Journal of Artificial Intelligence Research 54 (2015) 159-192 Submitted 02/15; published 09/15

Leveraging Online User Feedback to Improve Statistical Machine Translation

Lluís Formiga LFORMIGA@VERBIO.COM
Verbio Technologies, S.L.,
Loreto, 44, 08029 Barcelona

Alberto Barrón-Cedeño ALBARRON@QF.ORG.QA
Lluís Màrquez LMARQUEZ@QF.ORG.QA
Qatar Computing Research Institute
Hamad Bin Khalifa University,
Tornado Tower, Floor 10, P.O. Box 5825, Doha, Qatar

Carlos A. Henríquez CARLOS.HENRIQUEZ@UPC.EDU
José B. Mariño JOSE.MARINO@UPC.EDU
TALP Research Center - Universitat Politècnica de Catalunya,
Jordi Girona, 1-3, 08034 Barcelona

Abstract

In this article we present a three-step methodology for dynamically improving a statistical machine translation (SMT) system by incorporating human feedback in the form of free edits on the system translations. We target at feedback provided by casual users, which is typically error-prone. Thus, we first propose a filtering step to automatically identify the better user-edited translations and discard the useless ones. A second step produces a pivot-based alignment between source and user-edited sentences, focusing on the errors made by the system. Finally, a third step produces a new translation model and combines it linearly with the one from the original system. We perform a thorough evaluation on a real-world dataset collected from the Reverso.net translation service and show that every step in our methodology contributes significantly to improve a general purpose SMT system. Interestingly, the quality improvement is not only due to the increase of lexical coverage, but to a better lexical selection, reordering, and morphology. Finally, we show the robustness of the methodology by applying it to a different scenario, in which the new examples come from an automatically Web-crawled parallel corpus. Using exactly the same architecture and models provides again a significant improvement of the translation quality of a general purpose baseline SMT system.

1. Introduction

Statistical machine translation (SMT) has become a widespread technology, used by millions of people to satisfy a multiplicity of needs in their daily interactions and information seeking. In contrast to business-oriented translation services, on-line machine translation services (e.g., Google Translate, see Google Inc., 2015; Bing, see Microsoft Inc., 2015; Reverso, see Reverso-Softissimo, 2015) offer free general-purpose translations with fairly acceptable levels of quality and for a large number of language pairs. The fact that they are easily accessible from any computer, tablet or smartphone connected to the Internet has contributed to create a huge community of heterogeneous users.

However, SMT systems have significant limitations and produce translation errors at different levels (e.g., morphology and agreement, phrase structure and reordering, lexical selection, and fluency). This is not only due to the inherent complexity of the task but also to the limitations of the currently available translation models and training corpora, which might not be fully representative of the domain, genre and style of the texts to be translated. This behavior may cause frustration and fatigue to the users; but the users themselves are in the right position to spot such mistakes. The response of machine translation (MT) systems developers has been to allow users to provide feedback by proposing corrections to the system-generated translations. Gathering new and improved information from users' edits has shown to be a valuable resource to improve the translation systems in both on-line cost-free services (Simard, Goutte, & Isabelle, 2007; Ambati, Vogel, & Carbonell, 2010; Potet, Esperança-Rodier, Blanchon, & Besacier, 2011) and professional computer-assisted translation frameworks (Bertoldi, Cettolo, & Federico, 2013; Mathur, Mauro, & Federico, 2013). The raising interest of this topic in the MT community has emerged also in the form of research projects (e.g., MateCat, FAUST, and Casmacat, see European Comission - 7th Framework Program, 2010) specialized workshops (e.g., the Workshop on Post-editing Technology and Practice at MT Summit XIV) and special issues (such as the Machine Translation Journal Special Issue of Machine Translation on MT Post-editing).

In this article we explore the use of real translation feedback from non-professional users. Our aim is to automatically improve the general translation quality of the underlying MT system. This approach differs significantly from the more common setting in which professional post-editors are used to produce high-quality translations from an imperfect —automatic— output. In our setting users are casual, with limited skills, and sometimes having low command of the languages being translated. They perform free edits on the machine-translated text to produce a supposedly better alternative translation. This translation is sometimes a proper post-edition, but frequently it is partial, contains errors, or it is simply a piece of unrelated text. The challenge in this particularly noisy setting is being able to filter out part of the noise and select the potentially useful translation edits.

One advantage of this *crowd-sourcing* approach to MT enrichment is its potential to reach a vast community of contributors. This scenario conveys a mutual-interest framework: on the one side, an active user wills to correct translations as long as the system responds better to her needs in the future; on the other side, a system requires the input of an intelligent agent, which is able to provide information to improve its translation models. If a high level of engagement is achieved, a committed live community can constantly contribute to improve the free on-line translators. Another fundamental aspect necessary to reach circle is a mechanism that efficiently and accurately incorporates user feedback into the translation engine. *Accurately*, because we want the MT system not to repeat the same mistakes and, at the same time, not to worsen its overall translation quality; *efficiently* to engage users, we need the system to react quickly (if not instantaneously) to their feedback.

Even though the exploitation of user edits (UE) is a widespread practice with increasing interest, few methods aim at improving existing MT models. Most of the work is highly focused on translation dictionaries and centered in minimizing the out-of-vocabulary (OOV) ratio of the translations (Cettolo, Federico, Servan, & Bertoldi, 2013). Studying the performance of the enriched translation models on a variety of aspects of translation quality (e.g., morphology, word ordering, lexical selection, etc.) is an issue that deserves further attention from the MT community.

In this work we explore to which extent we can use translation edits collected from non-professional users of a commercial on-line translation portal to improve the translation quality of a general purpose SMT system. The main contribution is twofold. First, we address the noisy crowd-sourcing scenario by training supervised classifiers to identify useful UE instances. Second, we devise `SimTer`, a pivot-based method for aligning user-edited translations to both the source text and the original automatic translations, with an aim to detect the specific corrected errors and to build enriched translation models accordingly. Both aspects, UE filtering and pivot-based selection of phrase pairs, are novel and we show that they contribute significantly to a better translation quality. We support this claim by extensive experimentation and analysis. The improvement achieved is remarkable compared to a simple *corpus-concatenation* strategy, since we work with a relatively low quantity of UEs. Additionally, we conduct a manual evaluation of the output of the enriched system. This study reveals the strengths and weaknesses of the enriched system and the source of its improved results, which are not only achieved by means of the reduction of the OOV ratio. Interestingly, other more linguistically-founded aspects of translation quality are also improved. Finally, we show the generality of our approach by successfully applying the same architecture and models to a *noisy domain-adaptation* scenario, where the new examples come from an automatically-crawled bilingual corpora and filtering out noisy examples is a key aspect of the adaptation.

The rest of the article is organized as follows. Section 2 puts the current article in context by overviewing related work. Section 3 describes and locally evaluates our classification approach for identifying useful UE instances. Section 4 discusses our proposal for improving machine translation models with UEs. Section 5 presents our experiments on real datasets showcasing the proposed methodology. Finally, the conclusions are presented in Section 6.

2. Related Work

In this section we overview the most relevant work related to the two main contributions of this article: "automatically identifying useful user edits (UEs)" and "improving existing translation models (TMs) on the basis of such UEs."

Identifying the useful UE instances is necessary because many of the ones collected from non-professional users may not represent better translations as compared to the ones produced by the system itself. To the best of our knowledge, no other research on this particular topic exists. Some analog tasks can be found either on obtaining quality material by filtering automatically gathered corpora or on domain adaptation. Usually, the subsampling selection method (Foster, Goutte, & Kuhn, 2010; Axelrod, He, & Gao, 2011) is used when dealing with corpus selection problems. It consists of a simple rationale: a language model (LM) is created with reliable in-domain data and, subsequently, those parts of the corpus having a lower perplexity with respect to the model are selected. In our scenario, we have to select the best UE instead.

In our previous research on this topic (Pighin, Màrquez, & May, 2012; Barrón-Cedeño et al., 2013) we defined a basic set of features to perform such identification. We aim at capturing different aspects, such as: (*i*) whether the UE text is a more adequate translation from the source sentence than the automatic translation (computed with simple surface similarity features), (*ii*) whether the UE includes mistakes or typos, and (*iii*) whether the source contains mistakes or typos that prevent from obtaining sensible translations. The identification is then cast as a supervised classification problem using the above mentioned features. Our work extends this approach, which is explained and evaluated in Section 3.

Assuming a set of good edited translations (revised by an expert), the enrichment of the translation system involves two separate steps. First, an alignment between the source sentences and the edited translation is computed (*alignment*). Second, improved translation models are learned using these alignments (*adaptation*). Regarding the alignment step, it is widely accepted that the best possible alignment is obtained by adding to the training corpus, new sentence pairs in which the edited sentence is treated at the target-side and models are estimated from schratch (Hardt & Elming, 2010). However, the large amount of training data makes this approach computationally expensive; an obstacle to the goal of reacting quickly to the user feedback. To overcome this problem, several *incremental* alignment models have been proposed in the literature(Levenberg, Callison-Burch, & Osborne, 2010). With the exception of the *stream-based* translation approach, which adds or updates the original TM scores according to the new material (Ortiz-Martínez, García-Varea, & Casacuberta, 2010; Martínez-Gómez, Sanchis-Trilles, & Casacuberta, 2012; Mathur et al., 2013), the adaptation step is usually carried out by creating specific translation tables from the edited translations (using the standard phrase-extraction and phrase-scoring algorithms) and then combining them with the original translation tables. It is important to note that most of the work on incremental adaptation has been tested in the scenarios where references are used instead of UEs. Hence, related work mainly addresses simulated or artificial scenarios where the conclusions might not be totally representative. Only a few models use human-edited translations. Mathur et al. (2013) along with Bertoldi et al.(2013) rely on a business-oriented specific corpus in computer-assisted translation. Simard et al. (2007) introduced the so-called automatic post-editors (i.e., monolingual SMT systems designed to improve translation errors). Potet et al. (2011) considered a small corpus of UEs from non-professional users, but not an incremental methodology. On the basis of all previous aspects we describe next a few selected papers.

Hardt and Elming (2010) proposed an approach to produce approximate alignments. They defined an approximate alignment as the one that allows to extend a phrase table, even if it is not perfect or exact. Given a ⟨*source*, *user-edit*⟩ pair and baseline alignment (Och & Ney, 2003), they explored all the available links by selecting the ones producing the highest probability according to IBM model 4. Moreover, they improved the alignments applying two heuristics: the first non-aligned word in the source is aligned to the first non-aligned word in the UE and unlinked fragment pairs surrounded by corresponding alignments are linked. One drawback of this method is that the alignments produced are noise-sensitive, as they are built upon heuristics. This makes the methodology unpredictable and unstable to deal with words that have not been seen by the baseline alignment model. For the adaptation step, they built a separate phrase table with UEs and decoded with both phrase tables. They used approximate and exact `GIZA++` alignments and showed that the performance of the approximate alignment yields half of the improvement of that obtained by the `GIZA++` alignment. However, only simulated data, using references instead of actual human UEs, were considered in this work.

Ortiz-Martínez et al. (2010) and Martínez-Gómez et al. (2012) applied an incremental version of the Expectation Maximization (EM) algorithm (Neal & Hinton, 1998) that minimizes an error function with small sequences of mini-batched data. This paradigm is commonly known as *stream-based* translation, as small portions of data are processed over time. More specifically, they incrementally adapted seven models including language models, length penalty, phrase translation models, and distortion. This strategy is reported to perform reasonably well in non-stationary environments where fast adaptation is required, such as interactive machine translation (Ortiz-Martínez, Sanchis-Trilles, González-Rubio, & Casacuberta, 2013). For a general-purpose web-based scenario, the resulting models are too sensitive to the lately integrated data. In this approach no normalization was carried out due to its computational cost. Mathur and Federico (2013) opted for leaving the original features unaltered and added extra (normalized) feature that reflects the impact of the UEs in the adaptation of the system.

Simultaneously to Ortiz-Martínez et al. (2010), Levenberg et al. (2010) proposed an on-line strategy for phrase-based model enrichment using the stepwise EM alignment algorithm (Cappé & Moulines, 2009). The incorporation of new knowledge in this approach is based on the re-estimation of scores of the phrase table by re-computing the counts throughout a computationally efficient dynamic suffix array (Callison-Burch, Bannard, & Schroeder, 2005; Lopez, 2008). A suffix array contains the starting index of each suffix of the string containing the phrases in lexicographical order, allowing for an easy computation of on-the-fly translation probabilities for a given source phrase. A dynamic variant of the suffix array supports deletions and insertions, making it suitable for a stream-based approach. Moses uses this algorithm to provide an incremental training strategy (Haddow & Germann, 2011).

`mGIZA++` is a parallel implementation of the IBM and HMM models (Gao & Vogel, 2008). As a byproduct, it performs forced alignment,[1] which is an alternative to the step-wise and incremental EM approaches. `mGIZA++` builds multiple alignment models in parallel, allowing for filtering and merging them afterwards to produce the exact alignment for the aggregation. Consequently, one can obtain forcedly-aligned material and improved alignment models with the same quality as if the concatenated dataset had been used from the beginning. This tool is efficient in terms of processing time and storage requirements, making it suitable for exact incremental alignment.

In contrast to the alignment methods described above, the *pivot-based* approach tries to identify the specific edits performed by the user on the original translation and project them to the source (Henríquez, Mariño, & Banchs, 2011). More precisely, it uses the original translation as a pivot to obtain alignments between the source and the UE translation, with which to enrich the existing translation models. An advantage of this approach is that it allows to spot incorrect fragments in the translation, which are the potential source of errors in the MT system. Therefore, the new alignments obtained from the edited fragments are the ones to be promoted within the TMs of the enriched translator. Formiga et al. (2012) and

1. The term *forced alignment* refers to coercively aligning unseen parallel text by selecting the maximum probability given by the model, even if its value is low.

Blain, Schwenk, Senellart and Systran (2012) studied this idea further to perform adaptation with good results. The recently developed `MateCat` tool (Matecat, 2015), a Web-based CAT tool, is a good example in this category. This approach has three advantages: it is fast, it does not rely on baseline alignment models, and it has low memory requirements.

In all previous pivot-based alignment implementations, the edit distance is computed with a translation-error-rate (TER) alignment algorithm, which takes into account reordering operations and paraphrasing. As this strategy is the basis of the methodology presented in this article, we explain it in detail in Section 4.1. Concerning the adaptation strategy, Formiga et al. (2012) combined UE-specific translation models with the baseline models using Foster and Kuhn's (2007) interpolation with empirically-set weights. Blain et al. (2012) studied two decoding strategies considering the baseline and UE phrase-tables separately: *back-off*, if a phrase is not found in the baseline translation model, the phrase table is considered and *multiple decoding*, if the same phrase is found in both translation models, both translations and scores are used. They found multiple decoding to be the best strategy while restricting the TER alignment only to the substitution operations (i.e., neglecting addition, deletion, and shifting edits). Some refined combination methods have been presented recently. Sennrich (2012) used the L-BFGS algorithm (Byrd, Lu, Nocedal, & Zhu, 1995) to find the optimal interpolation values for each feature function of the TMs. Bisazza, Ruiz and Federico (2011) defined a fill-up strategy to complement the missing information of the original TMs. Nevertheless, these methods have not been challenged under an incremental scenario.

3. Learning to Classify User Edits

This section describes our strategy to identify useful user-edits for improving the machine translation system. We approach the task as a binary classification problem identifying the cases in which the edited translation is more adequate than the system-produced translation as positive. In doing so, we follow some of our previous work on human feedback filtering (cf. Section 2).

3.1 Training Corpus

As training material we used the English–Spanish Faust Feedback Filtering (FFF^{+}) [2] corpus, developed within the FAUST EU project.It contains 550 examples of real translation requests and user-edits from the Reverso.net translation Web service. Each example contains seven fields:

- `SRC` source sentence in English (user's translation request to the system);
- `TGT` automatic translation of `SRC` into Spanish;
- `UE` a potentially improved user-edit of `TGT` provided by the user;
- `BTGT` automatic translation of `TGT` back into English;
- `BUE` automatic translation of `UE` back into English;
- `LANG` the language set in the translator's interface; and

2. Available at `ftp://mi.eng.cam.ac.uk/data/faust/UPC-Mar2013-FAUST-feedback-annotation.tgz`.

`CL` class label, i.e., whether `UE` is a more adequate translation of `SRC` than `TGT`.

Observe that the language set in the translator's interface is a very likely indicator of the user's native language; a factor that may indicate that user edits in that language are more reliable. The 550 examples of FFF^+ were independently labeled as *acceptable* or *unnaceptable* edits by two human annotators —both were native Spanish speakers with high command of English. All cases of disagreement (∼100) were discussed until consensus was reached. The main criterion to decide whether the user-edit is acceptable is based on *translation adequacy* (i.e., the degree to which the meaning of the source sentence is conveyed in the translation). Concretely, `UE` is considered acceptable if it is strictly more adequate than `TGT`, even if still imperfect. We believed that this lax criterion of acceptability could work well as a proxy for *usefulness*, when thinking of enriching the MT system. For a detailed description of the annotation guidelines, including examples, one may refer to the work of FAUST (2013, Section 3.2). The levels of inter-annotator agreement achieved (Cohen's kappa 0.5–0.6) can be considered only moderately high. This fact evinces the inherent difficulty of this task, even for humans. The positive–negative distribution of the corpus is almost balanced: 50.5% versus 49.5%, indicating that the edits provided by casual users are very noisy.

3.2 Learning Features

We considered five sets of features to characterize the examples' fields and the relationships between them: *surface comparison*, *back-translation*, *noise-based*, *similarity-based*, and *quality estimation*. The first four sets require no external resources other than a MT system from the target back to the source language (having such a system at hand is very likely, as the necessary resources to build it are practically the same as for the original-direction system). This fact makes them particularly appealing for under-resourced languages. The fifth set is composed of a combination of well-known MT quality estimation measures. In all cases, features were extracted after text pre-processing, including case-folding and diacritics elimination —but original texts were incorporated into the translation system. Following, we provide a short description of the main principles guiding each family of features. The full list, comprising more than 90 individual features, is described in detail by FAUST (2013, Section 3.2).

3.2.1 Surface Features

They consider the text strings `SRC`, `TGT` and `UE`, and compute surface similarities among them at the level of word tokens and characters (length, length ratios, Levehnstein distance, vocabulary containment, etc.). There is also a binary feature to account for the language of the interface (`LANG`).

3.2.2 Back-Translation Features

These features account also for surface properties, but considered on the back-translations of `TGT` (`BTGT`) and `UE` (`BUE`). Levenshtein distances both at the token and character level as well as vocabulary intersections are included.[3]

3.2.3 Noise-Based Features

These are binary features intended to determine the likelihood of the text fragments to include noisy sections. We consider instances of `SRC` or `UE` that include characters' repetitions (longer than three characters) or tokens whose length is too high to be a regular word (>10 chars). Some of them try to determine a "length-based translation difficulty" by asuming that the longer sentences are harder to translate (we consider features for the ranges $[1-5]$, $[6-10]$, and $[11, \infty)$ words).

3.2.4 Similarity-Based Features

This set includes different similarity measures between `SRC`, `TGT`, `UE` and their back-translations. Borrowing concepts from cross-language information retrieval we estimate cosine similarities across languages by using character 3-grams (Mcnamee & Mayfield, 2004). From machine translation itself we consider models for parallel corpora alignment based on both pseudo-cognates and lengths (Simard, Foster, & Isabelle, 1992; Gale & Church, 1993; Pouliquen, Steinberger, & Ignat, 2003). These features intend to be an alternative to Levenshtein-based features in surface and back-translation sets.

3.2.5 Quality-Estimation-Based Features

We applied the 26 system-independent quality estimation (QE) measures provided by Asiya (Giménez & Màrquez, 2010) to the `SRC`–`TGT` and `SRC`–`UE` pairs. These measures intend to estimate translation quality without human references, making them appealing for our current framework. The QE measures are quite shallow, but they incorporate linguistic information at the level of part of speech, syntactic phrases, and named entities. A bilingual external dictionary is also used. Consequently, we can say that this set of features contains more linguistically-oriented generalizations than the previous ones. Perplexity, the number of out-of-vocabulary words of the translation sentence, as well as bilingual-dictionary-based similarity between `SRC` and `UE` (or `TGT`) are included among the 26 measures.

3.3 Classifier Learning and Intrinsic Evaluation

We trained support vector machines (SVM) with the previously described features to learn the classifiers. We used SVM$^{\text{light}}$ (Joachims, 1999) with linear, polynomial, and RBF kernels and we tuned the classifiers with 90% of the FFF^{+} corpus. The remaining 10% was left aside for testing purposes. Feature values were clipped to fit into the range $\mu \pm 3 * \sigma^2$ to decrease the impact of outliers. Normalization was then applied by means of z-score: $x = (x - \mu)/\sigma$. Later on, the mean and standard deviation of the tuning dataset were used to normalize the remaining test set instances.

3. All back-translations were produced by Spanish-to-English MT engines using the same Reverso.net technology.

Table 1: Cross-validation results for linear SVMs trained with incremental sets of features, with and without the application of feature selection. Best results are *italic faced*.

	all features				after feature selection			
feature sets	F_1	P	R	Acc	F_1	P	R	Acc
surface + bt	64.6	63.5	65.7	63.0	67.8	65.7	70.1	65.9
+ noise	70.1	63.7	78.0	65.9	73.5	67.0	81.5	69.9
+ similarity	69.3	64.3	75.2	65.9	73.6	68.1	79.9	70.5
+ QE	72.0	67.2	77.6	69.1	*76.1*	*71.0*	*81.9*	*73.5*

We evaluated on the basis of standard measures: *classification accuracy*, *precision* (ratio of predicted useful instances between all instances classified as useful), *recall* (ratio of predicted useful instances between all useful instances in the dataset), and F_1 (harmonic mean of precision and recall). Our training strategy aimed at optimizing F_1 and consisted of two iterative steps: (*a*) parameter tuning: a grid search for the most appropriate SVM parameters (Hsu, Chang, & Lin, 2003), and (*b*) feature selection: a wrapper strategy, implementing backward elimination to discard redundant or irrelevant features (Witten & Frank, 2005, p. 294). In short, this process performs an iterative search to remove the worst feature from the dataset at a time, according to the performance obtained with the classifier that neglects such feature. See further details in the work of Barrón-Cedeño et al. (2013).

We present here an incremental evaluation to see the contribution of every feature family and the effect of feature selection. The base feature set is composed of *surface* and *back-translation* (bt) features. Then, we incrementally add *noise*-based, *similarity*-based and *quality estimation* (QE) features. Table 1 presents the results. Figure 1 displays the corresponding precision–recall curves. The learning setting is restricted to linear SVMs in this experiment. A first observation is that the feature selection procedure consistently provides better accuracy and F_1 scores; i.e., it allows to discard irrelevant features and also some harming ones. Results show that all feature families contribute positively to the final performance of the classifiers, the gains are accumulative. This improvement is especially noticeable when quality estimation features come into play and feature filtering is applied. The numerical results are backed by the shape of the precision-recall curves: including all feature sets leads to better results, with precision levels clearly above 70% at recall levels of 60–70%.

A complementary study on using non-linear kernels for the task (not included here for brevity), revealed that even though slightly better accuracy and F_1 results can be obtained by using non-linear kernels, the shape of the precision–recall curve is better for the linear classifier in the high-precision zone.[4] Avoiding false positives is a very important property when thinking of selecting useful user edits for MT improvement. Therefore, we used linear classifiers in all the translation experiments in Section 5. The extended study, including kernel variants, is available at the description in the report of FAUST (2013, Section 3.4). This report also includes more detailed experiments on the relevance of individual features

4. For values above 0.6 precision, the curve for the linear classifier is smoother and contains a much larger area below it.

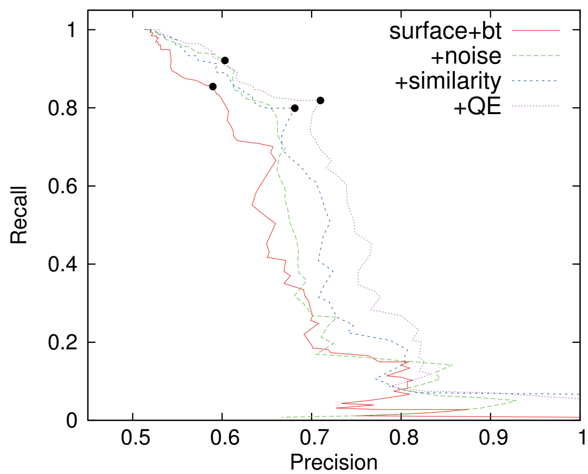


Figure 1: Precision-Recall curves for different learning settings and feature sets over the development partition. Black dots represent the optimal F_1 values.

and a comparison of the example rankings produced by different classifier variants on the set of examples.

Finally, we evaluated the different classifiers on the 10% test partition from the FFF^+ corpus. Absolute resuts are slightly lower: the linear SVM with all feature sets obtains F_1 and accuracy values of 73.2 and 69.9, respectively,[5] but the same results are observed. That is, precision–recall curves for linear SVMs including all feature sets are better than the rest, allowing to obtain higher precision scores at similar levels of recall.

3.4 Discussion

The classifiers analyzed in this section showed modest levels of precision, only slightly over 70% at acceptable levels of recall. Significantly higher precision values can be reached only at the price of lowering recall below 10%. This behavior reflects the difficulty of confidently characterizing positive examples with the type of features used to train the classifiers. It is worth noting that the task is also difficult for humans. The agreement achieved between annotators (Cohen's kappa below 0.6) can be considered only moderately high, but certainly not fully satisfactory, evincing the inherent difficulty of the task. Translation quality is a multi-faceted concept, which encompasses *adequacy* (i.e., whether the translation conveys the meaning of the source), *fluency* (i.e., whether the translation is a fluent utterance in the target language) and many other aspects; some of them defined on an application basis (e.g., vocabulary usage, language register, post-editing effort, etc.). As a result, human perception of translation quality is a highly subjective matter, depending on small details, which are difficult to capture and delimit in a set of simple annotation guidelines. This effect is amplified in our corpus by the noisy nature of the input text, where the input sentences

5. The FFF^+ test set is very small and susceptible of statistical unstability when computing performance scores.

often lack the necessary context to make fully reliable decisions. Fortunately, classifying good user edits is not the end task. The ultimate goal of the UE classifiers is to perform selection, i.e., to rank UE instances by acceptability and set an appropriate threshold to select useful UEs to improve the SMT engine. In the following sections we empirically show that, regardless of the limited performance of the local classifiers, the proper inclusion of the selected highest-ranked user-edited translations into a general purpose SMT can significantly improve its quality. Finally, one might argue that rather than training classifiers to optimize accuracy on the local task, it would be better to define a joint "UE-selection"—"MT-enrichment" learning setting, where the classifiers were optimized directly against the translation quality of the enriched SMT system. Although this is attractive in theory, it would be extremely inefficient and practically infeasible in our pipeline.

4. A Method for Incorporating User-Edits into the SMT System

In this section we describe a method for incorporating user-edits into the machine translation model. Our approach is developed under the assumption that the translation model and the user-edited materials are the only data at our disposal to improve the translation system. Our approach is divided in two steps: (*i*) using the original automatic translation as a pivot to align the source text to the edited translation and extracting new phrase pairs (Section 4.1), and (*ii*) the interpolation of the new phrase pairs with the original translation model (Section 4.2). Some validation experiments are discussed in Section 4.3.

4.1 Pivot-Based Word Alignment

In order to detect and correct translation errors, we consider three pieces of information: the source input text `SRC`, the target output translation given by the translation system `TGT`, and its user-edited version `UE`. A monolingual alignment between `TGT` and `UE` allows for predicting translation errors, by identifying the parts that have been edited. The projection of this alignment to `SRC` allows for the extraction of new translation pairs, representing the corrections provided in `UE`.

We propose a three-step alignment procedure. First, we compute the TER path (Snover, Madnani, Dorr, & Schwartz, 2009) between `TGT` and `UE`, aligning the identical words from both sides. Second, we estimate the best alignment among the possible combinations of unaligned `TGT` and `UE` words by maximizing a similarity function between pairs of words in `TGT` and `UE` —$\text{sim}(w_t, w_u)$. Finally, once the monolingual alignment is made, we pivot the alignment towards `SRC`: taking advantage of the decoder-provided word alignment between `SRC` and `TGT`, we link words between `SRC` and `UE` if and only if there is a word in `TGT` that connects them. This alignment process, which we call as `SimTer` is described in Algorithm 1.

The comparison between the translated and edited sentences is based on translation edit rate, TER (Snover et al., 2009).[6] In addition to the minimum number of edits, TER obtains an alignment and edit path: the required sequence of edits that change the output translation into the reference —the user-edited sentence in our case. Figure 2 shows an

6. TER is an error metric that estimates the number of edits required to convert a translation output into one of its references. Although based on the Levenshtein distance, TER additionally allows word movements before considering the usual addition, deletion, and substitution operations in order to reduce the number of changes.

Algorithm 1 `SimTer`. A pivot-based algorithm to align `SRC` and `UE` through `TGT`

1: Translate `SRC` into `TGT` with the decoder and obtain the corresponding word alignments $\text{align}(w_s, w_t)$ for every $w_s \in$ `SRC` and $w_t \in$ `TGT`;
2: Compute the TER path between `TGT` and `UE` and align identical words (cf. Section 4.1);
3: Align every non-aligned word $w_t \in$ `TGT` to words $w_u \in$ `UE` such that the summation of the similarities of all new pairs, $\text{Sim}(w_t, w_u)$, is maximized;
4: For every pair of alignments $\text{al}(w_s, w_t)$ and $\text{al}(w_t, w_u)$, create a new alignment $\text{al}(w_s, w_u)$;

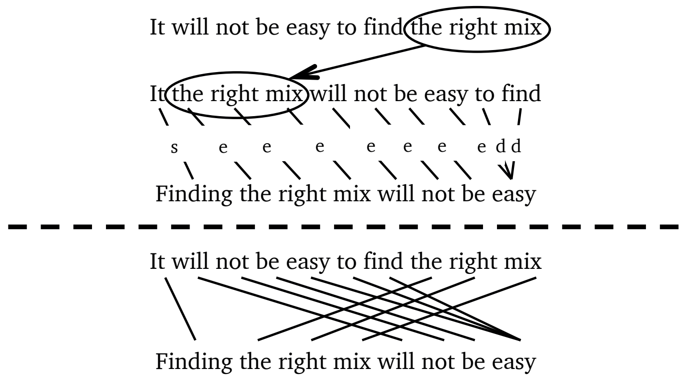


Figure 2: TER-path computed for a monolingual sentence pair. 'd', 'e' and 's' stand for deletion, exact (no change), and substitution. The minimum number of edits is three: two deletions and one substitution.

example. First, in the upper part of the figure, TER allows word movements, so *the right mix* is moved next to the first word. The minimum number of edits is then computed, resulting in one substitution and two deletions. Finally, in the lower part of the figure, the word movements roll back to their original positions and we obtain a word alignment between the two sentences.

Although TER is able to align most of the words correctly, it may fail in some circumstances such as when the words affected by an edit are actually aligned according to their position in the sentence, rather than their semantic similarity. In the example of Figure 2, *finding* and *to find* should be aligned. As a counter measure to this issue, we propose to consider the similarity between w_t and w_u as a linear combination of simple similarity features:

$$\text{sim}(w_t, w_u) = \sum_{i=1}^{8} \beta_i h_i(w_t, w_u)\,, \tag{1}$$

where $h_i(w_t, w_u)$ are the similarity features and β_i are the corresponding contribution weights. Each h_i models a different relationship between w_t and w_u as follows:

h_1 - A binary feature that indicates if w_t and w_u are identical.

$h_{2,3}$ - Two binary features accounting for the existence of links between w_{t-1} and w_{u-1} (h_2) or w_{t+1} and w_{u+1} (h_3).

h_4 - A feature to penalize multiple links (one–to–many). It considers the possibility of link(w'_t, w_u) aligning a user-edited word w_u to the TGT word w_t when a link between w'_t and w_u has already been set. The feature penalizes this new link proportionally to the distance between w_t and w'_t:

$$h_4(w_t, w_u) = - \max_{\forall \text{link}(w'_t, w_u)} \left(\frac{\|\text{pos}(w_t) - \text{pos}(w'_t)\|}{|\texttt{TGT}|_t} \right) , \quad (2)$$

where pos$(\cdot)$ is the position of $\cdot$ in TGT and $|\texttt{TGT}|_t$ is the number of tokens in TGT.

$h_{5,6}$ - Two lexical features designed to evaluate the strength of the semantic relationship between w_t and w_u according to their proximity to w_s. This is done by considering the bidirectional conditional probabilities between both (w_s, w_t) and (w_s, w_u) in the translation table. Feature $h_5(w_t, w_u)$ is approximated as a normalized conditional probability based on bilingual dictionaries:

$$\begin{aligned} h_5(w_t, w_u) &= \sum_{s:link(w_s,w_t)} \frac{p(w_t \mid w_s) p(w_s \mid w_u)}{\sum_{w'_u \in \texttt{UE}} p(w_t \mid w_s) p(w_s \mid w'_u)} & (3) \\ &= \sum_{s:link(w_s,w_t)} \frac{p(w_s \mid w_u)}{\sum_{w'_u \in \texttt{UE}} p(w_s \mid w'_u)} , & (4) \end{aligned}$$

where a normalization factor is included in order to consider the contribution of $p(w_s \mid w_u)$ only in the context of sentence UE. Feature h_6 is analogous to h_5, but considering $p(w_s \mid w_t)$ and $p(w_u \mid w_s)$ instead. If w_s is an unknown word (i.e., it is replicated from SRC to TGT without mapping in both bilingual dictionaries), we take $h_5 = h_6 = 0$.

h_7 - This feature is applied only when w_t is an unknown word that has been duplicated by the translation system from the input sentence into the output. If w_u and w_t are the same, we force them to be linked by giving h_7 an arbitrarily large value. Otherwise, the feature takes a real value as a function of the Levenshtein distance (LD) at character level between the unknown w_t and w_u:

$$h_7(w_t, w_u) = 1 - \frac{LD(w_t, w_u)}{|w_u|} , \quad (5)$$

where $|w_u|$ represents the length of w_u in characters. This feature becomes a penalty when the Levensthein distance exceeds the length of w_u, preventing its linking to longer unrelated words.

h_8 - A penalty feature to prevent alignments between an unknown word w_t and a stopword w_u. It takes a large negative value if w_u is a stopword; zero otherwise. We take as stopwords determiners, articles, pronouns, prepositions, and auxilary verbs.

The β weights relative to each h_i feature are obtained through a downhill simplex algorithm (Nelder & Mead, 1965). We give more details in Section 4.3.2.

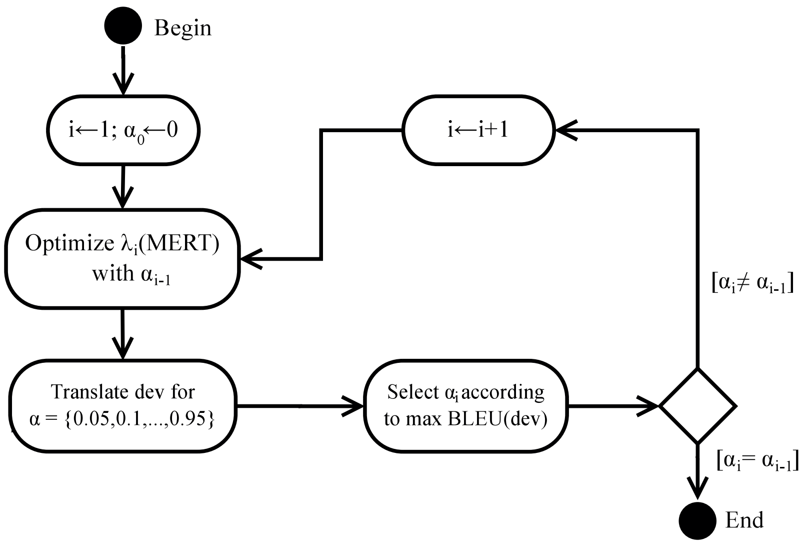


Figure 3: Process to compute the interpolation weight (α) and re-tune the TM feature coefficients (λ).

4.2 Incorporation of the Aligned Phrase Pairs

After computing the `SimTer` word alignment between source and edited translations, we use this new parallel corpus (`SRC`,`UE`) to enrich and retrain the translator. We have to deal with two types of newly extracted phrase pairs (translation units): (*i*) *previously–seen phrase pairs* are those already included in the original translation model; they are relatively up-weighted within the translation models, favoring their selection when facing a similar situation, and (*ii*) *new phrase pairs* which are those missing in the original translation model and represent the principal resource to improve the translator. New phrase pairs must be added to the final translation model in order to provide the decoder with new translation options when facing similar source sentences.

We include the new phrase pairs into the original model using Foster and Kuhn's (2007) interpolation method, initially designed to address domain adaptation problems. The relatively small translation models extracted from the user edits are estimated by means of symmetrization and phrase-extraction standard algorithms with *grow-diag-final-and* heuristic. Then, the original and the new translation models are linearly interpolated according to:

$$\mathrm{TM}(\mathrm{phrase}_i) = \alpha \ \mathrm{TM}_{\mathrm{original}}(\mathrm{phrase}_i) + (1-\alpha) \ \mathrm{TM}_{\mathrm{pe}}(\mathrm{phrase}_i) \quad (6)$$

where TM, $\mathrm{TM}_{\mathrm{original}}$ and $\mathrm{TM}_{\mathrm{pe}}$ are the final, original, and `UE`-based translation model scores for a phrase_i pair. The setting of the interpolation parameter α is strongly coupled to the re-tuning of the *classical* set of weights (λ) used to combine the SMT features. We applied the iterative two-step process outlined in Figure 3. The process starts with the set of weights λ_1 from the original translation system and iterates while the updates to the α weight yield BLEU improvements. This is a two-step iterative process. First, the best λ_i

Table 2: Statistics of the English–Spanish corpora used to obtain the SimTer similarity function weights β.

Corpus		Sent.	Words	Vocab.	avg.len.
EPPS	Eng	1.90 M	49.40 M	124.03 k	26.05
	Spa		52.66 M	154.67 k	27.28
EPPS_UE	Eng	100	2,862	1,017	28.6
	Spa		3,022	1,089	30.2

weighting is computed using the best α_{i-1} available. In the case of $i = 1$ no interpolation is taken into account ($\alpha_0 = 0$). Afterwards, the optimum α_i interpolation parameter is computed using the best set of translation feature weights λ_i obtained.

The combination procedure is simpler for the reordering models. It follows a fill-up strategy that preserves every entry and score from the original model and adds new entries and scores only for the new phrases (Bisazza et al., 2011). Units that appear both in the original model and the one obtained from user-edits preserve the reordering score of the original model.

4.3 Validation Experiments

The objective of the experiments described in this section is twofold: (*i*) tuning the meta-parameters for our algorithms and (*ii*) validating the proposed methodology in a well established domain-adaptation task. We consider these experiments preliminary, as translation references are used instead of proper user edits.

4.3.1 Datasets

We selected different datasets for these experiments. In order to optimize the β parameters of the similarity function in Equation (1), we used the Europarl v6 corpus, EPPS (Koehn, 2005), to build a base phrase-based SMT system. To evaluate the alignments, a small manually-aligned corpus, EPPS_UE (Lambert, de Gispert, Banchs, & Mariño, 2005), was used to perform the pivot-translations and subsequent SimTer alignments. This small corpus belongs to the same domain as EPPS but there is no intersection between them. Table 2 shows some statistics.

In order to tune the α and λ parameters, and to validate the proposed methodology, we used the corpora from the WMT'12 campaign (Callison-Burch, Koehn, Monz, Post, Soricut, & Specia, 2012). It contains parallel sentences from the EPPS corpus already mentioned, News Commentary (NC), and United Nations (UN). It also includes a monolingual version of Europarl and monolingual corpora based on News (broken down by years: 2007-2011). Tables 3 and 4 provide descriptive statistics of these datasets, computed after cleaning and pre-processing. Additionally, we used the WMT'08-11 test material for tuning the α and the TM's λs (dev), and WMT'12/13 tests for testing the methodology (test12 and test13). Table 5 shows the statistics for the tuning/test material.

Table 3: Statistics of the additional WMT'12 English–Spanish parallel corpora for training the translation models (used for tuning and validation purposes).

Corpus		**Sent.**	**Words**	**Vocab.**	**avg.len.**
Preliminary Experiments					
NC	Eng	0.15 M	3.73 M	62.70 k	24.20
	Spa		4.33 M	73.97 k	28.09
UN	Eng	8.38 M	205.68 M	575.04 k	24.54
	Spa		239.40 M	598.54 k	28.56

Table 4: Statistics of the Spanish monolingual corpora used to build the language models.

Corpus	**Sent.**	**Words**	**Vocab.**	**avg.len.**
EPPS	2.12 M	61.97 M	174.92 k	29.18
NC	0.18 M	5.24 M	81.56 k	28.55
UN	11.20 M	372.21 M	725.73 k	33.24
News.07	0.05 M	1.33 M	64.10 k	28.91
News.08	1.71 M	49.97 M	377.56 k	19.19
News.09	1.07 M	30.57 M	287.81 k	28.63
News.10	0.69 M	19.58 M	226.76 k	28.54
News.11	5.11 M	151.06 M	668.63 k	29.55

Table 5: Statistics of the development and test corpora used to tune and test the translation system.

Corpus		**Sent.**	**Words**	**Vocab.**	**avg.len.**
WMT based dev/test					
dev	Eng	7,567	189.01 k	18.61 k	25.0
	Spa		202.80 k	21.75 k	26.8
test12	Eng	3,003	63.78 k	14.34 k	21.2
	Spa		69.45 k	16.47 k	23.1
test13	Eng	3,000	56.09 k	13.34 k	18.7
	Spa		62.05 k	15.16 k	20.7

4.3.2 Tuning the Parameters of the Similarity Function

We built the baseline SMT system following the standard pipeline of a Moses training with EPPS (Koehn & Hoang, 2007). We applied the resulting system to translate the source side of the manually-aligned corpus (EPPS_UE). Then, we carried out a downhill simplex process to adjust the β weights (Nelder & Mead, 1965), except for β_8 that was fixed to 1. Recall that h_8 assigns large costs to prevent alignments between unknown words and function words. When activated, it works a as hard constraint, pruning the hypotheses space, so the value of β_8 can be chosen arbitrarily to any positive number different from zero. `SimTer` was applied at each of the three completed iterations. Our final goal is to produce an alignment between the source sentence and its edited translation. Therefore, we evaluated using Alignment Error Rate (AER) with respect to the manual source–reference

Table 6: Contribution weights of the similarity function features.

β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8
0.08	0.75	0.91	3.08	0.47	2.02	1.50	1.00

Table 7: Translation quality for different values of α.

α	0.1	0.2	0.3	0.4	0.5	**0.6**	0.7	0.8	0.9
BLEU	27.86	28.18	28.33	28.49	28.68	**28.75**	28.69	28.62	28.45

alignment (Och & Ney, 2003). AER was reduced from 20.12% to 17.57% (13% relative error reduction), reaching a performance equivalent to that of mGIZA++ with the same corpora. Table 6 includes the resulting βs. As the β_4 value shows, the penalization for distant links (h_4) is the most important feature. As for the lexical features, h_6 is significantly more relevant than h_5. Interestingly, the feature h_1 (same word) was not considered relevant compared to the others.

4.3.3 Tuning and Validating the Combination Method

The last step before fully testing our strategy is to compute the α parameter from equation (6). As mentioned before, we used the corpus from the WMT'12 campaign (Callison-Burch et al., 2012). We trained a baseline English_to_Spanish system as a factored Moses phrase-based system (Koehn & Hoang, 2007) from words into words and POS tags (Formiga et al., 2012).[7] The base system considered EPPS and UN concatenated as a whole corpus. Regarding the monolingual data, a language model (LM) was built for each corpus and then interpolated by minimizing the perplexity on the development set (Schwenk & Koehn, 2008). In this experiment we translated the English sentences of the NC parallel corpus and took the Spanish references to simulate user edited translations (UE). We performed a `SimTer` word alignment to build UE-specific translation and reordering models. Finally, we applied the α optimization method depicted in Section 4.2.

Table 7 shows the BLEU scores obtained with different values of α. When combining the translation models, the BLEU improved from 27.86 to 28.75, achieving its highest value with $\alpha = 0.6$ (i.e., a 60–40% distribution of the weight for the base and edited translation models, respectively). After setting β and α, we validated our adaptation method with the obtained hyper-parameters. We also compared our combination method (referred to as '*Linear Interpolation* ($\alpha = 0.6$)') to other methods available in Moses, namely:

Concatenation NC, EPPS, and UN are aggregated as a single training corpus.

Multiple Tables *Either* Two parallel decodings corresponding to each TM are launched separately, selecting the one with the best score.

Multiple Tables *Both* One decoding is launched considering all the options available in both phrase tables, doubling the number of translation features in the log-linear model.

7. The text was POS-tagged with the Freeling suite of NLP analyzers (Padró, Collado, Reese, Lloberes, & Castellón, 2010).

Moses Incremental training The base phrase-table is updated with approximate alignments (see Levenberg et al., 2010 in Section 2) . The alignments are computed with the Moses incremental `inc-giza-pp` algorithm instead of the `SimTer` algorithm.

Table 8: Results with different combinations of base and `SimTer`-specific translation models. In BLEU and NIST columns, '†' and '‡' indicate significant differences over the 'Concatenation' system with 0.95 and 0.90 confidence levels, respectively. Best absolute results are depicted in bold face. Moses incremental training is shown for comparison purposes although it does not use `SimTer` alignment

	BLEU	**NIST**	**TER**	**METEOR**	**ULC**
test12					
Base	32.77	8.63	48.65	55.48	70.78
Concatenation	33.03	8.66	48.48	55.64	71.16
Linear Interpolation ($\alpha = 0.6$)	**33.25**†	**8.70**†	48.24	**55.84**	**71.66**
Multiple tables *either*	33.20‡	**8.70**†	**48.19**	55.76	71.61
Multiple tables *both*	31.72	8.51	49.42	54.88	69.21
Moses Incremental training	32.58	8.61	48.86	55.40	71.04
test13					
Base	28.74	8.01	51.60	52.82	70.94
Concatenation	28.96	**8.07**	51.29	53.11	71.56
Linear Interpolation ($\alpha = 0.6$)	**29.15**†	**8.07**	**51.28**	**53.15**	**71.73**
Multiple tables *either*	29.13‡	**8.07**	51.29	53.03	71.66
Multiple tables *both*	28.02	7.91	52.40	52.26	69.52
Moses Incremental training	28.57	7.97	52.01	52.67	70.74

We used BLEU, NIST, TER and METEOR (Papineni, Roukos, Ward, & Zhu, 2002; NIST, 2002; Snover et al., 2009; Denkowski & Lavie, 2011) as automatic translation quality measures. Additionally, we considered a linear combination of the former ones: ULC (Giménez & Màrquez, 2010). Table 8 presents the obtained results. The adaptation strategy outperforms Baseline and Concatenation configurations practically for every test corpora and quality measure. More precisely, we found significant differences under BLEU (test12 and test13) and NIST (test12) metrics.[8] The performance of the *either* combination was very close to our linear interpolation method. However, the *either* combination is computationally expensive, almost doubling the time required by the linear interpolation method. Table 9 includes the translation times for reference.[9] The *incremental* approach is the fastest one, but it yields no improvement over the baseline.

In summary, the alignment and combination methods proposed in this work offer significantly better results without sacrificing computational efficiency, compared to the other alternative methods provided in Moses. Therefore, we do not consider multiple decoding and incremental strategies in the remaining experiments of the paper.

8. In this work, significances are computed through paired bootstrap resampling (Riezler & Maxwell, 2005)
9. These figures were computed on a Linux server with 96 GB of RAM and 24-core CPU Xeon processors 1.6 GHz (134064 Bogomips in total). Multi-threading was not used to compute the decoding times.

Table 9: Translation times in seconds (Collecting+Decoding) for each combination method.

Combination method	Num. sent.	Total time		Time per Sentence	
		test12	test13	test12	test13
Moses Incremental training	3,003	7,502.52	6,553.45	2.50	2.18
Linear Interpolation ($\alpha = 0.6$)		8,284.79	7,684.65	2.76	2.56
Multiple tables *both*		13,353.80	12,291.70	4.10	4.45
Multiple tables *either*		13,097.40	11,364.80	4.36	3.79

Table 10: Statistics of the English–Spanish parallel corpora used in the FAUST scenario.

Corpus		Sent.	Words	Vocab.	avg. length
FAUST UE	Eng	6,610	43,310	8,250	6.6
	Spa		47,800	10,430	7.2
FAUST_dev Clean	Eng	1,998	24,588	3,758	12.3
	Spa_ref0		24,588	3,758	12.3
	Spa_ref1		25,270	3,743	12.6
FAUST_test Raw	Eng	998	9,941	4,184	10.0
	Spa_ref0		10,135	4,484	10.2
	Spa_ref1		10,333	4,499	10.4
FAUST_test Clean	Eng	1,996	19,773	4,737	9.9
	Spa_ref0		20,270	4,484	10.2
	Spa_ref1		20,666	4,499	10.4
FAUST Monolingual	Spa	98,199	1.15 M	89,378	11.67

5. Experiments with Real Data

In this section we present experiments on using our methodology to improve already existing MT systems with real data. We describe two experiments. In the first scenario, the new material comes from a collection of user-edited translations submitted to the Reverso.net MT Web service (cf. Section 5.1). In the second scenario, new material is selected (cf. Section 5.2) from *CommonCrawl* (Smith et al., 2013)

5.1 User-Provided Edited Translations (FAUST)

In the the FAUST project (cf. Section 3) the goal was to improve the quality of on-line MT services by leveraging users feedback, mainly in the form of suggested improved translations. In these experiments, we take advantage of parallel and monolingual data supplied by a set of user translation queries and their edits. These users belong to an on-line community motivated to edit the response to their translation queries.

The FAUST parallel corpora are composed of two non-overlapping collections of translation requests gathered from the Reverso.net website: FAUST UE and FAUST dev/test.[10] FAUST UE includes triplets composed of input source, MT output, and the user edit. We use this corpus for training purposes. The FAUST dev/test corpus includes target refer-

10. A sample of FAUST UE is available at `ftp://mi.eng.cam.ac.uk/data/faust/FaustFeedbackSample.xls.gz` under "FAUST User-edited corpus". FAUST dev/test are available at `ftp://mi.eng.cam.ac.uk/data/faust/FAUST-1.0.tgz`.

ences provided by two professional translators. Moreover, the translators processed the source inputs to reduce noise (e.g., removing slang words, misspellings, smileys, etc.). This process resulted in two versions of the dev/test corpus: *Raw*, where the source inputs are the original ones, and *Clean*. In our experiments we considered the 'FAUST_dev Clean' version for tuning (less error prone), and the real 'FAUST_test Raw' for testing. The FAUST monolingual corpus is composed of 98,199 translation requests to the Reverso.net website that had Spanish as the source language. No selection with respect to the target language was made. Table 10 shows some statistics of the FAUST corpora.

For our first experiment we took the *Concatenation* MT system from Section 4.3.3 and adapted its target language model to the FAUST scenario as follows: (*i*) we built a specific web-domain language model from the FAUST Monolingual corpus, (*ii*) we obtained a language model by means of a new interpolation of all language models according to perplexity minimization on the FAUST_dev Clean corpus, and (*iii*) we tuned the weights of the translation features using MERT again to maximize BLEU on the FAUST_dev Clean corpus. Our goal was to set a strong baseline system for the experimental comparison on this datset. We will refer to it as Base_FAUST.

In order to select the most suitable feedback material to improve the Base_FAUST system, we ranked the FAUST UE collection of 6,610 user-edited instances according to the SVM classifier scores (cf. Section 3). The SVM labeled 61% of the data as useful feedback (we call that point TH0, for 'decision threshold=0'). However, there is no guarantee that this level of selection maximizes the translation quality of the adapted system. Therefore, we carried out an analysis of the quality as a function of the percentage of selected user-edited instances.

5.1.1 Results

Figure 4 depicts the performance obtained on the FAUST_test *Raw* corpus with different percentages of selected user-edited data. This figure focuses on two evaluation metrics: NIST, which is based on a classical n-gram matching approach with an improved brevity penalty providing more robustness to noise than BLEU and TER, which tries to mimic the editing effort that would be addressed by humans in order to obtain a high-quality translation. Very similar curves are observed when using FAUST_test *Clean* (not shown for brevity). Table 11 presents a more complete comparative results on the FAUST_Raw corpus, with the set of extended evaluation metrics from Section 4.3.3: BLEU, NIST, METEOR, TER, and ULC. Different adaptation and filtering strategies are also presented in the table, including: (*i*) different filtering methods (FFF+ and Subsampling), (*ii*) adaptation methods (Concatenation vs. `SimTer`), and (*iii*) different percentages of included user-edits: 50%, 61%(TH0), and 100%. Significances are computed in the same way as described in Section 4.3.3. We only performed subsampling by computing the perplexity between the existing models and the `UE` part, as we wanted to select the best user edits.

When analyzing the results, it is worth noting that we already have a strong baseline, tuned *in-domain* by using domain specific monolingual data. Several observations can be drawn. The first block (SimTer_0.6 & Subsamp.) shows that adding new material by subsampling filtering provides none or little improvement to the baseline depending on the evaluation metric under analysis. More precisely n-gram based metrics BLEU and NIST

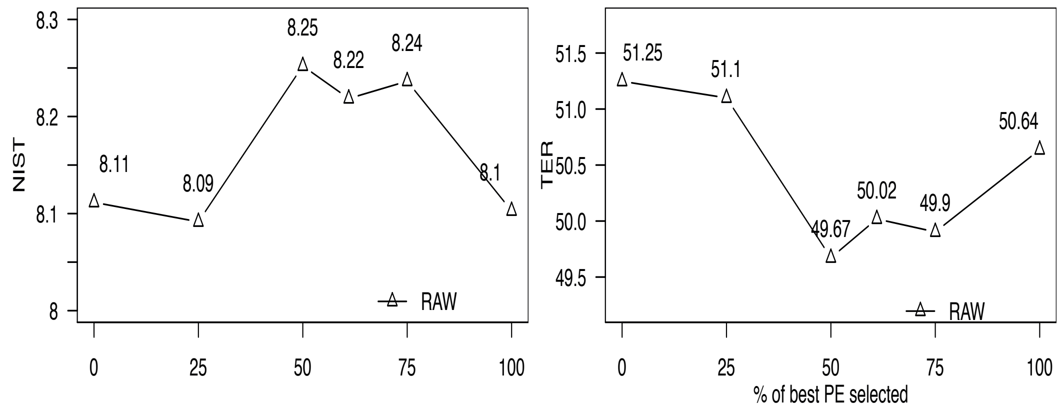


Figure 4: NIST and TER scores in function of the percentage of best ranked user-edits used.

Table 11: Results obtained before and after considering the feedback instances depending on the amount of user-edits used and the different filtering and adaptation methods. '$*$' and '†' indicate significant differences over the Baseline system with 0.99 and 0.95 confidence levels. Best absolute results are highlighted.

Translation system		**% of edits**	**BLEU**	**NIST**	**TER**	**METEOR**	**ULC**
Adaptation	**Filtering**						
Baseline	—	0%	33.34	8.11	51.25	55.10	71.05
SimTer_0.6	Subsamp.	+25%	33.26	8.11	51.27	55.06	70.96
		+50%	33.41	8.16	50.48	55.55	71.85
		+75%	32.95	8.13	50.58	55.32	71.39
	FFF+	+50%	**34.01**†	**8.25**$*$	**49.67**	56.06	**73.23**
		+TH0-61%	33.68	8.22$*$	50.02	**56.14**	72.78
	—	100%	33.18	8.10	50.64	55.54	71.68
Concatenation	FFF+	+50%	32.89	8.15	50.29	55.79	71.83
		+TH0-61%	33.13	8.08	50.63	55.46	71.44

do not capture any improvement as TER and METEOR slightly do. The most important evidence is provided by the second block (SimTer_0.6 & FFF+), which is the strategy we propose in this paper. The results evince the appropriateness of the FFF+ filtering as it yields significantly better results in all metrics compared to Subsampling. However, it is remarkable that the FFF+ and Subsampling learning curves obtain the best results with only the 50% of the total user-edits considered. The filtering strategy results crucial in obtaining a final improvement. Not only regarding the method, but also because using all the edits with no filtering (+100%) has no impact on the n-gram-based metrics (BLEU and NIST), and only marginally improves with the other metrics (TER and METEOR). The last block (Concatenation & FFF+) confirms the important contribution of the SimTer_0.6 adaptation strategy compared to the straightforward approach of adding the new material by concatenation. In this case, using exactly the same filtered material, SimTer_0.6 yields better results compared to the concatenation strategy.

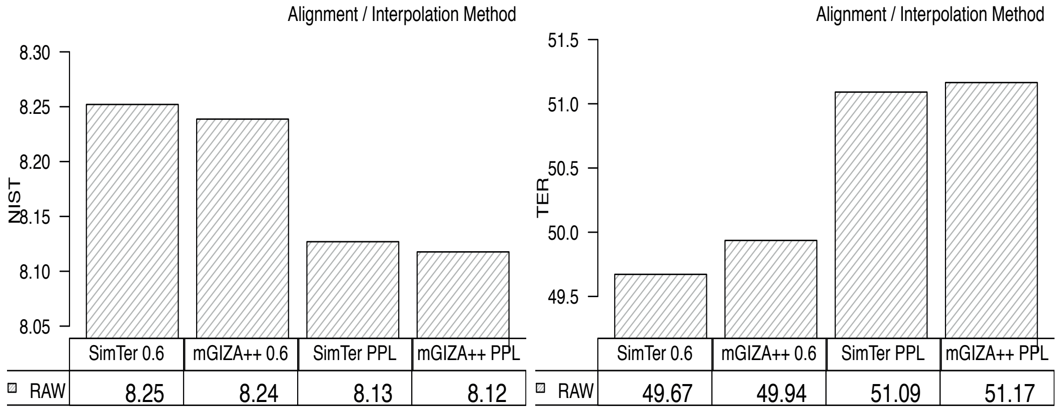


Figure 5: NIST and TER translation performance over the FAUST *Raw* and *Clean* tests achieved with different alignment/phrase-table interpolation methods.

In short, the UE-enriched translation models following our SimTer_0.6 & FFF+ yield a significant final improvement of (+0.67) BLEU points and (-1.57) TER points on the test set. Introducing user edits without any pre-selection nor boosting scheme does not allow the MT system to achieve consistent improvements.

Additionally, we compared our alignment/phrase-table interpolation approach to other competitive variants present in the literature. Concerning the alignment, we considered the forced alignment capability of `mGIZA++` as defined by Gao and Vogel (2008): the *m*-alignments are obtained from a re-trained IBM 4 model that was iterated with all the data —the original training and the edited material. Regarding the weight-interpolation strategy, we considered the perplexity minimization method (PPL) of Sennrich (2012), especially suited for domain adaptation. In this approach, a development translation model (TM) (i.e., phrase-table) was built from a small development set. This development TM is used to minimize the perplexity when combining different TMs —in our case base and UE-based models. The strength of this approach is the granularity of the weights: instead of giving a different weight to each phrase-table, it assigns different weights to each feature function within the phrase-table, trying to lower the perplexity as much as possible. We used the optimization method L-BFGS with numerically approximated gradients (Byrd et al., 1995).

Figure 5 presents the results obtained in the four alignment/interpolation combined scenarios. Concerning the comparison of alignment methods, NIST shows no significant differences between the `SimTer` pivot-based and the `mGIZA++` approaches. However, TER reflects a bigger difference in favor of `SimTer`. This behavior is coherent with our alignment strategy, focused on finding the particular edits given by the user. In terms of computational time, there are no big differences: both alignments are computed in less than a minute for 6,610 sentences. One of the advantages of using `SimTer` is that it does not require nor depend on previous alignment models (`mGIZA++` or alike). [11] Comparing them in terms of alignment error rate is beyond the scope of this paper. `SimTer` is specifically tailored to find the differences between the original and the edited translations in order to extract

11. We are not claiming that `SimTer` can work well as a general purpose alignment algorithm, being competitive to `mGIZA++` or other state-of-the-art aligners.

Table 12: Real examples of translations changed (but not necessarily improved) by the SimTer_0.6 & FFF+ (50%) system and their categorization according to the linguistic phenomena studied

	Source	**Baseline**	**SimTer_0.6 & FFF+ (50%)**
Function Words	Applicant's authorized representative information:	*El* representante autorizado solicitante información:	Representante autorizado **del** solicitante **de la** información:
Additions/ Omissions (worse)	Tell me how I'm supposed to breathe with no air.	DIME cómo *se* supone *que tengo* para respirar sin aire.	DIME cómo **me** supone para respirar sin aire.
Lexical	Do you use all the letters in the English alphabet to write in Persian?	¿utilizar todas las letras del alfabeto *persa* para escribir en *español?*	¿utilizar todas las letras del alfabeto **inglés** para escribir en **persa?**
Reorder.	These measures were updated and developed in the 2005 strategy review.	Estas medidas *fueron actualizados* y la revisión de la estrategia *desarrollada en el* 2005.	Estas medidas **se han actualizado** y **desarrollado en** la revisión de la estrategia **de** 2005.
Bad Feedback	Did it?	*¿verdad?*	**Hiza intentoHázlo?**
Morphol.	If an alien comes (. . .), will you leave me for him?	Si un extranjero llega (. . .), *me dejan* por él?	Si un extranjero llega (. . .), **vas a dejarme** por él?
Combined	Please be informed that we will be provide you 5 years warranty on material and workmanship.	Informamos que vamos a *darle garantía de* 5 años sobre *los* materiales y *la* mano de obra.	Informamos que vamos a **proporcionarle** 5 años **de garantía** sobre materiales y mano de obra.

new phrase pairs, useful to improve the translation models. Thus, the two tools serve a different purpose.[12] Finally, regarding the interpolation strategy, setting the interpolation weights with the perplexity minimization method from Sennrich (2012) does not provide enough boosting to the UE-based models. This issue is particularly observed when looking at the weights set by the L-BFGS algorithm: they are in the order of ≈ 0.99 for the baseline model and $\approx 1 \cdot 10^{-3}$ for the UE-based model. On the contrary, an optimization based on the quality of the provided translation addresses the point directly.

5.1.2 Qualitative Output Analysis

The results presented so far are based on automatic evaluation metrics. We now complement the study with a set of human assessments in order to verify if the improvement is also perceived by humans and to identify the characteristics that make the new translations better. Five expert annotators analyzed 414 instances from the 'FAUST_test Clean' corpus. The annotators observed triplets composed of source sentence, the translation produced by the Baseline, and the translation of the better performing SimTer_0.6 & FFF+ (50%) system. They had to determine which of the two translations was better or, instead, if they had the same quality. An additional option "I cannot tell" was possible as well. Annotators

12. For the sake of completeness, we can mention that some experiments conducted to evaluate the performance of SimTer as a general aligner showed AER results significantly lower than those of `mGIZA++` (Henriquez, 2014)

Table 13: Results of the comparative analysis carried out by five human annotators on the translation of 414 sentences of the 'FAUST_test Clean' corpus by the Baseline and SimTer_0.6 & FFF+ (50%) systems. For each criterion (row), *Better* indicates that user-adapted models provide a better translation compared to the non-adapted system, *Worse* indicates it conversely and *Changed* indicates that the translations are different.

	Better	**Same**	**Worse**	**Cannot Tell**
Adequacy	34.54%	40.58%	15.70%	9.18%
Fluency	32.61%	49.76%	17.63%	–
	Better	**Same**	**Worse**	**Changed**
Function Words	50.00%	18.56%	31.52%	13.04%
Additions / Omissions	31.54%	20.53%	47.93%	17.63%
Lexical	47.59%	31.99%	20.40%	**60.39%**
Reordering	57.50%	18.37%	24.13%	21.01%
Bad Feedback	–	–	100.00%	5.31%
Morphology	35.77%	40.73%	23.45%	19.57%

Table 14: Progression of OOV ratio and BLEU for the 'FAUST_test Clean' corpus along with the different acceptance levels of user-edited instances.

Selection Ratio	**OOV Words**	**Total Words**	**OOV Ratio**	**BLEU**
0% (no-feedback)	563	22,898	2.46%	37.85
25%	559		2.44%	37.86
50%	557		2.43%	**38.65**
61%	554		2.42%	38.32
75%	550		2.40%	38.47
100% (all-feedback)	550		2.40%	37.79

did not know which system produced which translation, and the order of presentation of the two options was randomized. The overall quality assessment was based on translation adequacy and fluency, but annotators were also asked to provide detailed information on which linguistic aspects made one translation better than the other one; e.g., changes on function words, addition or omission of spurious words, lexical coverage, reordering, morphology, and presence of harmful elements (*bad feedback*, i.e., mistranslations clearly introduced by erroneous user edits). Table 12 includes real samples of the studied phenomena. Cohen's kappa agreement for all the annotators on a selection of 10 common phrases was $\kappa = 0.57$.

Table 13 presents the overall results. The percentage of sentences in which the translation changes significantly in terms of adequacy or fluency is around 50%-60%. The number of translations in which adequacy and fluency is improved by the SimTer_0.6 & FFF+ (50%) system doubled the number of cases in which it lowered its quality. This fact confirms the results obtained with the automatic evaluation measures. Table 13 shows that, about the 60% of the sentences underwent a lexical modification. Other aspects received less bus significant impact: reordering (21.01% of sentences affected), morphology (19.57%), addi-

Table 15: Statistics of the English–Spanish parallel CommonCrawl corpus.

Corpus		Sent.	Words	Vocab.	avg. length
CommonCrawl	Eng	1.84 M	46.54 M	750.01 K	25.22
	Spa		50.33 M	775.75 K	27.30

tions/omissions (17.63%), function words (13.04%) and, the least frequent, bad feedback (5.31%). All the aspects are improved, except for additions/omissions and, bad feedback. It is worth mentioning that the most frequent changes (lexical and reordering) are also changes whose benefit doubles the cases of a quality decrease. Contrary to what could be thought, the lexical correction addresses mistranslations in a greater deal than Out-of-Vocabulary words (OOV), as OOVs were reduced only by 0.03% under the best performance (cf. Table 14 for a detailed analysis).

5.2 Using a Web-Crawled Parallel Corpus

In this application scenario, we use the CommonCrawl corpus, a collection of parallel texts automatically mined from the Web (Smith et al., 2013). This corpus offers two interesting characteristics for our experiments: (*i*) its vocabulary and expressions go far beyond the controlled scenario of EPPS acts and the formality of the News UN corpora and (*ii*) it is a very noisy corpus. The large vocabulary size in Table 15 gives an intuition of the nature of its content, with a high presence of noise and spurious words. In addition, its size allows to set a trade-off between quantity (amount of new material selected) and quality (the threshold of the selection algorithm). Moreover, the selection method from the FAUST corpus is also applicable due to an analogy between scenarios: under CommonCrawl, we can compare the automatic translation of the source sentences against the references automatically obtained by the crawler and select the cases in which the latter is better. To perform the selection we use the same classifiers from the FAUST scenario without retraining or adaptation — as we want to study the generalization ability and no specific training material for the CommonCrawl corpus is available for the selection task. This experiment represents a double challenge: (*i*) determining if the presented proposal is also suitable for crawled parallel corpora, and (*ii*) studying whether the trained selection models generalize well across both corpora and domain. The CommonCrawl experimental setting can be seen as an *artificial* post-editing scenario: the references represent the edits, which ideally should provide more adequate translations.

In this experiment, we consider as baseline the best obtained system so far for translation of news texts (cf. Section 4.3.3). We call this baseline *Base_News_SimTer_0.6*. This baseline system might be considered as already strong, since it is 0.25 BLEU points better pure baseline system (trained once with all the data). In order to assess the trade-off between quantity and quality when using more parallel text, we enrich the baseline by adapting the original models with different portions of the CommonCrawl corpus and filtered either with the subsampling or FFF+ strategies. As in the FAUST scenario, we analyzed the translation performance depending on several factors: (*i*) the ratio of CommonCrawl data selected, (*ii*) the data selection strategy (FFF+ vs Subsampling), and (*iii*) the adaptation strategy. Table 16 and Figure 6 show the evaluation results over the 'test12' and 'test13' datasets from Section 4.3.1. The curves in Figure 6 show a consistent pattern with that observed in Figure 4 for the FAUST scenario, reinforcing the evidences: both FFF+ selection and SimTer_0.6 adaptation are important in order to obtain a final gain. Using CommonCrawl without selection does not result in any performance gain, but worsens the results slightly. Subsampling improves some metrics at 75% point but at the cost of worsening others. Moreover, concatening the 25% best ranked CommonCrawl to the training data 'Concat. FFF+(25%)' provides a slight improvement. However, the improvements obtained from 'Concat. FFF+(25%)' and 'SimTer_0.6 Subsampling(75%)' are not significant.

Table 16: Results obtained with the base and CommonCrawl-enriched SMT systems depending on different filtering and adaptation methods. For BLEU and NIST, '∗', '†' and '‡' indicate significant differences over the News_SimTer_0.6 system (experiment baseline) with 0.99, 0.95 and 0.90 confidence levels, respectively. The best results on each corpus are boldfaced.

Translation system		% of edits	BLEU	NIST	TER	METEOR	ULC
Adaptation	**Filtering**						
test12							
News_SimTer_0.6	—	0%	33.25	8.70	48.24	55.84	71.66
SimTer_0.6	SubSamp.	+25%	33.21	8.67	48.38	55.48	71.86
		+50%	33.38	8.68	48.25	55.54	72.04
		+75%	33.47†	8.70	48.27	55.81	72.32
	FFF+	+25%	33.73*	**8.78***	**47.84**	**56.21**	**72.52**
		+TH0-60%	**33.74***	8.75*	47.87	56.05	72.42
	—	+100%	33.19	8.68	48.46	55.50	71.20
Concat.	FFF+	+25%	33.41	8.71	48.09	55.88	72.44
		+TH0-60%	33.20	8.68	48.18	55.72	72.15
test13							
News_SimTer_0.6	—	0%	29.15	8.07	51.28	53.15	71.73
SimTer_0.6	SubSamp.	+25%	29.17	8.05	51.49	52.87	71.71
		+50%	29.22	8.03	51.43	52.71	71.63
		+75%	29.29	8.05	51.36	52.88	71.85
	FFF+	+25%	**29.61***	**8.13***	**50.99**	**53.39**	**72.41**
		+TH0-60%	29.57*	8.10‡	51.12	53.13	72.07
	—	+100%	29.32	8.07	51.36	52.87	71.53
Concat.	FFF+	+25%	29.27	8.07	51.11	53.14	72.26
		+TH0-60%	29.00	8.04	51.14	53.01	71.94

Results also show that selecting only the 25%-best CommonCrawl data produces the best improvement. This is approximately +0.50 BLEU and −0.40 TER on the test sets. It is important to recall that Base_News_SimTer_0.6 is the strong baseline. It is remarkable that the same selection models trained on the data from the FAUST scenario generalize well to the CommonCrawl domain adaptation scenario. The optimal 25% selection threshold represents a much stricter selection than that required in the FAUST scenario (50%). However, while in FAUST we were selecting around 3,000 sentences among 6,000, in this case we are selecting around 460 thousand sentences over 1.84 million. Hence, the final selection threshold is a compromise between the aggressiveness of the method and the minimum amount of new material necessary to cause a real impact. We also analyzed the effect of

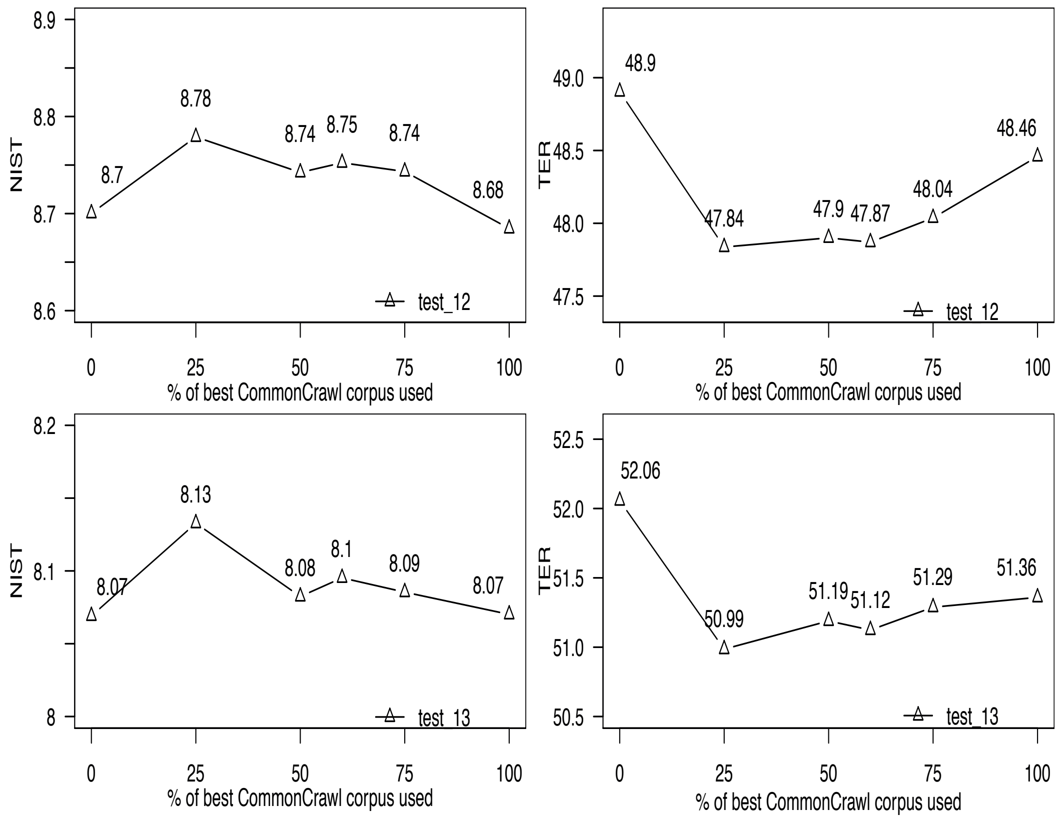
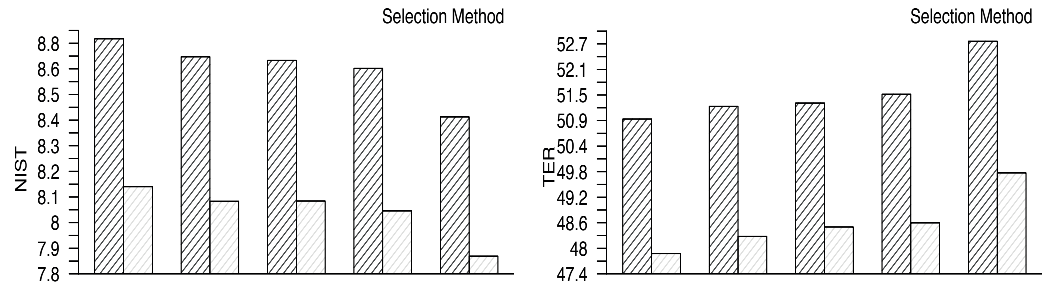


Figure 6: NIST and TER scores on the test12 and test13 corpora as a function of the percentage of best ranked CommonCrawl segments used.

selecting the 25% of the CommonCrawl data randomly (averaged over 10 times) or selecting the 25% of the data that our selection classifiers considered the worst. This analysis allows to assess how good our selection algorithm is on ranking the parallel examples of CommonCrawl corpus. Figure 7(a) shows the results, which indicate that the classifiers are able to generate sensible rankings to detect both best and worst examples. This allows to properly enrich the translation models, avoiding the negative effect of using the really bad instances. The results obtained with 25% of the examples selected at random are below the results obtained selecting the best 25% according to the classifier.

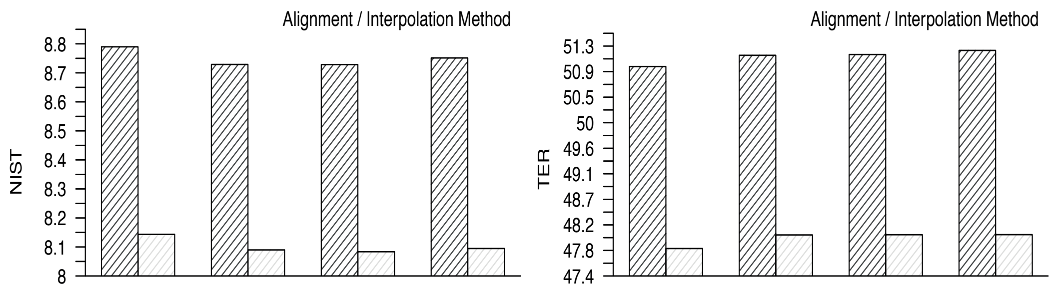
Lastly, we repeated the comparison of our alignment/adaptation strategy with the methods considered in the FAUST scenario experiment. Figure 7(b) shows the obtained results. Similar to Figure 5, the differences are small in favor of `SimTer`. However, `SimTer` 60-40% performs significantly better than the other alignment/adaptation strategies in both test sets ($p < 0.01$).



	25%_best	baseline	all	25%_rand	25%_worst
test_12	8.78*	8.7	8.68	8.65	8.44
test_13	8.13*	8.07	8.07	8.03	7.83

	25%_best	baseline	all	25%_rand	25%_worst
test_12	50.99	51.28	51.36	51.57	52.81
test_13	47.84	48.24	48.46	48.55	49.72

(a) Achieved by using different selection criteria: baseline (0%), all (100%), best 25%, random 25% and worst 25%.



	SimTer 0.6	mGIZA++ PPL	SimTer PPL	mGIZA++ 0.6
test_12	8.78*	8.72	8.72	8.74
test_13	8.13*	8.08	8.07	8.08

	SimTer 0.6	mGIZA++ PPL	SimTer PPL	mGIZA++ 0.6
test_12	50.99	51.18	51.19	51.26
test_13	47.84	48.07	48.08	48.08

(b) Achieved by applying different alignment/adaptation methods.

Figure 7: NIST and TER scores on the test12 and test13 corpora. '$*$' indicates significant differences over the Baseline system with 0.99 confidence level.

6. Conclusions

In this article we proposed a new automatic strategy to incrementally train machine translation (MT) models with the edited translations coming from casual unreliable users. Our strategy builds upon three main blocks, namely: (*i*) automatic identification of useful user-edited instances (UE); (*ii*) alignment of the UEs with the source text, focusing on the *errors* made by the original MT system; and (*iii*) incorporation of the new parallel segments through specific translation models trained with UEs. Our proposal is novel in the application of some techniques from information retrieval, quality estimation and domain adaptation to the problem of MT system enrichment. The datasets explored have also interesting and challenging properties.

The selection of useful UEs is important to filter out noisy feedback from the users. We accomplish this by training a classifier from supervised data using features derived from similarity metrics used in information retrieval and MT quality estimation. Although the classification results achieved are only moderate, classification scores allow to approximately rank UEs by quality and tune different selection thresholds. Experiments show that this is a useful strategy to select examples which, combined with the other two steps, yields significant improvements over the original MT system.

Regarding the *source–translation*–UE alignment, we proposed `SimTer`, a simple incremental approach based on pivoting, which uses a TER alignment augmented with similarity features. This approach has two advantages: it does not depend on previous software-specific alignment models (e.g., `GIZA++`, `mGIZA++` or `Berkeley Aligner`) and the monolingual nature of the alignment implicitly allows the algorithm to focus on correcting translation errors, rather than achieving an optimal alignment between words. By experimenting with two real datasets, we showed the positive contribution of `SimTer` in the MT enrichment pipeline. For our particular application, using `SimTer` is better than using existing general-purpose aligners, such as `mGIZA++`, especially in the CommonCrawl scenario. The experiments also confirmed the validity of the proposal, in terms of computational efficiency.

The third step deals with building UE-specific translation models using standard phrase extraction and scoring tools. After experimentally analyzing different ways of combining the UE-based and the original translation models, we concluded that a simple linear interpolation is a good and efficient strategy. By properly tuning the parameters, this combination has a real impact in the final translation models, something that, for instance, perplexity minimization is not able to achieve.

The complete architecture was thoroughly tested with real UEs collected from non-professional users through a commercial on-line translation portal (the so called FAUST scenario). We experimented with different thresholds to select examples and alternative ways to perform the alignment and the integration of the new aligned sentences. Results showed that our approach significantly improves the translation quality of a basic, general purpose SMT system, being generally superior to alternative methods. Apart from evaluating with several automatic quality measures, we also conducted a manual analysis in order to verify the quality of improvements and to gain more insight on the cases in which the enriched MT system performs better or worse. The improvements did not come mainly from a reduction of the out-of-vocabulary words, which are actually reduced only marginally. The major improvements in translation quality came from a much better lexical selection, reordering, and morphology. On the down side, the enriched system introduces from time to time incorrect words or expressions learned from wrongly selected and aligned examples. It also performed poorly in terms of adding and omitting spurious words, slightly worsening quality over the baseline system.

The approach is general enough to be applied to different scenarios. We finally used CommonCrawl, a collection of parallel texts automatically extracted from the Web, to enrich a general purpose baseline SMT system. The same three steps were applied; a selection was also necessary in this case because the corpus is very noisy, due to its automatic extraction. Exactly the same classifiers trained with the FAUST corpus were used to identify examples in which the automatically extracted target sentence is better than the automatic translation of the source provided by the baseline translation system. The classifiers showed robustness even when noisy references were used instead of UEs evincing their capacity to deal with human or automatically-generated noise.The same conclusions can be drawn in the adaptation experiment, which shows that our methodology works well across different corpora sources and types of noise.

Acknowledgments

The major part of this work was carried out when the authors worked at *TALP Research Center - Universitat Politècnica de Catalunya*. We would like to thank Nadir Durrani for the proofreading of the paper and also the anonymous reviewers for their valuable feedback. This work was partially funded by the Spanish *Ministerio de Economía y Competitividad*, under contracts TEC2012-38939-C03-02 and TIN2012-38523-C02-02, as well as from the European Regional Development Fund (ERDF/FEDER) and the European Community's FP7 (2007-2013) program under the following grants: 247762 (FAUST, FP7-ICT-2009-4-247762) and 246016 (ERCIM "Alain Bensoussan" Fellowship).

References

Ambati, V., Vogel, S., & Carbonell, J. (2010). Active Learning and Crowd-Sourcing for Machine Translation. In *Proceedings of the LREC*, pp. 2169–2174.

Axelrod, A., He, X., & Gao, J. (2011). Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 355–362, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Barrón-Cedeño, A., Màrquez, L., Henríquez, Q. C. A., Formiga, L., Romero, E., & May, J. (2013). Identifying useful human correction feedback from an on-line machine translation service. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, pp. 2057–2063. AAAI Press.

Bertoldi, N., Cettolo, M., & Federico, M. (2013). Cache-based online adaptation for machine translation enhanced computer assisted translation. In *Proc. of MT Summit*, pp. 35–42.

Bisazza, A., Ruiz, N., & Federico, M. (2011). Fill-up versus Interpolation Methods for Phrase-based SMT Adaptation. In *Proceedings of IWSLT*, pp. 136–143.

Blain, F., Schwenk, H., Senellart, J., & Systran, S. (2012). Incremental adaptation using translation information and post-editing analysis. In *Proceedings IWSLT 2012*, pp. 229–236.

Byrd, R. H., Lu, P., Nocedal, J., & Zhu, C. (1995). A Limited Memory Algorithm for Bound Constrained Optimization. *SIAM Journal on Scientific Computing*, *16*(5), 1190–1208.

Callison-Burch, C., Bannard, C., & Schroeder, J. (2005). Scaling phrase-based statistical machine translation to larger corpora and longer phrases. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pp. 255–262, Stroudsburg, PA, USA. Association for Computational Linguistics.

Callison-Burch, C., Koehn, P., Monz, C., Post, M., Soricut, R., & Specia, L. (2012). Findings of the 2012 Workshop on Statistical Machine Translation. In *Proceedings of the*

Seventh Workshop on Statistical Machine Translation, pp. 10–51, Montréal, Canada. Association for Computational Linguistics.

Cappé, O., & Moulines, E. (2009). On-line expectation-maximization algorithm for latent data models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *71*(3), 593–613.

Cettolo, M., Federico, M., Servan, C., & Bertoldi, N. (2013). Issues in Incremental Adaptation of Statistical MT from Human Post-edits. In *Proceedings of MT Summit XIV Workshop on Post-editing Technology and Practice*, pp. 111–118.

Denkowski, M., & Lavie, A. (2011). Meteor 1.3: Automatic Metric for Reliable Optimization and Evaluation of Machine Translation Systems. In *Proceedings of the 6th Workshop on Statistical Machine Translation*, pp. 85–91, Edinburgh, Scotland.

European Comission - 7th Framework Program (2010). Matecat, FAUST and Casmacat Projects. `http://www.matecat.com` `http://www.faust-fp7.eu` `http://www.casmacat.eu`. Accessed: 2015-02-01.

FAUST (2013). Final report on the methods for the evaluation of translation requests, system outputs, and modelling of user feedback. Tech. rep. D4.6, FAUST – Feedback Analysis for User-Adaptive Statistical Translation. ftp://svr-ftp.eng.cam.ac.uk/pub/pub/faust-pub/Deliverables/FAUSTD4.6.pdf.

Formiga, L., Henríquez Q., C., Hernández, A., Mariño, J., Monte, E., & Fonollosa, J. (2012). The talp-upc phrase-based translation systems for wmt12: Morphology simplification and domain adaptation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pp. 275–282, Montréal, Canada. Association for Computational Linguistics.

Foster, G., Goutte, C., & Kuhn, R. (2010). Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 451–459, Cambridge, MA. Association for Computational Linguistics.

Foster, G., & Kuhn, R. (2007). Mixture-Model Adaptation for SMT. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pp. 128–135.

Gale, W., & Church, K. (1993). A Program for Aligning Sentences in Bilingual Corpora. *Computational Linguistics*, *19*, 75–102.

Gao, Q., & Vogel, S. (2008). Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pp. 49–57, Columbus, Ohio. Association for Computational Linguistics.

Giménez, J., & Màrquez, L. (2010). Asiya: An Open Toolkit for Automatic Machine Translation (Meta-)Evaluation. *The Prague Bulletin of Mathematical Linguistics*, pp. 77–86.

Google Inc. (2015). Google Translate. `http://translate.google.com`. Accessed: 2015-02-01.

Haddow, B., & Germann, U. (2011). Moses Incremental Training. http://www.statmt.org/moses/?n=Advanced.Incremental. Accessed: 2015-08-11.

Hardt, D., & Elming, J. (2010). Incremental re-training for post-editing smt. In *In proc. of AMTA 2010: the Ninth conference of the Association for Machine Translation in the Americas*, Denver, CO. USA.

Henriquez, C. (2014). *Improving statistical machine translation through adaptation and learning*. Ph.D. thesis, Universitat Politècnica de Catalunya.

Henríquez, C., Mariño, J., & Banchs, R. (2011). Deriving translation units using small additional corpora. In *Proceedings of the 15th Conference of the European Association for Machine Translation*, pp. 121–128.

Hsu, C.-W., Chang, C.-C., & Lin, C.-J. (2003). A practical guide to support vector classification. Tech. rep., Department of Computer Science, National Taiwan University. `http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf`.

Joachims, T. (1999). Making large-scale support vector machine learning practical. In Schölkopf, B., Burges, C. J. C., & Smola, A. J. (Eds.), *Advances in Kernel Methods*, pp. 169–184. MIT Press, Cambridge, MA, USA.

Koehn, P. (2005). Europarl: A Parallel Corpus for Statistical Machine Translation. In *Machine Translation Summit X*, pp. 79–86, Phuket, Thailand.

Koehn, P., & Hoang, H. (2007). Factored Translation Models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 868–876, Prague, Czech Republic.

Lambert, P., de Gispert, A., Banchs, R., & Mariño, J. B. (2005). Guidelines for word alignment evaluation and manual alignment. *Language Resources and Evaluation*, *39*(4), 267–285.

Levenberg, A., Callison-Burch, C., & Osborne, M. (2010). Stream-based translation models for statistical machine translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 394–402, Los Angeles, California. Association for Computational Linguistics.

Lopez, A. (2008). Tera-scale translation models via pattern matching. In *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pp. 505–512, Stroudsburg, PA, USA. Association for Computational Linguistics.

Martínez-Gómez, P., Sanchis-Trilles, G., & Casacuberta, F. (2012). Online adaptation strategies for statistical machine translation in post-editing scenarios. *Pattern Recognition*, *45*(9), 3193 – 3203. Best Papers of Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA'2011).

Matecat (2015). Matecat official repository. https://github.com/matecat/MateCat. Accessed: 2015-07-24.

Mathur, P., Mauro, C., & Federico, M. (2013). Online learning approaches in computer assisted translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pp. 301–308, Sofia, Bulgaria. Association for Computational Linguistics.

Mcnamee, P., & Mayfield, J. (2004). Character N-Gram Tokenization for European Language Text Retrieval. *Information Retrieval*, *7*(1-2), 73–97.

Microsoft Inc. (2015). Bing Translator. `http://www.bing.com/translator`. Accessed: 2015-02-01.

Neal, R. M., & Hinton, G. E. (1998). A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pp. 355–368. Springer.

Nelder, J. A., & Mead, R. (1965). A Simplex Method for Function Minimization. *The Computer Journal*, *7*, 308–313.

NIST (2002). Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics. Tech. rep., National Institute of Standards and Technology. `http://www.itl.nist.gov/iad/mig/tests/mt/doc/ngram-study.pdf`.

Och, F. J., & Ney, H. (2003). A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, *29*, 19–51.

Ortiz-Martínez, D., García-Varea, I., & Casacuberta, F. (2010). Online learning for interactive statistical machine translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 546–554, Los Angeles, California. Association for Computational Linguistics. `http://www.aclweb.org/anthology/N10-1079`.

Ortiz-Martínez, D., Sanchis-Trilles, G., González-Rubio, J., & Casacuberta, F. (2013). Progress report on adaptive translation models. Tech. rep. D4.2, Casmacat: Cognitive Analysis and Statistical Methods for Advanced Computer Aided Translation.

Padró, L., Collado, M., Reese, S., Lloberes, M., & Castellón, I. (2010). FreeLing 2.1: Five Years of Open-Source Language Processing Tools. In *Proceedings of the 7th Language Resources and Evaluation Conference (LREC 2010)*, La Valletta, MALTA.

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Pighin, D., Màrquez, L., & May, J. (2012). An Analysis (and an Annotated Corpus) of User Responses to Machine Translation Output. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey.

Potet, M., Esperança-Rodier, E., Blanchon, H., & Besacier, L. (2011). Preliminary experiments on using users' post-editions to enhance a smt system. In *Proceedings of the 15th Conference of the European Association for Machine Translation*, pp. 161–168.

Pouliquen, B., Steinberger, R., & Ignat, C. (2003). Automatic Identification of Document Translations in Large Multilingual Document Collections. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP-2003)*, pp. 401–408, Borovets, Bulgaria.

Reverso-Softissimo (2015). Reverso — Free online translator and dictionary based on SDL technology. `http://www.reverso.net`. Accessed: 2015-02-01.

Riezler, S., & Maxwell, J. (2005). On some pitfalls in automatic evaluation and significance testing for MT. In *ACL-05 Workshop on Intrinsic and Extrinsic Evaluation*

Measures for Machine Tranlsation and/or Summarization (MTSE'05) at the 43rd Annual Meeting of the Association for Computational Linguistics, Ann Arbor, Michigan, USA.

Schwenk, H., & Koehn, P. (2008). Large and diverse language models for statistical machine translation.. In *Proceedings of IJCNLP*, pp. 661–666, Hyderabad, India.

Sennrich, R. (2012). Mixture-Modeling with Unsupervised Clusters for Domain Adaptation in Statistical Machine Translation. In *Proceedings of the 16th EAMT Conference*.

Simard, M., Foster, G. F., & Isabelle, P. (1992). Using cognates to align sentences in bilingual corpora. In *in Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation*, pp. 67–81.

Simard, M., Goutte, C., & Isabelle, P. (2007). Statistical phrase-based post-editing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pp. 508–515, Rochester, New York. Association for Computational Linguistics.

Smith, J., Saint-Amand, H., Plamada, M., Koehn, P., Callison-Burch, C., & Lopez, A. (2013). Dirt cheap web-scale parallel text from the Common Crawl. In *Proceedings of the 2013 Conference of the Association for Computational Linguistics (ACL 2013)*, Sofia, Bulgaria. Association for Computational Linguistics.

Snover, M., Madnani, N., Dorr, B., & Schwartz, R. (2009). TER-Plus: Paraphrase, Semantic, and Alignment Enhancements to Translation Edit Rate. *Machine Translation*, *23*(2), 117–127.

Witten, I., & Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques* (2 edition). Morgan Kaufmann, San Francisco, CA.