consisting of more than one token: syntactic and statistical phrases ($n$-grams) have been investigated for a long time and many works report classification improvements over the use of single tokens, especially by introducing not too long $n$-grams (Fürnkranz, 1998), outcome that we exploit by defining the repetition rate over 1- to 4-grams.

Readability assessment is a form of text classification aiming at retrieving texts that suit a particular target reading level. In a school setting, it can help teachers to find texts appropriate to their students; other real-life contexts where it can play an important role are those involving people with intellectual disabilities, dyslexics, immigrant populations, and second or foreign language learners. Commendably, Vajjala and Meurers (2012) present dozens of features used in previous research on text readability and complexity and group them into three broad categories: lexical, syntactic and traditional features. Examples of features from the first group are the type-token ratio (see Section 4) and the lexical density, defined as the ratio of the number of lexical word tokens (nouns, adjectives, verbs, adverbs) and the number of all tokens (total number of words) in the analysed text. Syntactic features include mean length of clauses and sentences, and co-ordinate phrases and complex nominals per clause. The average sentence length in words and the number of characters or syllables per word are listed as traditional features. Reported experiments showed that the most predictive features are from the lexical group, result that led us to include them in our comparison.

Plagiarism detection can be divided into two main strategies, namely intrinsic plagiarism detection, that utilizes only information within the suspected document, and external plagiarism detection, that compares the suspected document against a set of possible sources. Just as a hint, the last “International competition on plagiarism detection” focused on external plagiarism detection only (Potthast et al., 2013). For both types, first works relied on $n$-grams, possibly sorted to bring them into a canonical form which cancels out plagiarism obfuscation.[3] Further efforts were devoted to text pre-processing, like synonym normalization, stemming, stop words deletion. Successively, attention was extended to lexical, syntactic and semantic features, like reordering and alignment of words, POS and phrase tags, semantic similarity of sentences, etc. (Lin et al., 2012).

The same or similar features mentioned so far also appear in works on translationese: $n$-grams in (Baroni and Bernardini, 2006); POS-based features and average sentence length, parse tree depth, proportion of simple/complex sentences, ambiguity as the average of senses per word, word length as the proportion of syllables per word, lexical richness, and information load as the proportion of lexical words to tokens in (Ilisei et al., 2010); most frequent words and

[3] Obfuscation is the strategy adopted by real plagiarists to rewrite their source passages in order to make detection more difficult.

a list of some hundred function words are instead used in (Islam and Hoenen, 2013) and (Koppel and Ordan, 2011), respectively.

## 3 Repetition Rate

We recently introduced the repetition rate (Bertoldi et al., 2013) as a way to measure the repetitiveness inside a text, by looking at the rate of non-singleton $n$-gram types ($n$=1. . .4) it contains. As shown there, 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 different sized documents, statistics are collected on a sliding window of one thousand 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 $n$-gram types in $S$, and $V(n)$ is the total number of $n$-gram types in $S$. RR ranges between 0 to 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).

In addition to get RRs that are comparable across texts of different lengths, the reason for using a sliding window is to preserve as much as possible the sequential structure of the original text, and hence its linguistic features, as opposed to what would happen if the sentences to be processed together were sampled.

## 4 Features for Comparison

### 4.1 Lexical features

Type-token ratio (TTR) is the ratio $T/N$ of the number $T$ of word types to the total number $N$ of word tokens in a text. It has been widely used as a measure of lexical diversity or lexical variation in language acquisition studies. However, since it is dependent on the text size, various alternative transformations of TTR came into existence. Then, besides TTR, we also considered Vajjala and Meurers (2012): square root TTR, defined as $T/\sqrt{N}$; corrected TTR, $T/\sqrt{2N}$; and bilogarithmic TTR, $\log T / \log N$.

### 4.2 Entropy-based features

In information theory, perplexity is a measurement of how well a probability distribution or probability model predicts a sample. In case of a discrete probability distribution $p$, the perplexity is defined as

$$PP = 2^{H(p)} = 2^{-\sum_x p(x) \log_2 p(x)}$$

where $H(p)$ is the entropy of the distribution and $x$ ranges over events. In NLP, perplexity is a way of evaluating LMs: the better the model for a given text, the lower the PP computed on that text. In our experiments, PP refers the source side of considered subsamples (see Section 5.3) with respect to LMs estimated on the source side of bitexts used for training the models of MT engines; in fact, only the source side of the text to be translated is assumed to be available.

Clearly, we are interested in intrinsic features of texts; on the contrary, the PP of a document is computed with respect to an “external” LM; therefore, we also considered a “self-contained” version of the perplexity, named `incremental perplexity` (incPP): for each segment of a document, its perplexity is computed on the LM estimated on previous segments

| domain | pair | segments | tokens | |
|---|---|---|---|---|
| | | | source | target |
| IT | en→it | 1,614 | 14,388 | 14,837 |
| | en→fr | 1,614 | 14,388 | 15,860 |
| Legal | en→it | 472 | 10,822 | 11,508 |
| | en→fr | 472 | 10,822 | 12,810 |
| | en→es | 472 | 10,822 | 12,699 |

Table 1: Overall statistics on parallel data used for evaluation purposes: number of segments and running words of source and target sides.

and the so-far PP value is incrementally updated; this procedure is iterated until the whole document is processed.

### 4.3 Out-of-vocabulary rate

The out-of-vocabulary rate (OOV) measures the number of unknown words; usually, it is expressed in percentage. We computed it for each considered subsamples with respect to the same external LMs used for the computation of the perplexity. Although the OOV can hardly be associated to repetitions, we decided to test it as well because some dynamic adaptation techniques, as that of our experiments, can learn unknown words and then yield MT performance gains.

## 5 Experiments and Evaluation

The investigation presented in this paper has been conducted on data employed in field tests organized by the MateCat project[4] which is developing a Web-based CAT tool for professional translators that integrates new MT functions, like offline and online adaptation performed on user feedback. Texts belong to two domains, namely information technology (IT) and legal (LGL); the language directions are from English into French and into Italian for both domains, while Spanish is the target language for the LGL domain only.

### 5.1 Evaluation data

For the IT domain, the evaluation document was supplied by the industrial partner of MateCat and consists of 1,614 segments.

For the LGL domain a document (2013/488/EU) was taken from the website of the European Union law,[5] for which translations into the three languages of interest were available. The document was pre-processed so that the segments of the three versions were all aligned. The full document consists of 605 segments and 13,900 English words; the first segments including about 3,000 words were used for development purposes, while the last 472 segments have been used in the experiments reported here.

Table 1 provides some statistics of evaluation texts. The target word counts refer to human references. Note that for each domain, the document to be translated is shared among all language-pairs.

### 5.2 Preliminary analysis

First of all, we checked if repetitions are randomly distributed or, on the contrary, if some words tend to re-occur more frequently in some portion than in others of the evaluation sets. Figures 1 plot the position of English words of the evaluation sets of the two domains: each

[4] http://www.matecat.com

[5] http://eur-lex.europa.eu

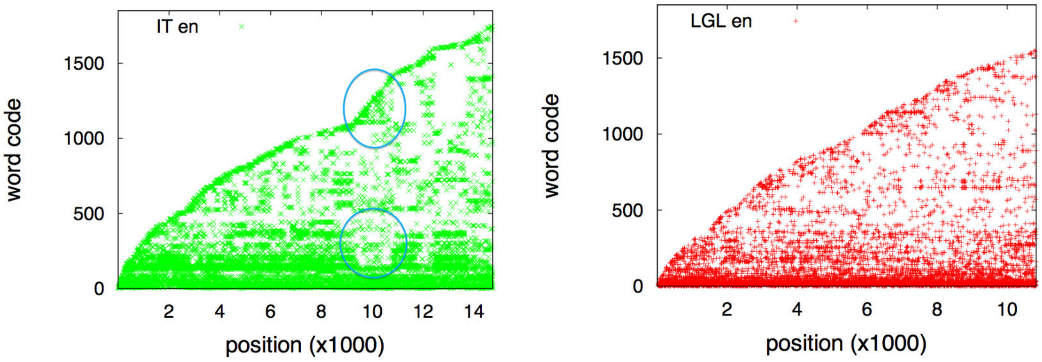

Figure 1: Word positions in the source side of the two test sets: each point $(x, y)$ says that the word whose code is $y$ occurs at position $x$ in the text. The circles show an isolated cluster of repeated words (top) replacing other common words (bottom).

word encountered in the text is assigned a progressive index (ordinate) and for all positions (abscissa) where it occurs a colored point is added to the plot. The envelope of the curves corresponds to the dictionary growth, while the distributions of points show those areas (if any) where repetitions are concentrated. The most evident deviation from the uniform distribution occurs for a portion of one thousand words of the IT test set centered at position 10,000 (top blue circle): a definitely higher growing rate of dictionary can be seen there, with many new words that re-occur often just in that text window. As an example, this is the segment number 1026 (out of 1614), at the beginning of that part of the document:

*If Web Services is enabled while deploying Weblogic 12c...*

The words *Web*, *Services*, *Weblogic* and *12c* are new, that is they were not observed before. Moreover, they re-occur just within the successive 100 segments (29, 16, 13 and 5 times, respectively) but no more in the remaining four hundred segments.

Another interesting phenomenon of that portion of the document is highlighted by the down blue circle: many words whose index is close to 200 do not occur there, while they appear quite uniformly in the rest of the text. For example, the words *return* (code 212) and *value* (code 250), which are observed 141 and 118 times in overall, are much less frequent in the hundred segments between 1026 and 1125, where *value* occurs just twice, *return* never.

That excerpt is a striking example of a text which differs from the rest of the document in terms of repetitions. Other portions isolated from the rest can be seen not only in the IT document but even in the Legal text, although to a lesser extent. Such a “localism” in the repetitions yielded us to perform measures on subsamples (windows) rather than on the whole documents, as described in Section 5.3.

For each test set (again, for the English source side only), Figure 2 plots the cumulative distributions of the distances between repetitions: each point $(x, y)$ of the curves says that repetitions distant no more than $x$ positions cover a percentage of all repetitions equal to $y$. For distances showed in the plot (lower than 120), the IT curve is above the LGL curve by 7-8 percentage points: it means that repetitions in the IT document occur significantly closer than in the LGL data. As a consequence, those adaptation methods affected not only by the amount of repetitions but also by their closeness, will be more effective in the IT domain than in the LGL domain.

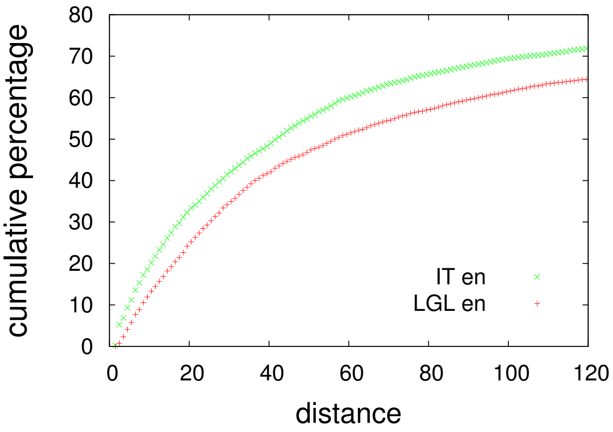

Figure 2: Cumulative distribution of repetition distances in the source side of the two test sets: each point $(x, y)$ of curves says that the repetitions occurring at a distance not larger than $x$ are the $y\%$ of the total amount of repetitions.

### 5.3 Experimental setup

The main goal of this investigation is to discover if the features listed in Sections 3 and 4 can predict the effectiveness of adaptation metrics, and in case to what extent. A straightforward approach is to measure the correlation between the values of features and of automatic MT quality metrics. As we are interested in the impact of text features on MT adaptation, we chose as target values for our predictors the relative gains in MT scores achieved with adaptation, i.e. the difference between the MT scores of the engines with and without the online adaptation module. Concerning which MT quality metric, for this investigation, we focused on the BLEU score, leaving the experiments on other metrics to future activity.

The localism observed in Section 5.2 determined us to perform the measure of features and of BLEU gains on shifting windows of text. The involvement of windows has also the positive side-effect to allow the computation for each test set of a number $N$ of `(features, gain)` pairs instead of a single pair got from the whole document. Figure 3 illustrates the scheme used to compute a list of `(features, gain)` pairs for a given document. It is assumed that the source text and the reference translation of that document are given; moreover, two MT engines are available, one representing the reference system (in our experiments, that without the adaptation module), the other being the boosted system which should yield some performance gain (in our experiments, that including the online adaptation module). First, the whole source text is translated by the two engines, so that the two automatic translations $\text{MT}_1$ and $\text{MT}_2$ are obtained. Then, for each window $W$ of text, the BLEU scores of the two translations are computed and their difference is paired to the text feature(s) we are interested in, like the RR of the source side. Once the whole document have been processed by means of the sliding window $W$, the procedure ends by outputting a record of $N$ pairs, which defines the dataset for the computation of the correlation. In our experiments, the size of $W$ is set to 2,000 words, a reasonable trade-off between the needs of preserving the localism and of reliable computations. $N$ is equal to the number of considered windows, which is determined by the window size, the size of the whole document and the moving step. Since the two sizes were given, the moving step was chosen so that $N$ is large enough to allow a reliable computation of

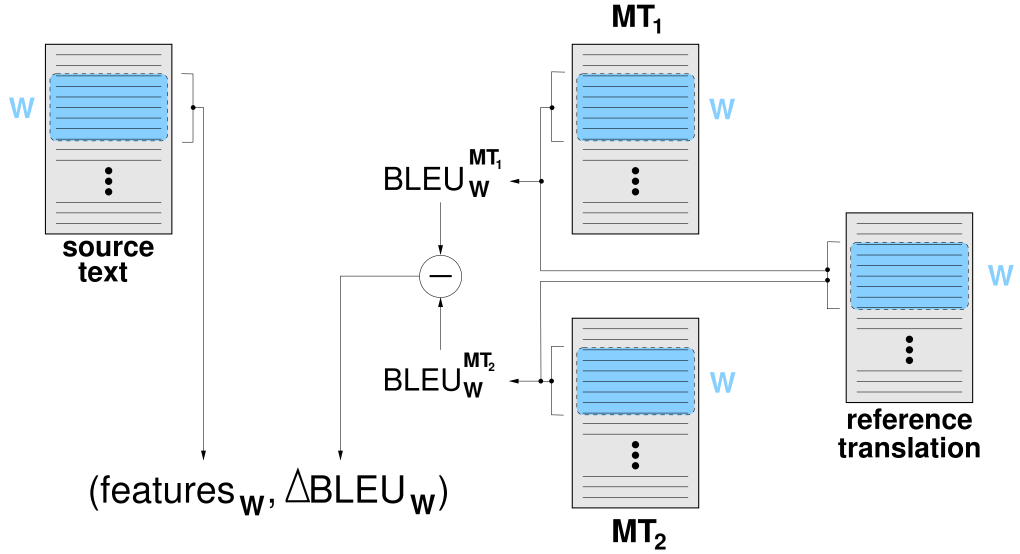

Figure 3: Scheme for the computation of `(features, MT gain)` pairs.

the correlation ($N \geq 100$).

For measuring the statistical dependency we are interested in, several regression algorithms could be used, such as polynomial regression, robust regression, regression trees, etc. We decided to employ support vector machines (SVMs) (Vapnik, 1995). SVMs are supervised learning models used for classification and regression analysis and are particularly suitable for our problem where not large training data are involved. In fact, SVM can generalize complicated input patterns with only a very few support vectors.

Practically, we used the LIBSVM (Chang and Lin, 2011), a software that provides support vector regression (SVR). In particular, we adopted a linear kernel with a $\epsilon$-SVR, since in a preliminary investigation this setup overtook other kernel types (we tried: polynomial, radial basis function and sigmoid). In $\epsilon$-SVR the goal is to find a function $f(x)$ that has at most $\epsilon$ deviation from the actually obtained targets $y_i$ for all the training data and at the same time as flat as possible. The model produced by $\epsilon$-SVR only depends on a subset of the training data, because the cost function for building the model ignores any training data that is close (within a threshold $\epsilon$) to the model prediction. The two parameters C and $\epsilon$ in loss function of $\epsilon$-SVR were set by running a randomized parameter optimization (Bergstra and Bengio, 2012).

For $\epsilon$-SVR, LIBSVM outputs mean squared error (MSE) and the squared correlation coefficient $r^2$: in our experiments, they were computed by cross validation on the basis of a 10-parts split of evaluation sets.

### 5.4 SMT engines

As already stated, two domains and three language pairs, for a total of five different tasks, are involved in our experiments. In the following two sections, training data and SMT engines involved in the experimental stage are described.

#### 5.4.1 Training data

For training purposes we relied on several language resources, including parallel corpora and translation memories. As far as the IT domain is concerned, software manuals from the OPUS corpus (Tiedemann, 2012), namely KDE4, KDE4-GB, KDEdoc, and PHP were used. They are all publicly available. In addition, a proprietary large translation memory (TM), that is a collection of parallel entries, was employed. It mostly consists of real projects on software documentation commissioned by a specific customer.

For what concerns the legal domain, the publicly available JRC-Acquis collection (Steinberger et al., 2006) was used, which mostly includes EU legislative texts translated into 22 languages.

Table 2 provides detailed statistics on the actual bitexts used for training purposes. In particular, the `train` entries refer to the whole generic training texts, while `development set` entries to additional data on which the parameters of the phrase-based MT model were optimized.

The `domain selection` entry of the IT en→fr task refers to data selected from out-of-domain texts (Giga English-French, United Nation, and Common Crawl corpora[6] (Bojar et al., 2013)) by using the in-domain text as seed in the method proposed by Axelrod et al. (2011) and available within the XenC toolkit (Rousseau, 2013); this was done to augment the amount of training data, since the size of in-domain text available for that language pair (15.4/17.9 million words) is about four times smaller than for the other tasks.

| domain | pair | corpus | segments | tokens | |
|---|---|---|---|---|---|
| | | | | source | target |
| IT | en→it | train | 5.4 M | 57.2M | 59.9M |
| | | development set | 2,156 | 26,080 | 28,137 |
| | en→fr | train | 1.1 M | 15.4M | 17.9M |
| | | domain selection | 1.2 M | 20.0M | 22.2M |
| | | development set | 4,755 | 26,747 | 30,100 |
| Legal | en→it | train | 2.7 M | 61.4M | 63.2M |
| | | development set | 181 | 5,967 | 6,510 |
| | en→fr | train | 2.8 M | 65.7M | 71.1M |
| | | development set | 600 | 17,737 | 19,613 |
| | en→es | train | 2.3 M | 56.1M | 62.0M |
| | | development set | 700 | 32,271 | 36,748 |

Table 2: Overall statistics on parallel data used for training and development (tuning) purposes: number of segments and running words of source and target sides. Symbol $M$ stands for $10^6$.

### 5.4.2 Decoders

The SMT systems have been built upon the open-source MT toolkit Moses (Koehn et al., 2007). The translation and lexicalized reordering models were trained on parallel training data, i.e. entries `train` (IT English-to-Italian, legal English-to-Italian/French/Spanish tasks), and `train` plus `domain selection` (IT English-to-French task) of Table 2. Back-off 5-gram language models smoothed with the improved Kneser-Ney technique (Chen and Goodman, 1999) were estimated on the target side of the available bilingual training data. The standard MERT procedure provided within the Moses toolkit was used to optimize the weights of the log-linear interpolation model on development sets whose content is coherent to training data and of adequate size (entries `development set` of Table 2).

For each task, an SMT engine was built over the above mentioned models and used for the standard translation of the test sets. Since the models do not change during the translation,

[6] Available from http://www.statmt.org/wmt13/translation-task.html.

| pair | MT engine | IT BLEU | IT TER | IT GTM | LGL BLEU | LGL TER | LGL GTM |
|---|---|---|---|---|---|---|---|
| en→it | STA | 57.5 | 26.3 | 78.6 | 35.0 | 49.1 | 64.6 |
| | DYN | 64.6 | 21.0 | 83.7 | 36.5 | 48.1 | 65.6 |
| en→fr | STA | 41.4 | 37.9 | 69.9 | 36.4 | 49.1 | 65.1 |
| | DYN | 56.3 | 28.9 | 78.8 | 41.2 | 45.4 | 68.7 |
| en→es | STA | – | – | – | 36.4 | 50.2 | 65.6 |
| | DYN | – | – | – | 41.2 | 46.1 | 69.4 |

Table 3: Overall performance of MT engines with respect to human references on evaluation sets.

these systems are named *static* (STA) and represent the reference systems.

For each static system, a companion *dynamic* (DYN) system has been built that dynamically adapts to post-edits, as they become available. More in detail, in the DYN system a global model, which is the same as that in the STA system, is combined with a local model, which is empty when the learning process starts. The combined model is used to generate the translation of the current input. The user post-edits the automatic translation and the amended text feeds back the system. The local model is then refined on the user feedback and the combined model updated, ready to translate the next segment. The process iterates over all sentences of the document to be translated.

In our systems, the local model is implemented by a caching mechanism. The caching regards both translation and language models: phrase pairs extracted from the alignment of the source and post-edit are extracted and inserted into the cache-based translation model, while $n$-grams of the post-edit fill the cache-based language model. More details are provided in (Bertoldi et al., 2013). Note that in our experiments the post-editing is simulated by using human references.

### 5.5 Results and comments

First of all, Table 3 provides BLEU (Papineni et al., 2002), TER (Snover et al., 2006) and GTM (Turian et al., 2003) scores computed on the evaluation documents with respect to human references for each of the five considered translation tasks.

In both domains, but especially for IT, the improvements over the static systems yielded by the dynamic adaptation technique are remarkable. Focusing on the BLEU score, in the IT domain it improves by more than 7 absolute points for English-to-Italian (57.5 to 64.6), and almost 15 absolute points for English-to-French (41.4 to 56.3); in the LGL domain, the gain is quite limited in English-to-Italian (1.5 absolute points), but definitely notable – almost 5 points – for the other two directions.

The successive experiments regard the actual prediction of MT score gains on the basis of text features. Following the shifting window scheme described in Section 5.3 and shown in Figure 3, for each window the BLEU difference (ΔBLEU) of `STA` and `DYN` MT was recorded and features computed; the total number of considered windows was 135 for the IT test set, 200 for the LGL domain. The linear regression between ΔBLEU and either each single feature or all features was then calculated by means of the LIBSVM software (Section 5.3): the cross-validation MSE and $r$ are collected in Table 4. Postponing for a while any comment on the IT en→it task which is an outlier, the main outcomes on the other four tasks are:

| cvMSE/cv-$r$ | IT |  | LGL |  |  |
|---|---|---|---|---|---|
|  | en→fr | en→it | en→es | en→fr | en→it |
| RR | 5.98/0.76 | 11.35/0.05 | 0.81/0.51 | 1.15/0.46 | 0.98/0.60 |
| TTR | 8.25/0.64 | 11.58/0.02 | 0.87/0.43 | 1.06/0.53 | 1.19/0.46 |
| sqrtTTR | 7.57/0.68 | 11.74/0.00 | 0.90/0.39 | 2.15/0.33 | 1.17/0.46 |
| crrTTR | 7.57/0.68 | 12.08/0.08 | 3.66/0.21 | 1.09/0.51 | 1.18/0.47 |
| blgTTR | 8.45/0.63 | 11.11/0.15 | 0.89/0.42 | 1.07/0.52 | 1.18/0.46 |
| PP | 14.06/0.26 | 11.11/0.01 | 1.15/0.05 | 1.46/0.08 | 1.55/0.10 |
| incPP | 8.37/0.63 | 11.43/0.07 | 0.88/0.43 | 1.04/0.55 | 1.18/0.47 |
| OOV | 11.04/0.46 | 11.53/0.22 | 1.07/0.22 | 1.45/0.22 | 1.55/0.29 |
| all features | 5.85/0.77 | 6.35/0.69 | 0.56/0.69 | 0.81/0.66 | 0.55/0.79 |

Table 4: Cross validation mean squared errors (cv-MSE) and correlation coefficients (cv-$r$) between $\Delta$BLEU and text features for the five different tasks.

• RR is highly correlated with $\Delta$BLEU in any task;

• in three out of four tasks, RR is the best predicting feature; only in the LGL en→fr task it is not the best, but it is anyway competitive both in terms of correlation and of mean squared error;

• although it is a good predictor, TTR performs pretty worse than RR;

• the variants of TTR proposed for making its measure independent from the size of the text do not seem to outperform the original formulation;

• incPP undeniably outperforms PP, it is competitive with TTR and it is even the best predictor in the LGL en→fr task;

• if considered all together, the features definitely correlate with MT gains better than individually, especially on LGL tasks; for the IT en→fr however, it has to be considered that some single features are indeed very good predictors (mainly RR), with performance hard to beat.

Concerning the IT en→it task, no single feature is capable to effectively predict the MT gain; on the contrary, the correlation is very high if they work together. The latter outcome is positive because it confirms that the gains from MT adaptation are somehow predictable; from the other side, inefficacy of individual features is disappointing, especially considering the opposite results on the companion IT en→fr task, where the same source document is translated. For further investigation, the scatter plot (with the regression line) of the 135 (RR,$\Delta$BLEU) points of the IT en→it task is shown in the plot on the left of Figure 4. It is evident the weak dependency between the two variables, as already suggested by the negligible correlation coefficient (0.05) reported in Table 4.

Nevertheless, the points appear not to be randomly distributed; in fact, by naively splitting the test set in four equal-sized and contiguous parts, the points are grouped as shown in the second plot of Figure 4, where much smaller deviations from the regression lines are revealed: in fact, the correlation coefficient for each quarter is 0.84, 0.56, 0.54 and 0.54, respectively. And they could be even higher if the split was more precise, for example by separating the II and III blocks so that the blue points with RR higher than 35 could be merged with the green points. A similar behavior has been observed for the other features, as well. That means that for the IT en→it task, the linear dependency between single features and MT gains exists but it is local and changes through the test document. This issue will be investigated in the future.

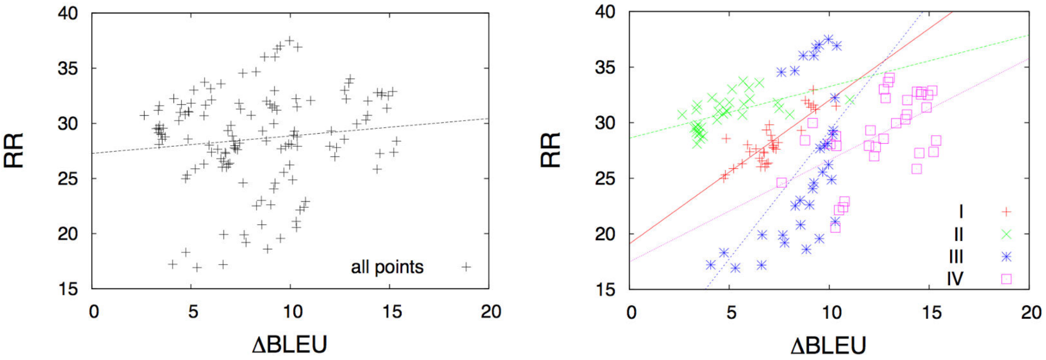

Figure 4: IT en→it task, RR vs. $\Delta$BLEU: scatter plots and regression lines for all points (left) and for quarters (right).

## 6 Conclusions

In this paper, we experimentally assessed the repetition rate as a novel text feature for the prediction of the effectiveness of MT adaptation. Taking the gains yielded by a paradigmatic online adaptation technique as the dependent variable, we performed a regression analysis over a number of text features considered as independent variables. Results on five MT tasks, different in terms of domain or language pair, showed that the repetition rate is the best predictor, although the most accurate regression model uses the features all together.

Concerning the future work, besides the problem on the IT en→it task sketched out at the end of Section 5.5, we will handle some other open issues. First of all, here we focused our analysis to MT gains expressed in terms of BLEU score; we think it is recommendable to consider other measures as well, like TER and GTM, to be sure that our outcomes are not "metric-dependent". Another interesting extension regards "negative" text samples: in previous papers, we showed that if the RR of the text to be translated is not high enough, the online adaptation cannot significantly improve the reference performance of the static MT engine; then, we will extend the systematic investigation presented here to such problematic tasks, like the translation of news and of TED talks.

Results showed that taken all together, the considered features correlate better than individually with MT gains: it will be shown the relative contribution of each single feature to the overall performance. Finally, the assessment of minor aspects of our experimental setup should be considered: (i) the size $n$ of the $n$-grams involved in the definition of the RR, here set to 4; (ii) the size of the subsample $S$ on which the RR is computed, here set to $1,000$ words (see Section 3); and (iii) the size of the sliding window for handling the localism, here set to $2,000$ words (see Section 5.3).

## Acknowledgements

This work was supported by the MateCat project (grant agreement 287688),[4] which is funded by the EC under the 7th Framework Programme.

## References

Axelrod, A., X. He, and J. Gao. 2011. Domain Adaptation via Pseudo In-Domain Data Selection. In *Proc. of EMNLP*, pp. 355–362, Edinburgh, UK.

Baroni, M. and S. Bernardini. 2006. A new approach to the study of translationese: Machine-learning the difference between original and translated text. *LLC*, 21(3):259–274.

Bergstra, J. and Y. Bengio. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305.

Bertoldi, N., M. Cettolo, and M. Federico. 2013. Cache-based Online Adaptation for Machine Translation Enhanced Computer Assisted Translation. In *Proc. of MT Summit*, pp. 35–42, Nice, France.

Blatz, J., E. Fitzgerald, G. Foster, S. Gandrabur, C. Goutte, A. Kulesza, A. Sanchis, and N. Ueffing 2003. Confidence Estimation for Machine Translation Technical report. Johns Hopkins University, Baltimore, US-MD. Available at http://web.eecs.umich.edu/˜kulesza/pubs/confest_report04.pdf

Bojar, O., C. Buck, C. Callison-Burch, C. Federmann, B. Haddow, P. Koehn, C. Monz, M. Post, R. Soricut, and L. Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proc. of WMT*, pp. 1–44, Sofia, Bulgaria.

Chang, C.-C. and C.-J. Lin. 2011. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27.

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

Fürnkranz, J. 1998. A study using n-gram features for text categorization. TR 98-30, OEFAI - Austrian Research Institute for Artificial Intelligence.

Ilisei, I., D. Inkpen, G. Corpas Pastor, and R. Mitkov. 2010. Identification of translationese: A machine learning approach. In *CICLing*, Lecture Notes in Computer Science, pp. 503–511. Springer.

Islam, Z.l and H. Armin. 2013. Source and translation classification using most frequent words. In *Proc. of IJCNLP*, pp. 1299–1305, Nagoya, Japan.

Koehn, P. 2010. *Statistical Machine Translation*. Cambridge University Press.

Koehn, P., et al. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *ACL Companion Volume Proc. of the Demo and Poster Sessions*, pp. 177–180, Prague, Czech Republic.

Koppel, M. and N. Ordan. 2011. Translationese and its dialects. In *Proc. of ACL:HLT*, pp. 1318–1326, Portland, US-OR.

Lin, W.-Y., N. Peng, C.-C. Yen, and S.-de Lin. 2012. Online plagiarism detection through exploiting lexical, syntactic, and semantic information. In *Proc. of ACL*, pp. 145–150, Jeju Island, Korea.

Papineni, K., S. Roukos, T. Ward, and W.-J. Zhu. bibtem 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. of ACL*, pp. 311–318, Philadelphia, US-PA.

Potthast, M., M. Hagen, T. Gollub, M. Tippmann, J. Kiesel, P. Rosso, E. Stamatatos, and B. Stein. 2013. Overview of the 5th international competition on plagiarism detection. In *Notebook Papers of CLEF 2013 LABs and Workshops*.

Rousseau, A. 2013. Xenc: An open-source tool for data selection in natural language processing. *Prague Bull. Math. Linguistics*, 100:73–82.

Sebastiani, F. 2002. Machine learning in automated text categorization. *ACM Computing Surveys*, 34:1–47.

Snover, M., B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proc. of AMTA*, Boston, US-MA.

Somers, H. 2003. Translation memory systems. In *Computers and translation: a translator's guide*, 35(3):31–48. Benjamins Translation Publishing Company.

Steinberger, R., B. Pouliquen, A. Widiger, C. Ignat, T. Erjavec, D. Tufiş, and D. Varga. 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *In Proc. of LREC*, pp. 2142–2147, Genoa, Italy.

Tiedemann, J. newblock 2012. Parallel Data, Tools and Interfaces in OPUS. In *Proc. of LREC*, Istanbul, Turkey.

Turian, J. P., L. Shen, and I. D. Melamed. 2003. Evaluation of machine translation and its evaluation. In *Proc. of MT Summit IX*, New Orleans, US-LA.

Vajjala, S. and D. Meurers. 2012. On improving the accuracy of readability classification using insights from second language acquisition. In *Proc. of NAACL-HLT Workshop on Innovative Use of NLP for Building Educational Applications*, Montreal, Canada.

Vapnik, V. N. 1995. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., US-NY.