Identifying Useful Human Correction Feedback from an On-line Machine Translation Service*

Alberto Barrón-Cedeño[1,2] **Lluís Màrquez**[1] **Carlos A. Henríquez Q.**[1]
Lluís Formiga[1] **Enrique Romero**[1] **Jonathan May**[3]
[1] Universitat Politècnica de Catalunya, Spain
[2] Universidad Politécnica de Madrid, Spain
[3] SDL Language Weaver
{albarron,lluism,eromero}@lsi.upc.edu {carlos.henriquez,lluis.formiga}@upc.edu jmay@sdl.com

Abstract

Post-editing feedback provided by users of on-line translation services offers an excellent opportunity for automatic improvement of statistical machine translation (SMT) systems. However, feedback provided by casual users is very noisy, and must be automatically filtered in order to identify the potentially useful cases. We present a study on automatic feedback filtering in a real weblog collected from Reverso.net. We extend and re-annotate a training corpus, define an extended set of simple features and approach the problem as a binary classification task, experimenting with linear and kernel-based classifiers and feature selection. Results on the feedback filtering task show a significant improvement over the majority class, but also a precision ceiling around 70-80%. This reflects the inherent difficulty of the problem and indicates that shallow features cannot fully capture the semantic nature of the problem. Despite the modest results on the filtering task, the classifiers are proven effective in an application-based evaluation. The incorporation of a filtered set of feedback instances selected from a larger corpus significantly improves the performance of a phrase-based SMT system, according to a set of standard evaluation metrics.

1 Introduction

On-line translation services, such as Google Translate or Reverso.net, produce instantaneous automatic translation to user-provided text fragments from one natural language into another. They are used by all types of users for all types of purposes. Consequently, a huge volume of translation requests per day are processed, and these exhibit a high diversity in terms of topic, genre, style, and length. This rich translation workflow involving millions of users offers an excellent opportunity to collect useful feedback information for dynamically improving or adapting translation systems.

Currently, most on-line translation platforms implement procedures for gathering users' feedback on the output of their translation requests (e.g., by making vocabulary suggestions or directly post-editing the output sentences), which may be further used to improve the translation engine. In practice, however, such feedback collection is irregular and very noisy for various reasons. Well-intended users may introduce errors in their post-editing process (product of a limited knowledge of the languages involved or just by inattention), producing translations that are even worse than the automatic ones. They can also neglect interface instructions and submit comments (either about the translation or the system), rather than a corrected translation. Users with ill intent may submit malicious translations or simply impertinent comments. Moreover, translation services are generally exploited by casual users, making translation requests very noisy and ungrammatical, certainly far from the controlled standard language translation scenarios seen in most MT evaluation workshops and conferences (e.g., [Callison-Burch *et al.*, 2012]). All these factors indicate that automatic mechanisms are needed to identify useful user responses for system improvement and filter out the erroneous ones.

In this work we experiment with a supervised approach to user feedback filtering, cast as a binary classification problem. Departing from our preliminary study Pighin *et al.* [2012]: (*i*) we extend and re-annotate a training English-to-Spanish corpus; (*ii*) we propose and adapt new features for characterizing a text and its alternative translations; and (*iii*) we explore different configurations of SVM learning for the problem (including kernels, parametrization, and feature selection). The analysis of the manual annotation process and the results of automatic feedback classification reveal that the problem is hard for humans and also very difficult to learn with the shallow features used. As a result, high-precision filters at acceptable levels of recall are not achievable under the current setting. Nonetheless, (*iv*) we show that the predictions by the feedback classifiers are still useful and can be integrated into a phrase-based SMT system, significantly improving translation quality on real-world translation requests.

The rest of the paper is organized as follows. Section 2 describes corpora compilation and annotation. Section 3 introduces the problem setting and lists the complete set of fea-

*This research work was carried out during the tenure of an ERCIM "Alain Bensoussan" Fellowship Programme. The research leading to these results received funding from the EU FP7 Programme 2007-2013 (grants 246016 and 247762) and the Spanish Ministry of Economy and Competitiveness (TIN2009-14675-C03, TEC2012-38939-C03-02).

tures used for feedback classification. Section 4 describes the learning procedure and the analysis of results. Section 5 shows the positive impact of the selected feedback instances in a real SMT system. Section 6 concludes the paper and outlines several directions for further research.

2 An Annotated Corpus with Human Correction Feedback

We work with a corpus that is an enhancement of the English–Spanish partition of the Faust Feedback Filtering (FFF) corpus [Pighin *et al.*, 2012]. The corpus, provided by the FAUST project[1], contains 245 tuples collected from the weblogs of the on-line translation platform Reverso.net, including:

src source sentence in English (user's query to the system), only sentences of up to 20 words are considered;

trans automatic translation of *src* into Spanish;

feed human feedback, a potentially improved translation of *src*, provided by the user;

bt automatic translation of *trans* back into English;

bf automatic translation of *feed* into English; and

cl manually assigned class label, i.e., whether *feed* is a better translation of *src* than *trans*.

To create our corpus, we extended the FFF with 305 new instances from the same weblog, and added an extra field: '*ilang*', the language set in the translator's interface. We manually annotated all the examples with class labels (including a revision of the original 245), following the procedure described in section 2.2. We refer to this corpus as FFF$^+$.[2]

2.1 Annotation Guidelines

Translation quality is a multifaceted concept, which encompasses *adequacy* (i.e., the translation conveys the meaning of the source), *fluency* (i.e., the translation is a fluent utterance in the target language), and other aspects; some of them defined on a per-application basis, e.g., vocabulary usage, language register, post-editing effort, etc. The ultimate goal of this study is to automatically improve on-line SMT quality by incorporating casual users' correction feedback (cf. Section 5), so one has to be cautious when considering which instances are useful. To determine which fragment (*trans* or *feed*) is a better translation of *src*, we followed the 9 rules (partially derived from Pighin *et al.* [2012]) described below. Table 1 includes some real examples for illustration.
Feed is considered **useful**, if:

1. *feed* is strictly more adequate than *trans* (even if still imperfect); or
2. *src* had a small typo, causing the translator to fail, but *feed* is clearly the expected translation.

Feed is considered **useless** if:

3. *feed* is indeed a comment (e.g., about the translator);
4. *feed* presents valid corrections or translation alternatives, but is not phrased as a grammatical sentence (e.g., "X should be better translated as Y";
5. *feed* is identical to *trans*;
6. *feed* is better than *trans* in some parts but worse in others;
7. *src* is too noisy for a translation to be feasible (e.g., it contains grammar errors or junk content); or
8. *feed* contains only noise.

To these, we add an additional rule:

9. If both *trans* and *feed* are valid translations of *src*, select the most natural according to the available context.

2.2 Corpus Annotation

Two volunteers A_1 and A_2 (native Spanish speakers with high command of English), annotated the FFF$^+$ corpus. On the basis of the aforementioned guidelines, they had to choose label "1" for useful or "0" for useless given triples composed of *src*, *trans*, and *feed*. When in doubt, "0" was the preferred label. The process involved two phases: (*i*) annotation of a small subset for calibration, and (*ii*) annotation of the remaining examples. At the end of each step, annotators discussed disagreed instances until a consensus label was adjudicated—occasionally causing minor criteria adjustments.

In the calibration phase, A_1 and A_2 judged 25 randomly selected instances from the original FFF corpus. The levels of inter-annotator agreement, in terms of Cohen's kappa coefficient [Cohen, 1960], were $\kappa(A_0, A_1) = 0.33$, $\kappa(A_0, A_2) = 0.22$, and $\kappa(A_1, A_2) = 0.66$, where A_0 represents the original annotation. Three common disagreement triggers were identified: (*a*) ill-formed *src* entries for which no correct translation was possible, but proper feedback had been provided; (*b*) long snippets with both better and worse sections; and (*c*) single-word entries for which both *trans* and *feed* represented valid translations, but no context was available to discriminate. Disagreements at this stage resulted in rules 7 and 9. After observing the low agreement with respect to A_0, every instance in the FFF corpus was re-annotated from scratch with the revised criteria, obtaining a much higher final agreement $\kappa(A_0, A_1) = 0.58$. During the second phase, A_1 and A_2 annotated the 305 new instances, obtaining a comparable $\kappa(A_1, A_2) = 0.53$ (corresponding to an absolute agreement ratio close to 80%).

In summary, the 550 instances were independently labeled by at least two human annotators, and the disagreement cases ($\sim$100) were all discussed until consensus was reached. The levels of agreement achieved (0.5–0.6) can be considered moderately high but not fully satisfactory. This indicates the inherent difficulty of this task, even for humans. Human perception of translation quality is a very subjective matter, dependent on small details which are difficult to capture in a set of simple guidelines. This effect is amplified by the noisy nature of the input text we are working with, where often the input sentences lack the necessary context to make fully reliable decisions. Not surprisingly, in Section 3 we observe how this task is also difficult for the automatic classification methods, which use much less informed features and semantic information than humans when deciding on translation quality.

[1] This project is focused on the development of machine translation systems that can respond rapidly and intelligently to user feedback (`http://www.faust-fp7.eu`).

[2] The FFF and FFF$^+$ corpora are freely available at `http://www.faust-fp7.eu/faust/Main/DataReleases`

Table 1: Instances of source, translation and human feedback tuples. ID corresponds to the rule number

ID	source	automatic translation	user feedback	explanation
1	I am still in a meeting	Soy todavía en una reunión	Aun estoy en una reunión	Verb fixed (still a diacritic missing)
2	Relentlessy	Relentlessy	implacablemente	Typo in source
3	He 's cinema	Él ' s cine	Ustedes no saben nada de inglés	Unrelated comment
4	saw	vio	vieron o vio	The entire sentence is not a valid translation of *src*
5	He learned that he would have to pay for the wedding himself	Él se enteró de que tendría que pagar por la boda él mismo	Él se enteró de que tendría que pagar por la boda él mismo	Identical sentences
6	position the rear piston at TDC by turning the crankshaft clockwise (right side view)	coloque el pistón de reverso en TDC por girando el cigüeñal en el sentido de las agujas del reloj (la vista lateral derecha)	coloque el pistón del cilindro trasero en posiciónTDC girando el cigüeñal en el sentido de las agujas del reloj (mirando del lado lateral derecho)	Some sections are better, some others are worse
7	you believe know, both is a ignorant	usted cree saben, ambos son unos ignorantes	usted cree saber, pero es un ignorante	Source is too low quality to expect any translation
8	Don't Stop the party	No Pare el partido	wqkjkwj	Just keyboard-striking
9	wherever they are found	cualquier parte donde ellos sean encontrados	dondequiera que se encuentren	Feedback flows better

Nonetheless, this task is only an intermediate filtering step to improve translation systems. Section 4 shows that despite its imperfections, the resulting automatic assessments can be used to significantly improve a phrase-based SMT system.

3 Features for Feedback Filtering

We model feedback filtering as a binary classification problem: tuples $(src, trans, feed)$ have to be assigned a positive label iff *feed* is a better translation from *src* than *trans*. We consider four sets of very simple features to characterize the tuple fields as well as the relationships within them: **surface**, **back-translation**, **noise-based**, and **similarity-based**. The former two sets are derived from those described in [Pighin *et al.*, 2012]. Back-translation and similarity-based features use the texts resulting from translating either *trans* or *feed* back into the source language. Text pre-processing includes case folding and diacritic elimination.

Surface features

These features consider *src*, *trans*, *feed*, and *ilang* (not considered before); i.e., information available at operation time:

(a) Length of [*src*, *trans*, *feed*] in tokens.

(b) Ratio of lengths in tokens between (*trans*, *src*), (*feed*, *src*), and (*trans*, *feed*).

(c) Ratio between length in characters and tokens for [*src*, *trans*, *feed*].

(d) Levenshtein distance between (*trans*, *feed*) divided by length of *trans*; both at token and character level.

(e) Number of words in *trans* not in *feed* divided by number of words in *trans* (and vice versa).

(f) Vocabulary containment [Broder, 1997] between (*trans*, *feed*), (*src*, *trans*), and (*src*, *feed*).

(g) Ratio of (c)'s resulting features between (*trans*, *src*), (*feed*, *src*), and (*trans*, *feed*).

(h) Length of longest word for [*src*, *trans*, *feed*].

(i) Ratio of (h)'s resulting features between (*trans*, *src*), (*feed*, *src*), and (*trans*, *feed*).

(j) Two binary features: 1 if $feed = trans$, $src = trans$; 0 otherwise.

(k) Three (complementary) binary features: 1 if interface is in English, Spanish, or other; 0 otherwise.

Back-translation features

These are also surface features, but now consider the back-translations of *trans* (*bt*) and *feed* (*bf*):

(l) Levenshtein distance between (*src*, *bt*) and (*src*, *bf*); both at token and character level.

(m) Ratio between (l)'s resulting features for [(*src*, *bf*), (*src*, *bt*)]; both at token and character level.

(n) Vocabulary containment [Broder, 1997] between (*src*, *bt*) and (*src*,*bf*)

(o) Number of words in *src* not in *bt* (*bf*) divided by number of words in *src*.

Noise-based features

These binary features are designed to indicate the likelihood of any of the text fragments including noisy sections. Some of them try to determine a "length-based translation difficulty".

(p) 1 if *src* contains one single word; 0 otherwise.

(q) 1 if *src* contains up to 5 words; 0 otherwise.

(r) 1 if *src* contains between 6 and 10 words; 0 otherwise.

(s) 1 if *src* contains more than 11 words; 0 otherwise.

(t) Six features: 1 if [*src*, *trans*, *feed*] contains a word of length in the range $[10, 14]$ and $[15, \infty)$; 0 otherwise

(u) Three features: 1 if [*src*, *trans*, *feed*] contains a sequence of three repeated characters; 0 otherwise.

Similarity-based features

This set assesses the resemblance of the elements in the tuple to each other on the basis of different similarity metrics, which are designed to be a better option than Levenshtein-based features.

(v) Character 3-gram cosine similarity between all pairwise choices from { *src*, *trans*, *feed*, *bt*, *bf* }.

(w) *Cognateness*-based [Simard *et al.*, 1992] cosine similarity between all pairwise choices from { *src*, *trans*, *feed*, *bt*, *bf* }.

(x) Two features using length factor [Pouliquen *et al.*, 2003] (lf) with the parameters estimated by [Potthast *et al.*, 2011]: lf(*src*, *trans*) and lf(*src*, *feed*).

(y) Two complementary boolean features: lf(*src*, *trans*) $>$ lf(*src*, *feed*) and lf(*src*, *trans*) $<$ lf(*src*, *feed*).

These features are based on concepts borrowed from cross-language information retrieval and machine translation. Short character n-grams are considered a useful choice when comparing texts across languages [McNamee and Mayfield, 2004]. Pseudo-cognates and the relative length of translations are well known options for aligning bilingual corpora [Simard *et al.*, 1992; Gale and Church, 1993].

4 Learning and Analysis of Classifiers

We trained support vector machines (SVM) [Vapnik, 1995] with the features described in the previous section to learn the feedback classifiers. In particular, we used SVM^{light} [Joachims, 1999] and experimented with three standard kernels: linear, degree 2 polynomial[3], and RBF.

The tuning of the classifiers was made with 90% of the FFF^{+} corpus. The remaining 10% was left aside for testing purposes. Before training, non-binary features were normalized. Values were bounded in the range $\mu \pm 3 * \sigma^2$ to reduce the impact of outliers during normalization. Normalization was then applied by means of z-score: $x = (x - \mu)/\sigma$. Later on, mean and standard deviation of the tuning dataset were used to normalize the remaining instances, which were then used to perform the final test.

We evaluated the task performance on the basis of standard classification and recognition measures, namely: *classification accuracy*, *precision* (ratio of correctly predicted useful instances and all predicted useful instances), *recall* (ratio of correctly predicted useful instances and all useful instances in the dataset), and F_1 (the harmonic mean of precision and recall).

Our SVM training strategy aims to optimize F_1. It consists of two iterative steps: (*a*) parameter tuning: a grid search for the most appropriate SVM parameters [Hsu *et al.*, 2003], and (*b*) feature selection: a wrapper strategy, implementing backward elimination to discard redundant or irrelevant features [Witten and Frank, 2005, p. 294]. We iteratively applied these two steps on the basis of a 10-fold cross-validation until F_1 stoped improving. Our backward elimination strategy is inspired by the well known concept of *look ahead*. The process starts by considering the entire feature set and proceeds iteratively, eliminating one feature per iteration until no F_1 improvement is observed. In order to decide which feature to eliminate, all possible pairs of features are inspected. In detail, let K be the active features at a certain iteration:

1. Build a classifier with the dataset D_K and obtain its F_1.
2. Build the $K*(K-1)/2$ possible datasets with $K-2$ features, train classifiers, and evaluate their performance.
3. If the best dataset gets a better (or equal) performance than that of D_K, build datasets D_{K-a} and D_{K-b}, where a, b are the two implied features.
4. Discard either a or b—the one that yields a better performance—from D_K and go back to step 1.

If step 3 is unable to get a better dataset, the process is repeated, generating only k datasets with $k-1$ features. The process stops if no better values of F_1 are obtained.

Table 2: Micro-averaged evaluation for model tuning. We consider three SVM kernels, basic (top) and enhanced (bottom) feature sets and applying feature selection (or not). The number of features maintained after convergence of the feature selection process is presented in column $|M|$.

kernel	$Acc.$	F_1	$Prec.$	$Rec.$	$\|M\|$
linear	66.5 (63.8)	69.9 (64.8)	64.8 (64.7)	76.0 (65.0)	21
poly	67.5 (61.0)	73.7 (69.1)	63.0 (58.2)	89.0 (85.0)	24
RBF	60.8 (60.4)	70.3 (70.0)	57.5 (57.3)	**90.6 (90.2)**	34
linear	70.5 (65.9)	73.6 (69.3)	**68.1** (64.3)	79.9 (75.2)	43
poly	70.3 (66.3)	**75.8** (72.3)	65.2 (62.5)	**90.6** (85.8)	60
RBF	**71.1** (67.9)	74.5 (71.6)	**68.1** (65.6)	82.3 (78.7)	52

The results of the model tuning process are summarized in Table 2. The table presents the results of all three kernels, for both *basic* (surface and back-translation) and *enhanced* (basic + noise- and similarity-based) features sets, and the application or not of the feature selection. A first observation is that the feature selection procedure consistently results in better accuracy and F_1 scores, i.e., it not only discards irrelevant features but also some harmful ones. Moreover, enhancing the basic feature set with the newly proposed features causes the performance of the different classifiers to increase in all the evaluation measures. We are particularly interested in precision, as we would like to avoid the insertion of noisy entries into the translation system. The highest precision, 68.1%, is obtained with both linear and RBF kernels; with identical values of accuracy and similar recall. The rest of our experiments are carried out with the selected features only.

Performance differences with respect to the length of *src* are shown in Table 3. It is unsurprising that one-word instances get the lowest F_1 —they do not provide information enough to characterize them (indeed, these were problematic cases for the manual annotation as well). Considering the low percentage of one-word useful instances, it could be worth simply discarding them. The best results come with mid-length sentences which also include the highest proportion of positive instances.

The precision-recall curves for the three resulting classifiers are displayed in Figure 1. The behavior of the three classifiers is very similar for the highest levels of recall. The polynomial kernel performs at its best with a high recall, at

[3]We explored higher degrees ($\leq$5) without improving results.

Table 3: Cross-validation results broken down by length of the source sentence. Columns include $\mu \pm \sigma$ values of Accuracy and F_1 for the three kernel classifiers, number of instances, and percent classified positive for each set of instances.

#tokens	$Acc.$	F_1	size	%pos
1	74.5±2.66	32.2±7.48	115	23.5
2–5	61.0±3.80	66.4±3.14	199	54.3
6–10	74.5±0.98	84.1±0.51	102	70.6
11–15	65.4±1.06	76.0±1.82	54	57.4
16–20	65.3±2.31	76.7±2.33	25	64.0

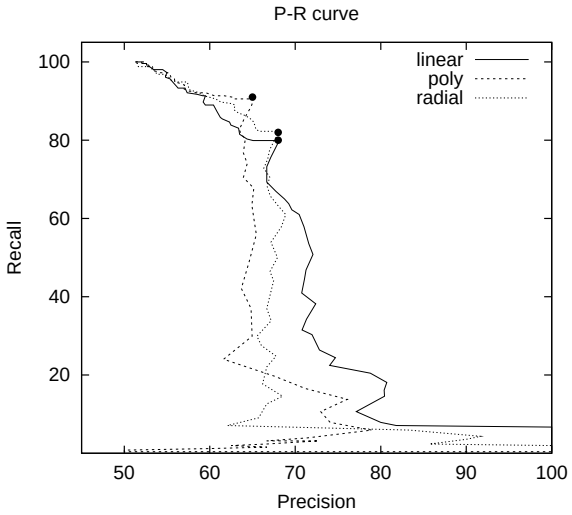

Figure 1: Precision-recall curve over the development partition for the three kernels. Values for the SVM classification threshold are highlighted.

the cost of a relatively low precision. Both linear and RBF kernels achieve their best result at similar levels of recall and precision. Still, precision of the linear kernel overcomes the rest from this point, a very important characteristic when considering the task of selecting error-free sets of useful feedback for MT improvement. All in all, the three classifiers exhibit a poor precision performance with values only slightly over 70% at acceptable levels of recall. Significantly higher values are reached only when recall is lower than 10%. This behavior reflects the difficulty of confidently characterizing positive examples with the shallow features we are using. The room for improvement may come at the form of deeper, even semantic, features.

What features are considered relevant by the feature selection process? Let us focus on the enhanced feature set. On average, 52% of the 77 features are maintained by the selection process (the linear kernel achieves competitive results with the fewest features). Still, only 27 features (35%) are considered relevant simultaneously by the three kernel configurations: 10 (13%) from the surface and back-translation sets and 17 (22%) from the noise- and similarity-based sets. Particularly, 13 out of the 20 features based on character 3-grams and cognateness similarity (cf. (v) and (w) in Section 3) are always maintained. In contrast, only 2 out of 8 features based on Levenshtein distance, are always kept. In agreement with previous results [Pighin *et al.*, 2012], back-translations (i.e., *bt* and *bf*) are found relevant, since they are mostly retained. On the other side, only 6 features were considered useless in all the cases, including those that look at tokens longer than 15 words and a couple of similarities between texts in the same language.

Table 4: Results over the unseen test partition by considering the features obtained from the enhanced set with look-ahead.

kernel	$Acc.$	F_1	$Prec.$	$Rec.$
linear	69.9	73.2	65.2	83.3
poly	71.2	75.3	65.3	88.9
RBF	72.6	75.6	67.4	86.1

We used the best kernel configurations and feature sets to train three new SVMs and classify the test partition from the FFF$^+$ corpus. As can be seen in Table 4, the obtained results are very similar to those of the cross-validation (Table 2). This performance is slightly better than the figures reported by Pighin *et al.* [2012] with the original FFF corpus: $A = 63.9$, $P = 64.3$, and $R = 85.3$. Nevertheless, these figures are not directly comparable: the current training dataset is bigger and some of the instances in common between FFF and FFF$^+$ were switched to a different class after the annotation revision (cf. Section 2.2). Indeed, preliminary experiments, not reported here, showed that the new annotation made the classification of the FFF corpus slightly harder. Additionally, the test set contains only a few dozen instances and small variations might cause drastic shifts. Rather than delving further into the comparison on the feedback classification task, we believe that it is more relevant to evaluate the utility of the output of these classifiers at improving the quality of a translation system.

5 Impact on an SMT System

In this section we present our preliminary experiments on the integration of automatically selected feedback in a real MT system. For that, we used a set of 6.6K fresh feedback instances, collected again from the Reverso.net weblogs. These instances were automatically classified with the models trained on the whole FFF$^+$ corpus, and used to enrich the translation model of a previously existing English-to-Spanish SMT system. This translator is a factored MOSES phrase-based system [Koehn and Hoang, 2007] from words into words and POS-tags[4], trained with corpora from WMT'12[5]. We consider two variants of the baseline: (*a*) 'Base_News', tuned on News bilingual corpora distributed by WMT'12, and (*b*) 'Base_Faust', which is adapted to the noisy scenario of on-line translation, with open-domain requests. This adaptation is made with both the Faust (development) corpus[6], for parameter tuning, and additional language models built from Reverso.net Spanish to English translation requests.

[4] Extracted with the Freeling suite [Padró *et al.*, 2010].

[5] `http://www.statmt.org/wmt12`

[6] Available at `ftp://mi.eng.cam.ac.uk/data/faust/FAUST-1.0.tgz`, this corpus, independent from the FFF$^+$, is composed of actual translation requests from Reverso.net, accompanied by manual references. It is split into development and test sets with slightly over one thousand sentences each.

Table 5: Results obtained by all baseline and feedback-enriched SMT systems on the Faust test set.

	BLEU	NIST	TER	MTR	ULC
Faust Raw					
Base_News	32.86	7.98	53.39	53.30	38.87
Base_Faust	*34.47*	*8.28*	*51.76*	*54.87*	*42.54*
Faust+Feed (all)	34.41	8.27	51.15	55.32	42.93
Faust+Feed (class.)	34.85	8.39	50.61	55.82	44.05
Faust+Feed (50%)	**35.22**†	**8.41***	**50.19**	**55.83**	**44.59**
Faust Clean					
Base_News	36.75	8.40	48.71	56.66	39.29
Base_Faust	*38.64*	*8.68*	*46.91*	*58.42*	*42.91*
Faust+Feed (all)	38.61	8.64	46.80	58.60	42.79
Faust+Feed (class.)	39.17	8.76	45.74	59.29	44.28
Faust+Feed (50%)	**39.49***	**8.80***	**45.42**	**59.31**	**44.78**

Our approach to enrich the translation model from the baseline system (TM_1) is as follows (more details can be found in [Formiga *et al.*, 2012]). Word alignments between corresponding *trans* and *feed* are computed to uncover the potential translator's mistakes. Identical words in the two fragments are linked to each other and used as a pivot to identify the corresponding fragments in *src*. Mismatching fragments are used to generate new alignments between *src*'s and *feed*'s fragments. The resulting set of alignments between *src* and *feed* is used to estimate a new translation model TM_2. Its inclusion into TM_1 is performed in two different ways: (*i*) phrases in TM_2 not in TM_1 are added, and (*ii*) phrases in TM_2 already in TM_1 are promoted. To increase robustness, mistaken translated fragments from *src* to *trans* cause the likelihood of related alignments in TM_1 to decrease.

Table 5 presents the results obtained on the Faust test set with the two baseline and the feedback-enriched MT systems. We limit the analysis of the enriched systems to the incorporation of feedback instances ranked by the linear classifier.[7] Three cases are considered: (*a*) 'Faust+Feed (all)', in which every instance of the corpus is added without filtering; (*b*) 'Faust+Feed (class.)', where we add exactly the instances predicted as useful by the classifier; and (*c*) 'Faust+Feed (50%)', where we add the 50% top scored examples by the classifier. The two available versions of the Faust test set were also considered. In 'Faust Raw' the source text comes "as is", whereas 'Faust Clean' incorporates some manual corrections to the source (misspellings, typos, slang, etc.).

We applied standard evaluation metrics for MT: BLEU, NIST, METEOR, and TER [Papineni *et al.*, 2002; Nis, 2002; Snover *et al.*, 2009; Denkowski and Lavie, 2011]. The last column includes ULC: a linear combination of several variants of the preceding four metrics, provided by the Asiya suite for MT evaluation [Giménez and Màrquez, 2010].[8] The best results in every test set are boldfaced.

Several conclusions can be drawn from Table 5. Firstly, 'Base_Faust' is a strong baseline, since it performs significantly better than the baseline translator optimized over standard text. We take 'Base_Faust' as the reference for the following comparisons. Secondly, 'Faust+Feed (all)' is unable to improve performance, showing that one cannot simply add all feedback instances collected without any filtering of the noisy cases. Instead, selecting a subset of instances according to the classifiers ('class.' or '50%') improves the results over Base_Faust consistently across all the evaluation metrics. The quality–quantity tradeoff appears to be an important aspect when enriching the baseline system. The top-50% cut obtains the best results in all metrics. This is slightly more conservative (so more precision oriented) than selecting the instances classified positively, which amount to 61.2% in this case. Empirical analysis of the optimal selection thresholds should be studied in future work. Finally, statistical tests performed using bootstrapping [Riezler and Maxwell, 2005] indicate that BLEU and NIST scores for 'Faust+Feed (50%)' are better than those of 'Base_Faust' with high probability in both versions of the test set ('$*$' and '$\dagger$' in the table indicate confidence levels of 0.99 and 0.95, respectively).

6 Conclusions and Future Work

In this paper we approached the automatic identification of useful translation corrections provided by users of an on-line MT service as a supervised classification task. In order to do so, we extended a previously existing corpus to more than 500 manually annotated instances and applied supervised learning (SVMs) to train classifiers. More than 70 features were used to characterize the text fragments and their inter-relations, requiring no external resources. Our tuning strategy, based on parameter adjustment and feature selection, showed that, regardless of the kernel, features aimed at measuring mono- and cross-language similarity between the fragments are particularly useful. The resulting classifiers showed precision levels in the 70-80% range. Regardless of this modest performance, we showed that an appropriate inclusion of the selected feedback into a state-of-the-art statistical machine translation system can significantly improve its performance. A recent manual analysis (not included here) revealed that the improvement is not only due to an enriched vocabulary, but mainly comes from better morphology, agreement and reordering.

Our ongoing work consists of considering more sophisticated features, including statistical language models, mono and bilingual dictionaries, and various quality estimation (QE) measures from the MT community (e.g., [Specia *et al.*, 2010; Callison-Burch *et al.*, 2012]). Our preliminary results using the Asiya QE measures as new features seem promising. Another research path consists of using a combination of classifiers in order to achieve higher levels of precision. Finally, the main direction for future work will be an in-depth study of the integration and impact of the selected feedback into the translation system. Among others, we should analyze: (*a*) the relation between quality and quantity of selected feedback instances and its influence in translation improvement and (*b*) the qualitative impact on translation —new vocabulary, new constructions, adaptation to input noise, potential degradation on other domains, etc.

[7] Experiments with the polynomial and RBF kernel-based classifiers did not improve over the results of the linear kernel. Pending a further study, this fact could be explained by the ability of linear models to achieve higher levels of precision (cf. Figure 1).

[8] `http://nlp.lsi.upc.edu/asiya/`

References

[Broder, 1997] Andrei Z. Broder. On the Resemblance and Containment of Documents. In *Compression and Complexity of Sequences (SEQUENCES'97)*, pages 21–29, Salerno, Italy, 1997. IEEE Computer Society.

[Callison-Burch *et al.*, 2012] Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. Findings of the 2012 Workshop on Statistical Machine Translation. In *Proceedings of the 7th Workshop on Statistical Machine Translation*, pages 10–51, Montréal, Canada, 2012.

[Cohen, 1960] Jacob Cohen. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

[Denkowski and Lavie, 2011] Michael Denkowski and Alon Lavie. Meteor 1.3: Automatic Metric for Reliable Optimization and Evaluation of Machine Translation Systems. In *Proceedings of the 6th Workshop on Statistical Machine Translation*, pages 85–91, Edinburgh, Scotland, 2011.

[Formiga *et al.*, 2012] Lluis Formiga, Carlos A. Henríquez Q., Adolfo Hernández, José B. Mariño, Enric Monte, and José A. R. Fonollosa. The TALP-UPC Phrase-based Translation Systems for WMT12: Morphology Simplification and Domain Adaptation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 275–282, Montréal, Canada, 2012. Association for Computational Linguistics.

[Gale and Church, 1993] William A. Gale and Kenneth, W. Church. A Program for Aligning Sentences in Bilingual Corpora. *Computational Linguistics*, 19:75–102, 1993.

[Giménez and Màrquez, 2010] Jesús Giménez and Lluís Màrquez. Asiya: An Open Toolkit for Automatic Machine Translation (Meta-)Evaluation. *The Prague Bulletin of Mathematical Linguistics*, (94):77–86, 2010.

[Hsu *et al.*, 2003] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University, 2003.

[Joachims, 1999] Thorsten Joachims. *Advances in Kernel Methods – Support Vector Learning*, chapter Making large-Scale SVM Learning Practical. MIT Press, 1999.

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

[McNamee and Mayfield, 2004] Paul McNamee and James Mayfield. Character N-Gram Tokenization for European Language Text Retrieval. *Information Retrieval*, 7(1-2):73–97, 2004.

[Nis, 2002] Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics. Technical report, National Institute of Standards and Technology, 2002.

[Padró *et al.*, 2010] Lluìs Padró, Miquel Collado, Samuel Reese, Marina Lloberes, and Irene Castellón. 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, 2010.

[Papineni *et al.*, 2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002.

[Pighin *et al.*, 2012] Daniele Pighin, Lluís Màrquez, and Jonathan May. 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, 2012.

[Potthast *et al.*, 2011] Martin Potthast, Alberto Barrón-Cedeño, Benno Stein, and Paolo Rosso. Cross-Language Plagiarism Detection. *Language Resources and Evaluation (LRE), Special Issue on Plagiarism and Authorship Analysis*, 45(1):1–18, 2011.

[Pouliquen *et al.*, 2003] Bruno Pouliquen, Ralf Steinberger, and Camelia Ignat. 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)*, pages 401–408, Borovets, Bulgaria, 2003.

[Riezler and Maxwell, 2005] Stefan Riezler and John .T. Maxwell. On some pitfalls in automatic evaluation and significance testing for MT. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 57–64, 2005.

[Simard *et al.*, 1992] Michel Simard, George F. Foster, and Pierre Isabelle. Using Cognates to Align Sentences in Bilingual Corpora. In *Proceedings of the 4th International Conference on Theoretical and Methodological Issues in Machine Translation*, 1992.

[Snover *et al.*, 2009] Matthew G. Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. TER-Plus: Paraphrase, Semantic, and Alignment Enhancements to Translation Edit Rate. *Machine Translation*, 23(2):117–127, 2009.

[Specia *et al.*, 2010] Lucia Specia, Dhwaj Raj, and Marco Turchi. Machine Translation Evaluation Versus Quality Estimation. *Machine Translation*, 24:39–50, March 2010.

[Vapnik, 1995] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, 1995.

[Witten and Frank, 2005] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, 2 edition, 2005.