Coping with the Subjectivity of Human Judgements in MT Quality Estimation

Marco Turchi **Matteo Negri** **Marcello Federico**
Fondazione Bruno Kessler, FBK-irst
Trento , Italy
{turchi|negri|federico}@fbk.eu

Abstract

Supervised approaches to NLP tasks rely on high-quality data annotations, which typically result from expensive manual labelling procedures. For some tasks, however, the subjectivity of human judgements might reduce the usefulness of the annotation for real-world applications. In Machine Translation (MT) Quality Estimation (QE), for instance, using human-annotated data to train a binary classifier that discriminates between *good* (useful for a post-editor) and *bad* translations is not trivial. Focusing on this binary task, we show that subjective human judgements can be effectively replaced with an automatic annotation procedure. To this aim, we compare binary classifiers trained on different data: the human-annotated dataset from the 7^{th} Workshop on Statistical Machine Translation (WMT-12), and an automatically labelled version of the same corpus. Our results show that human labels are less suitable for the task.

1 Introduction

With the steady progress in the field of Statistical Machine Translation (SMT), the translation industry is now faced with the possibility of significant productivity increases (*i.e.* amount of publishable output per unit of time). One way to achieve this goal, in Computer Assisted Translation (CAT) environments, is the integration of (*precise, but often partial*) suggestions obtained through "fuzzy matches" from a Translation Memory (TM), with (*complete, but potentially less precise*) translations produced by an MT system. Such integration can loosely consist in presenting translators with unranked suggestions obtained from the MT and the TM, or rely on tighter combination strategies. For instance, MT and TM translations can be automatically ranked to ease the selection of the most suitable one for post-editing (He et al., 2010), or the TM can be used to constrain and improve MT suggestions (Ma et al., 2011). In all cases, the effectiveness of the integration is conditioned by: *i)* the quality of MT, and *ii)* the accuracy in automatically predicting such quality. Higher productivity increases depend on the capability of the MT system to output *useful* material that is close to be publishable "as is" (Denkowski and Lavie, 2012), and the capability to automatically identify and present to human translators only such suggestions.

Recognizing good translations falls in the scope of research on automatic MT Quality Estimation (QE), which addresses the problem of estimating the quality of a translated sentence at run-time, without access to reference translations (Specia et al., 2009; Soricut and Echihabi, 2010; Bach et al., 2011; Specia, 2011; Mehdad et al., 2012b). In recent years QE gained increasing interest in the MT community, resulting in several datasets available for training and evaluation (Callison-Burch et al., 2012), the definition of features showing good correlation with human judgements (Soricut et al., 2012), and the release of open-source software.[1]

The proposed solutions to the QE problem rely on supervised methods that strongly depend on the availability of labelled data. While early works (Blatz et al., 2003) exploited annotations obtained with automatic MT evaluation metrics like BLEU (Papineni et al., 2002), the current trend is to rely on human annotations, which seem to lead to more accurate models (Quirk, 2004; Specia et al., 2009). Along this direction, the QE task consists in predicting scores that reflect human quality judgements, by learning from manually annotated datasets (*e.g.* collections of *source-target* pairs la-

[1] http://www.quest.dcs.shef.ac.uk/

Proceedings of the Eighth Workshop on Statistical Machine Translation, pages 240–251,
Sofia, Bulgaria, August 8-9, 2013 ©2013 Association for Computational Linguistics

belled according to an n-point Likert scale or with real numbers in a given interval). Within this dominant supervised framework, **we explore different ways to obtain labelled data for training a binary QE classifier suitable for integration in a CAT tool**. Since, to the best of our knowledge, labelled data with binary judgements are currently not available, we consider two alternative options.

The first option is to adapt an existing dataset, checking whether it can be partitioned in a way that reflects the distinction between *good* (useful for the translator, suitable for post editing) and *bad* translations (that need complete rewriting).[2] To this aim we experiment with the QE data released within the 7^{th} Workshop on Machine Translation (WMT-12). The corpus consists of *source-target* pairs annotated with manual QE labels (1-5 scores) indicating the post-editing needed to correct the translations. Besides explicit human judgements, the availability of post-edited translations makes also possible to calculate the actual HTER values (Snover et al., 2009), indicating the minimum edit distance between the machine translation and its manually post-edited version in the [0,1] interval.

The second option is to automatically re-annotate the same dataset, trying to produce labels that reflect an objective and more reliable binary distinction based on empirical observations.

Our analysis aims to answer the following questions:

1. Are human labels reliable and coherent enough to train accurate binary models?
2. Are arbitrarily-set thresholds useful to partition QE data for this task?
3. Is it possible to obtain reliable binary annotations from an automatic procedure?

Negative answers to the first two questions would respectively call into question: *i)* the intuitive idea that human labels are the most reliable for a supervised approach to binary QE, and *ii)* the possibility that thresholds on a single metric (*e.g.* the HTER) can be set to capture the subtle differences separating useful from useless translations. A positive answer to the third question would open to the possibility to create training datasets in a more coherent and replicable way compared to current data annotation methods. By answering these questions, this paper provides the following main contributions:

- We show that training a binary classifier on arbitrary partitions of an existing dataset is difficult. Our experiments with the WMT-12 corpus demonstrate that neither following standard indications (*e.g.* "*if more than 70% of the MT output needs to be edited, a translation from scratch is necessary*")[3], nor considering arbitrary HTER thresholds, it is possible to obtain accurate binary classifiers suitable for integration in a CAT environment;
- We propose a replicable automatic (hence non subjective) method to re-annotate an existing dataset in a way that the resulting binary classifier outperforms those trained with human labels.
- We show that, with our method, a smaller amount of training data is sufficient to obtain similar or better performance compared to that of the human-annotated dataset used for comparison.

2 Binary QE for CAT environments

QE has been mainly addressed as a classification or regression task, where a quality score (respectively an integer or a real value) has to be automatically assigned to MT output sentences given their source (Specia et al., 2010). Casting the problem in this way, the integration of a QE component in a CAT environment makes possible to present translators with estimates of the expected quality of each MT suggestion. Such intuitive solution, however, disregards the fact that even precise QE scores would not alleviate translators from the effort of reading useless MT output (or at least the associated score).

A more effective alternative is to use the estimated QE scores to filter out poor MT suggestions, presenting only those worth for post-editing. Binary classification, however, has to confront with the problem of setting reasonable cut-off criteria. The arbitrary thresholds, used in several previous works (Quirk, 2004; Specia et al., 2010; Specia et al., 2011) are in fact hard to justify, and even harder to learn from human-labelled training data.

[2] In the remainder of the paper we will consider as "good" translations those for which post-editing requires a smaller effort than translation from scratch. Conversely, we will label as "bad" the translations that need complete rewriting.

[3] This was a guideline for the professional translators involved in the annotation of a previous version of the dataset used for the WMT-12 evaluation (see `http://www.statmt.org/wmt12/quality-estimation-task.html`).

On one side, for instance, there is no evidence that the 70% HTER threshold used in some datasets yields the optimal separation between acceptable and totally useless suggestions. Such arbitrary criterion, based on the raw count of post-editing operations, is likely to reflect a partial view on a complex problem, disregarding important aspects such as the distribution of the corrections in the MT output. However, in some cases, having the first 30% of words correctly translated might take less post-editing effort than having 50% of correctly translated terms scattered throughout the whole sentence. In these cases, a 70% HTER threshold would wrongly consider useless translations as positive instances and vice-versa.

On the other side, when arbitrary thresholds are used as annotation guidelines (Callison-Burch et al., 2012), the moderate agreement between human judges might make manual labels ill-suited to learn accurate models.

Under the constraints posed by a CAT environment, where only useful suggestions can lead to a significant productivity increase, the ideal model should maximize the number of true positives (useful translations recognized as good) minimizing, at the same time, the number of false positives (useless translations recognized as good). To this aim, the more the training data are partitioned according to objective criteria, the higher the expected reliability of the corresponding cut-off and, in turn, the higher the expected performance of the binary classifier.

Focusing on these issues, the following sections discuss various methods to obtain training data for binary QE geared to the integration in a CAT environment. Partitions based on human judgements from the WMT-12 dataset will be compared with an automatic method to re-annotate the same corpus. The suitability of the resulting training sets for binary classification will be assessed by measuring the performance of classifiers built from each training set. Metrics sensitive to the number of false positives will be used for this purpose.

3 Partitioning the WMT-12 dataset

Due to the lack of datasets annotated with explicit binary (good, bad) judgements about translation quality, the most intuitive way to obtain training data for our QE classifier is to adapt existing manually-labelled data. The reasonable size of the WMT-12 dataset makes it a good candidate for our purposes. The corpus consists of 2,254 English-Spanish news sentences (1,832 for training, 422 for test) produced by the Moses phrase-based SMT system (Koehn et al., 2007) trained on Europarl (Koehn, 2005) and News Commentaries corpora,[4] along with their source sentences, reference translations and post-edited translations. Training and test instances have been annotated by professional translators with scores (1 to 5) indicating the estimated post-editing effort (percentage of MT output that has to be corrected). According to the proposed scheme, the highest score indicates lowest effort (MT output requires little or no editing), while the lowest score indicates that the MT output needs to be translated from scratch. To cope with systematic biases among the annotators,[5] the judgements were combined in a final score obtained from their weighted average, resulting in a labelled dataset with real numbers in the [1, 5] interval as effort scores.

In order to obtain suitable data for binary QE, the WMT-12 training set (1,832 instances) has been partitioned in different ways, leaving the test set for evaluation (see Section 5). The goal, for each partition strategy, was to label as *bad* (the assigned label is -1) only the translations that *need complete rewriting*, keeping all the other translations as *good* instances (labelled with +1). Considering the averaged effort scores, the actual human judgements, and the HTER values calculated between the translations and the corresponding post-edited version, we experimented with the following three partition criteria.

Average effort scores (AES). Three partitions have been generated based on the effort scores of *2*, *2.5*, and *3*, labelling the WMT-12 training instances with scores below or equal to each threshold as negative examples (-1), and the instances with scores above the threshold as positive examples (+1). Partitions with thresholds below *2* were also considered, including the most intuitive partition with cut-off set to *1*. However, the resulting number of negative instances, if any, was too scarce, and the overall dataset too unbalanced, to make standard supervised learning methods effective The creation of highly unbalanced data is a recurring issue for all the partition meth-

[4] http://www.statmt.org/wmt11/translation-task.html#download

[5] Such biases support the idea that labelling translations with quality scores is *per se* a highly subjective task.

ods we applied to the WMT-12 corpus. Together with the low homogeneity of human labels (even for very poor translations the three judges do not agree in assigning the lowest score), in most of the cases the small number of low-quality translations in the dataset makes the negative class considerably smaller than the positive one. This can be observed in Table 1, which provides the total number of positive and negative instances for each partition method. For instance, with our lowest AES threshold (*2*) the total number of negative instances is 113, while the positive ones are 1,719. Although considering different cut-off criteria aims to make our investigation more complete, it's also worth remarking that the higher the threshold, the higher the distance of the resulting experimental setting from our target scenario. While *2*, as an effort score threshold, is likely to reflect a reasonable separation between useless and post-editable translations, higher values are in principle more appropriate for "soft" separations into worse *versus* better translations.

Human scores (HS). Five partitions have been generated using the actual labels assigned by the three annotators to each translation instead of the average effort scores. In particular, we considered the following score combinations ("X" stands for any integer between 1 and 5): *1-X-X*, *2-2-2*, *2-2-X*, *2-3-3*, *3-3-3*. Also in this case, as shown in Table 1, partitions based on lower scores lead to highly unbalanced datasets of limited usability, while those based on higher scores are increasingly more distant to our application scenario.[6]

HTER scores (HTER). Seven partitions have been generated considering the following HTER thresholds: *0.75*, *0.7*, *0.65*, *0.6*, *0.55*, *0.5*, *0.45*. In this case, being the HTER an error measure, training instances with scores above or equal to the threshold were labelled as negative examples (-1), while instances with lower scores were labelled as positive examples (+1). Similar to the other partition criteria, some of our threshold values reflect our task more closely than others, but result in more unbalanced datasets. In particular, thresholds around *0.7* substantially adhere to the WMT-12 annotation guidelines (as far as translations that need complete rewriting are concerned) and produce training data with fewer negative instances. Other thresholds, which is still worth exploring since we do not know the optimal cut-off value, are in principle less suitable to our task but produce more balanced training data.

	Training instances	
Average effort scores (AES)	*Positive*	*Negative*
2	1,719	113
2.5	1,475	357
3	1,194	638
Human scores (HS)	*Positive*	*Negative*
1-X-X	1,736	96
2-2-2	1,719	113
2-2-X	1,612	220
2-3-3	1,457	375
3-3-3	1,360	472
HTER scores (HTER)	*Positive*	*Negative*
0.75	1,798	34
0.7	1,786	46
0.65	1,756	76
0.6	1,708	124
0.55	1,653	179
0.5	1,531	301
0.45	1,420	412

Table 1: Number of positive/negative instances for each partition of the WMT-12 training set.

4 Re-annotating the WMT-12 dataset

As an alternative to partitioning methods, we investigated the possibility to re-annotate the WMT-12 training set with an automatic procedure.

4.1 Approach

Our approach, which does not involve subjective human judgements, is based on the observation of similarities and dissimilarities between an automatic translation (TGT), its post-edited version (PE) and the corresponding reference translation (RT). Such comparisons provide useful indications about the behaviour of a post-editor when correcting automatic translations and, in turn, about MT output quality.

Typically, the PE version of a good-quality TGT preserves some characteristics (*e.g.* lexical, structural) that indicate a moderate correction activity by the post editor. Conversely, in the PE version of a low-quality TGT, such characteristics are more difficult to observe, indicating an intense correction activity. At the two extremes, the PE of a perfect TGT preserves all its characteristics, while the PE of a useless TGT looses most of them. In the first case TGT and PE are iden-

[6] The partition most closely related to our task (*i.e.* 1-1-1) was impossible to produce since none of the examples was labelled with *1* by all the annotators. Even for *1-1-X*, the negative class contains only one example.

tical, and their similarity is the highest possible (*i.e.* $sim(TGT, PE) = 1$). In the second case, TGT and PE show a degree of similarity close to that of TGT and a completely rewritten translation featuring different lexical choices and structure. This is where reference translations come into play: considering RT as a good example of rewritten sentence,[7] for low-quality TGT we will have $sim(TGT, PE) \approx sim(TGT, RT)$.

In light of these considerations, we hypothesize that the automatic re-annotation of WMT-12 training data can take advantage of a classifier that learns a similarity threshold T such that:

- a PE sentence with $sim(TGT, PE) \leq T$ will be considered as a rewritten translation (hence TGT is useless, and the corresponding *source-TGT* pair a negative example to be labelled as "-1");
- a PE sentence with $sim(TGT, PE) > T$ will be considered as a real post-edition (hence TGT is useful for the post-editor, and the corresponding *source-TGT* pair a positive example to be labelled as "+1").

Based on this hypothesis, to perform our automatic re-annotation procedure we: 1) create a training set Z of positive and negative examples (*i.e.* [TGT, *correct_translation*] pairs, where *correct_translation* is either a post-editing or a rewritten translation); 2) design a feature set capable to capture different aspects of the similarity between TGT and *correct_translation*; 3) build a binary classifier using Z; 4) use the classifier to label the [TGT, PE] pairs as instances of post-editings or rewritings; 5) assess the quality of the resulting annotation.

4.2 Building the classifier

Training corpus. To build a classifier capable of labelling PE sentences as rewritten/post-edited material, we first created a set of positive and negative instances from the WMT-12 training set. For each tuple [source, TGT, PE, RT] of the dataset, one positive and one negative instance have been respectively obtained as the combination of [TGT, PE] and [TGT, RT]. Figure 1, which plots the distribution of positive and negative instances against HTER, shows a fairly good separation between the two classes. This indicates that our use of the references as examples of rewritten translations builds on a reasonable assumption.

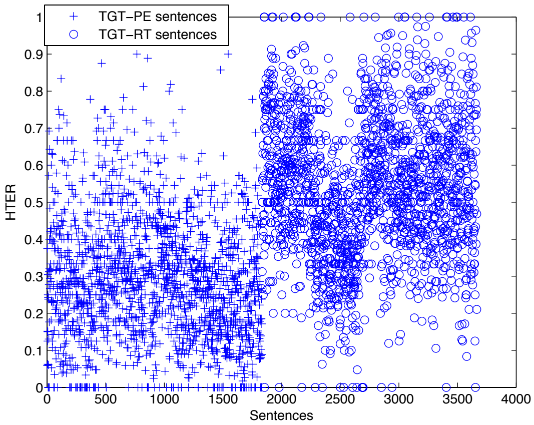

Figure 1: Distribution of [TGT, PE] and [TGT, RT] pairs plotted against the HTER.

Features. Crucial to our classification task, a number of features can be used to estimate sentence similarity. Differently from the binary QE task, where the possibility to catch common characteristics between two sentences is limited by language barriers, in our re-annotation task all the features are extracted by comparing two monolingual sentences (*i.e.* TGT and a *correct_translation*, either a PE or a RT). Although the problem of measuring sentence similarity can be addressed in many ways, the solutions should not overlook the specificities of the task. In our case, for instance, the scarce importance of the semantic aspect (TGT, PE and RT typically show a high semantic similarity) makes features used for other tasks (*e.g.* based on distributional similarity) less effective than shallow features looking at the surface form of the input sentences. Our problem presents some similarities with the plagiarism detection task, where subtle lexical and structural similarities have to be identified to spot suspicious plagiarized texts (Potthast et al., 2010). For this reason, part of our features (*e.g.* ROUGE scores) are inspired by research in such field (Chen et al., 2010), while others have been designed *ad-hoc*, based on the specific requirements of our task. The resulting feature set aims to capture text similarity by measuring word/n-gram matches, as well as the level of sparsity and density of the common words as a shallow indicator of structural similarity. In total, from each [TGT, *correct_translation*]

[7] Such assumption is supported by the fact that reference sentences are, by definition, free translations manually produced without any influence from the target.

pair, the following 22 features are extracted:

- Human-targeted Translation Error Rate – HTER. The editing operations considered are: shift, insertion, substitution and deletion.
- Number of words in common.
- Number of words in common, normalized by TGT length and *correct_translation* length (2 features).
- Number of words in TGT and in the *correct_translation* (2 features).
- Size of the longest common subsequence.
- Size of the longest common subsequence, normalized by TGT length.
- Aligned word density: total number of aligned words,[8] divided by the number of aligned blocks (more than 1 aligned word).
- Unaligned word density: total number of unaligned words, divided by the number of unaligned blocks (more than 1 unaligned word).
- Normalized number of aligned blocks: total number of aligned blocks, divided by TGT length.
- Normalized number of unaligned blocks: total number of unaligned blocks, divided by TGT length.
- Normalized density difference: difference between aligned word density and unaligned word density, divided by TGT length.
- Modified Lesk score (Lesk, 1986): sum of the squares of the length of n-gram matches, normalized by the product of the sentence lengths.
- ROUGE-1/2/3/4: n-gram recall with n=1,...,4 (4 features).[9]
- ROUGE-L: size of longest common subsequence, normalized by the *correct_translation* length.
- ROUGE-W: the ROUGE-L using different weights for consecutive matches of length L (default weight = *1.2*).
- ROUGE-S: the ROUGE-L allowing for the presence of skip-bigrams (pairs of words, even not adjacent, in their sentence order).
- ROUGE-SU: the extension of ROUGE-S adding unigrams as counting unit.

To increase the capability of identifying similar sentences, all sentences are tokenized, lowercased and stemmed using the Snowball algorithm (Porter, 2001).

Classifier. On the resulting corpus, an SVM classifier has been trained using the LIBSVM toolbox (Chang and Lin, 2011). The selection of the kernel (linear) and the optimization of the parameters (C=0.8) were carried out through grid search in 5-fold cross-validation.

Labelling the dataset. Using the best parameter setting obtained, [TGT, PE] and [TGT, RT] pairs have been re-labelled as post-editings or rewritings through 5 rounds of cross-validation. The final label of each instance was set to the mode of the predictions produced by each cross-validation round. Since we assume that the quality of the target sentence can be inferred from the amount of correction activity done by the post-editor, the labels assigned to the [TGT, PE] pairs represent the result of our re-annotation of the corpus into positive and negative instances.

At the end of the process, of the 1,832 [TGT, PE] pairs of the WMT 2012 training set, 1.394 are labelled as examples of post-editing (TGT is useful), and 438 as examples of complete rewriting (TGT is useless). Compared to the distribution of positive and negative instances obtained with most of the partition methods described in Section 3, our automatic annotation produces a fairly balanced dataset. The resulting proportion of negative examples (∼1:3) is similar to what could be reached only by partitions reflecting a "soft" separation into worse *versus* better translations rather than a strict separation into useless *versus* useful translations.[10] In Figure 2, the labelling results plotted against the HTER show that there is a quite clear separation between [TGT, PE] pairs marked as post-editings (lower HTER values) and pairs marked as rewritings (higher HTER values). Such separation corresponds to an HTER value around *0.4*, which is significantly lower than the threshold of *0.7* proposed by the WMT-12 guidelines as a criterion to label sentences for which "*a translation from scratch is necessary*". This confirms that our separation differs from those produced by partition methods based on human annotations or arbitrary HTER thresholds. Furthermore, our au-

[8] Monolingual stem-to-stem exact matches between TGT and *correct_translation* are inferred by computing the HTER, as in (Blain et al., 2012).

[9] All ROUGE scores, described in (Lin, 2004), have been calculated using the software available at `http://www.berouge.com`.

[10] Such partitions are: average effort scores = 3, human scores = 3-3-3, HTER score = 0.45.

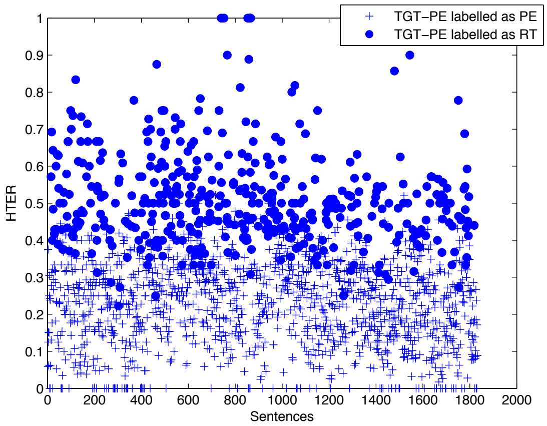

Figure 2: TGT-PE classification in post-editings and rewritings.

tomatic annotation procedure relies on the contribution of features designed to capture different aspects of the similarity between the TGT and a *correct_translation*, while some of the partition methods discussed in Section 3 rely on thresholds set on a single score (*e.g.* HTER). Considering the many facets of the binary QE problem, we expect that our features are more effective to deal with latent aspects disregarded by such thresholds.

5 Experiments and results

At this point, the question is: *are the automatically labelled data more suitable than partitions based on human labels to train a binary QE classifier?* To answer this question, all the proposed separations of the WMT-12 training set have been evaluated on different test sets. For each separation we trained a binary classifier able to assign a label (good or bad) to unseen *source-target* pairs. Since the classifiers use the same algorithm and feature set, differences in performance will mainly depend on the quality of the training data on which they are built. Using task-oriented metrics sensitive to the number of false positives, results highlighting such differences will indicate the best separation.

5.1 Experimental Setting

Binary QE classifier. Each separation of the WMT-12 training data was used to train a binary SVM classifier. Different kernels and parameters were optimized through a grid search in 5-fold cross-validation on each training set. Being the number of positive and negative training instances highly unbalanced, the best models were selected optimizing a metric that takes into account the number of true and false positives (see below).

Seventeen features proposed in (Specia et al., 2009) were extracted from each *source-target* pair. This feature set, fully described in (Callison-Burch et al., 2012), mainly takes into account the complexity of the source sentence (*e.g.* number of tokens, number of translations per source word) and the fluency of the target translation (*e.g.* language model probabilities). Results of the WMT 2012 QE task shown that these "baseline" features are particularly competitive in the regression task, with only few systems able to beat them. All the features are extracted using the Quest software[11] and the model files released by the organizers of the WMT 2013 workshop.

Test sets. To obtain different separations between good and bad translations, artificial test sets have been created using arbitrary thresholds on the HTER (the same used to partition the training set on a HTER basis) and the post-editing time (PET).[12] Two different datasets were split: *i*) the WMT-12 test (422 source, target, post-edited and reference sentences); *ii*) the WMT-13 training set for Task 1.3 (800 source, target and post-edited sentences labelled with PET). The first dataset, the most similar to the WMT-12 training set, should better reflect (and reward) the HTER-based partitions proposed in Section 3. The WMT-13 dataset contains sentences translated with a different configuration (data and parameters) of the SMT engine. This can result in different HTER-based partitions in good and bad, useful to test the portability of our automatic re-annotation method across different datasets. Finally, testing on data partitions based on PET allows us to check the stability of the automatic re-annotation method when evaluated on a test set divided according to a different concept of translation quality. In the end, the combination of different partition methods, thresholds and datasets results in 21 different test sets (see Table 2).

Evaluation metrics. F-score and accuracy are the classic evaluation metrics used in classification. In our evaluation, however, they would always result in high uninformative values due to the unbalanced nature of the test sets (positive instances $\gg$ negative instances). In order to bet-

[11] `http://www.quest.dcs.shef.ac.uk/`

[12] PET is the time spent by a post-editor to transform the target into a publishable sentence.

	Test instances	
WMT-12 HTER	*Positive*	*Negative*
0.45	289	133
0.5	319	103
0.55	352	70
0.6	371	51
0.65	386	36
0.70	398	24
0.75	406	16
WMT-13 Task 1.3 HTER	*Positive*	*Negative*
0.45	582	218
0.5	622	178
0.55	695	105
0.6	724	76
0.65	748	52
0.70	763	37
0.75	773	27
WMT-13 Task 1.3 PET	*Positive*	*Negative*
4	499	301
4.16*	517	283
4.50	554	246
5	594	206
6	659	141
7	698	102
8	727	73

Table 2: Number of positive and negative instances for each partition of the WMT-12 test set and WMT-13 training set. "*": Average PET computed on all the instances in the WMT-13 dataset.

ter understand the real quality of the classification, we hence opted for two task-oriented evaluation metrics sensitive to the number of false positives (the main issue in a CAT environment, where false positives and true positives should be respectively minimized and maximized). These are: *i)* the weighted combination of the false positive rate (FPR) and false discovery rate (FDR) (Benjamini and Hochberg, 1995), and *ii)* the weighed average of sensitivity and specificity (also called balanced/weighted accuracy). FPR measures the level of false positives, but does not provide information about the number of true positives. For this reason, we combined it with FDR (1-precision), which indirectly controls the level of true positives. FPR and FDR were equally weighted in the average; *lower values indicate good performance*. Furthermore, in our scenario it is desirable to have a classifier with high prediction accuracy over the minority class (specificity), while maintaining reasonable accuracy for the majority class (sensitivity). Weighted accuracy is useful in such situations. To better asses the performance on the minority (negative) class, we hence gave more importance to specificity (*0.7 vs 0.3*). As regards weighted accuracy *higher values in indicate better performance*. Penalizing majority voting classifiers, both metrics are particularly appropriate in our framework. Besides evaluation, the weighted average of FPR and FDR was also used to tune the parameters of the SVM classifier.

5.2 Results

Table 3 presents the results achieved by classifiers trained on different datasets, on the 21 splits produced from the test sets used for evaluation.

Although the total number of classifiers tested is 16 (15 resulting from partitions based on human labels, and 1 obtained with our automatic annotation method), most of them are not present in the table since they predict the majority class for all the test points. These are, in general, trained on highly unbalanced training sets where the number of negative samples is really small. However, it is interesting to note that increasing the number of instances in the negative class does not always result in a better classifier. For instance, the classifier built on an HTER separation with threshold at *0.55* performs majority voting even if it is built on a more balanced (but probably more noisy) training set than the classifier obtained with threshold at *0.6*. This suggests that the *quality* of the separation is as important as the actual proportion of positive and negative instances.

On all test sets, and for both the evaluation metrics used, the results achieved by the classifier built from the automatically annotated training set (*AA*) produces lower error rates (Weighted FPR-FDR) and higher accuracy (Weighted Accuracy), outperforming all the other classifiers. The effectiveness of the automatic annotation is confirmed by the fact that classifiers *3* (based on the average of effort scores - AES) and *3-3-3* (based on the actual human scores - HS), which are trained on more balanced training sets, achieve worse performances than the *AA* classifier.[13]

Results on the WMT-13 PET test set are not as good as in the other two test sets. This shows that test data labelled in terms of time are more difficult to be correctly classified compared to those based on the HTER. This can be explained considering the intrinsic differences between the HTER and the PET as approximations of the post-editing

[13] The distribution of positive/negative instances in the training sets is: 1194/638 for classifier *3*, 1360/472 for classifier *3-3-3*, 1394/438 for classifier *AA*.

Weighted FPR-FDR		Training: WMT-12 Separations						
		3 *AES*	*2-2-X* *HS*	*2-3-3* *HS*	*3-3-3* *HS*	*0.5* *HTER*	*0.6* *HTER*	*AA*
Test: WMT-12 HTER	*0.45*	0.61	0.66	0.66	0.66	0.66	0.66	**0.55**
	0.5	0.57	0.62	0.62	0.62	0.62	0.62	**0.49**
	0.55	0.52	0.58	0.58	0.58	0.58	0.58	**0.42**
	0.6	0.5	0.56	0.56	0.56	0.56	0.56	**0.4**
	0.65	0.5	0.54	0.54	0.54	0.54	0.54	**0.39**
	0.7	0.49	0.53	0.53	0.53	0.53	0.53	**0.39**
	0.75	0.49	0.52	0.52	0.52	0.52	0.52	**0.35**
Test: WMT-13 HTER	*0.45*	0.59	0.63	0.63	0.64	0.64	0.63	**0.54**
	0.5	0.57	0.6	0.6	0.61	0.61	0.6	**0.5**
	0.55	0.51	0.56	0.56	0.57	0.57	0.56	**0.41**
	0.6	0.49	0.54	0.54	0.55	0.55	0.54	**0.37**
	0.65	0.47	0.53	0.53	0.53	0.53	0.53	**0.33**
	0.7	0.44	0.52	0.52	0.52	0.52	0.52	**0.29**
	0.75	0.44	0.52	0.52	0.52	0.52	0.52	**0.28**
Test: WMT-13 PET	*4*	0.61	0.68	0.68	0.69	0.69	0.68	**0.58**
	4.16	0.61	0.67	0.67	0.67	0.67	0.67	**0.56**
	4.5	0.58	0.65	0.64	0.65	0.65	0.65	**0.54**
	5	0.55	0.63	0.62	0.63	0.63	0.62	**0.51**
	6	0.49	0.58	0.58	0.58	0.58	0.58	**0.45**
	7	0.45	0.55	0.55	0.56	0.56	0.55	**0.43**
	8	0.45	0.54	0.54	0.54	0.54	0.54	**0.41**

Weighted Accuracy		Training: WMT-12 Separations						
		3 *AES*	*2-2-X* *HS*	*2-3-3* *HS*	*3-3-3* *HS*	*0.5* *HTER*	*0.6* *HTER*	*AA*
Test: WMT-12 HTER	*0.45*	0.35	0.3	0.3	0.3	0.3	0.3	**0.41**
	0.5	0.35	0.3	0.3	0.3	0.3	0.3	**0.44**
	0.55	0.37	0.3	0.3	0.3	0.3	0.3	**0.48**
	0.6	0.37	0.3	0.3	0.3	0.3	0.3	**0.49**
	0.65	0.35	0.3	0.3	0.3	0.3	0.3	**0.47**
	0.7	0.35	0.3	0.3	0.3	0.3	0.3	**0.45**
	0.75	0.33	0.3	0.3	0.3	0.3	0.3	**0.49**
Test: WMT-13 HTER	*0.45*	0.33	0.31	0.31	0.3	0.3	0.31	**0.4**
	0.5	0.34	0.31	0.31	0.3	0.3	0.31	**0.42**
	0.55	0.35	0.31	0.31	0.3	0.3	0.31	**0.48**
	0.6	0.35	0.31	0.31	0.3	0.3	0.31	**0.51**
	0.65	0.36	0.3	0.3	0.3	0.3	0.3	**0.54**
	0.7	0.39	0.3	0.3	0.3	0.3	0.3	**0.56**
	0.75	0.38	0.3	0.3	0.3	0.3	0.3	**0.59**
Test: WMT-13 PET	*4*	0.37	0.3	0.31	0.3	0.3	0.3	**0.4**
	4.16	0.37	0.3	0.31	0.3	0.3	0.3	**0.4**
	4.5	0.37	0.3	0.31	0.3	0.3	0.3	**0.4**
	5	0.38	0.31	0.31	0.3	0.3	0.31	**0.41**
	6	0.41	0.31	0.31	0.3	0.3	0.31	**0.43**
	7	0.42	0.31	0.31	0.3	0.3	0.31	**0.44**
	8	0.4	0.31	0.31	0.3	0.3	0.31	**0.43**

Table 3: Weighted FPR-FDR (left table) and weighted Accuracy (right table) obtained by the binary QE classifiers trained on different separations of the WMT-12 training set. Several arbitrary partitions of the WMT-12 Test set and WMT-13 Training set are considered.

effort, as pointed out by several recent works (Specia, 2011; Koponen, 2012).

Comparing the results calculated with the two metrics, we note that weighted accuracy seems to be less sensible to small variations in terms of true and false negatives returned by the classifier, even if the specificity (accuracy on our minority class) is weighted more than sensitivity (accuracy on our majority class). This often results in scores very close (differences $\leq 10^{-3}$) to the accuracy obtained by majority voting classification (*0.3*).

Overall, our experiments demonstrate that the proposed automatic separation method is more effective than arbitrary partitions of datasets annotated with subjective human judgements.

5.3 Learning Curve

Our automatic re-annotation approach requires post-edited and reference sentences. Although all the datasets annotated for QE include post-edited sentences, this is not always true for the references. The cost of having both resources is in fact not negligible. For this reason, we investigated the minimal number of training data needed to re-annotate the WMT-12 training set without altering performance on binary classification. To this aim, we selected two of the test sets on which our re-annotation method produces classifiers with high performance results (*WMT-13 HTER 0.6* and *0.75*), and measured score variations with increasing amounts of data.

Nine subsets of the WMT-12 training set corpus were created (with 10%, 20%,..., 100% of the dataset) by sub-sampling sentences from a uniform distribution. The process was iterated 10 times. Then, for each subset, a new re-annotation process was run, the resulting training set was used to build the relative binary QE classifier, which was eventually evaluated on the test set in terms of weighted FPR-FDR. Figures 3 and 4 show the obtained learning curves. Each point is the average result of the 10 runs; the error bars show ±1std.

As can be seen from both curves, performance results with 60% of the training data are already comparable with those obtained using the whole training data. Similar trends have been observed for several learning curves created with different test sets. This shows that, besides avoiding the use of human labelled data, our approach allows to drastically reduce the amount of training instances. Considering the high costs of collecting post-editions, and the fact that reference translations can be taken from parallel corpora, our solution represents a viable way to overcome the lack of training data for binary QE geared towards integration in a CAT environment.

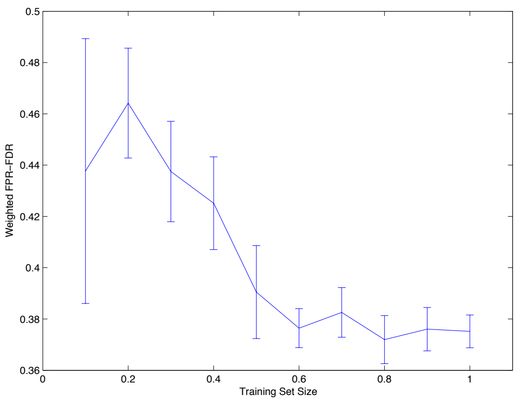

Figure 3: Learning curve for WMT-13 HTER *0.60*.

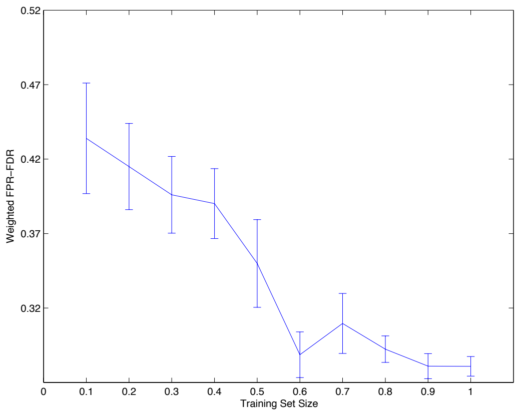

Figure 4: Learning curve for WMT-13 HTER *0.75*.

6 Conclusion

We presented a task-oriented analysis of the usefulness of human-labelled data for binary quality estimation. Our target scenario is computer-assisted translation, which calls for solutions to present human translators with *useful* MT suggestions (*i.e.* easier to correct than to rewrite from scratch). Within this framework, the integration of binary classifiers capable to distinguish "good" (useful) from "bad" (useless) suggestions would make possible to significantly increase translators' productivity. Such binary classifiers, however, need labelled training data (possibly of good quality) that are currently not available.

An intuitive solution to fill this gap is to take advantage of an existing dataset, adapting its manual annotations to our task. Exploring this solution (the first contribution of this paper) has to face problems related to the subjectivity of human judgements about translation quality, and the resulting variability in the annotation. In particular, our experiments with the WMT-12 dataset show that any adaptation (either based on human judgements or arbitrarily-set HTER thresholds) collides with the problem of setting reasonable partition criteria. Our results suggest that the subtle differences between useful and useless translations make subjective human judgements inadequate to learn effective models.

Instead of relying on manually-assigned quality labels, an alternative solution to the problem is to re-annotate an existing dataset. Proposing an automatic way to do that (the second contribution of this paper), we argue that reliable data separations into positive and negative examples can be obtained by measuring the similarities between: *i)* automatic translations and post-editings, and *ii)* automatic translations and their references. Our results demonstrate that binary classifiers built from training data produced with our supervised method are less prone to the misclassification of bad suggestions.

As in any supervised learning framework, the amount of data needed to obtain good results is of crucial importance. By analysing the demand of our automatic annotation method in terms of training data (the third contribution of this paper), we show that competitive results can be obtained with a fraction of the data needed by methods based on human labels. Our results indicate that a good-quality training set for binary classification can be obtained with 40% less instances of *[training, post_edited sentence, reference_sentence]*, totally avoiding manually-assigned quality judgements.

Our future works will address the improvement of the automatic annotation procedure using supervised methods suitable to learn from unbalanced training sets (*e.g.* one-class SVM, weighted random forests), and the integration of new features (*e.g.* GTM, meteor) to refine our classification of a *correct_sentence* into rewritten/post-edited. Then, to boost binary QE results on the resulting corpora, the "baseline" features used for experiments in this paper will be extended with new features explored in recent works (Mehdad et al., 2012a; de Souza et al., 2013; Turchi and Negri, 2013).

Acknowledgments

This work has been partially supported by the EC-funded project MateCat (ICT-2011.4.2-287688).

References

Nguyen Bach, Fei Huang, and Yaser Al-Onaizan. 2011. Goodness: a Method for Measuring Machine Translation Confidence. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 211–219. The Association for Computer Linguistics.

Yoav Benjamini and Yosef Hochberg. 1995. Controlling the False Discovery Rate: a Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 289–300.

Frédéric Blain, Holger Schwenk, and Jean Senellart. 2012. Incremental Adaptation Using Translation Information and Post-Editing Analysis. In *International Workshop on Spoken Language Translation*, pages 234–241, Hong-Kong (China).

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2003. Confidence Estimation for Machine Translation. Summer workshop final report, JHU/CLSP.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2012. Findings of the 2012 Workshop on Statistical Machine Translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation (WMT'12)*, pages 10–51, Montréal, Canada.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: a Library for Support Vector Machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27, May.

Chien-Ying Chen, Jen-Yuan Yeh, and Hao-Ren Ke. 2010. Plagiarism Detection using ROUGE and WordNet. *Journal of Computing*, 2(3).

José G. C. de Souza, Miquel Esplà-Gomis, Marco Turchi, and Matteo Negri. 2013. Exploiting Qualitative Information from Automatic Word Alignment for Cross-lingual NLP Tasks. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013).*

Michael Denkowski and Alon Lavie. 2012. Challenges in Predicting Machine Translation Utility for Human Post-Editors. In *Proceedings of AMTA 2012.*

Yifan He, Yanjun Ma, Josef van Genabith, and Andy Way. 2010. Bridging SMT and TM with Translation Recommendation.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180. Association for Computational Linguistics.

Philip Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X*, pages 79–86, Phuket, Thailand.

Maarit Koponen. 2012. Comparing Human Perceptions of Post-editing Effort with Post-editing Operations. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 181–190. Association for Computational Linguistics.

Michael Lesk. 1986. Automated Sense Disambiguation Using Machine-readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. In *Proceedings of the 5th annual international conference on Systems documentation (SIGDOC86).*

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proceedings of the ACL workshop on Text Summarization Branches Out.*, pages 74–81, Barcelona, Spain.

Yanjun Ma, Yifan He, Andy Way, and Josef van Genabith. 2011. Consistent Translation using Discriminative Learning: a Translation Memory-inspired Approach. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1239–1248.

Yashar Mehdad, Matteo Negri, and Marcello Federico. 2012a. Detecting semantic equivalence and information disparity in cross–lingual documents. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL'12)*, pages 120–124, Jeju Island, Korea.

Yashar Mehdad, Matteo Negri, and Marcello Federico. 2012b. Match without a Referee: Evaluating MT Adequacy without Reference Translations. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, WMT '12, pages 171–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

M. Porter. 2001. Snowball: A language for stemming algorithms.

Martin Potthast, Alberto Barrón-Cedeño, Andreas Eiselt, Benno Stein, and Paolo Rosso. 2010. Overview of the 2nd International Competition on Plagiarism Detection. *Notebook Papers of CLEF*, 10.

Christopher B. Quirk. 2004. Training a Sentence-Level Machine Translation Confidence Measure. In *In Proceedings of LREC*.

Matthew Snover, Nitin Madnani, Bonnie J. Dorr, and Richard Schwartz. 2009. Fluency, Adequacy, or HTER?: Exploring Different Human Judgments with a Tunable MT Metric. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, StatMT '09, pages 259–268, Stroudsburg, PA, USA. Association for Computational Linguistics.

Radu Soricut and Abdessamad Echihabi. 2010. TrustRank: Inducing Trust in Automatic Translations via Ranking. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 612–621, Stroudsburg, PA, USA. Association for Computational Linguistics.

Radu Soricut, Nguyen Bach, and Ziyuan Wang. 2012. The SDL language weaver systems in the WMT12 quality estimation shared task. In *Proceedings of the Seventh Workshop on Statistical Machine Translation (WMT'12)*, pages 145–151, Montréal, Canada.

Lucia Specia, Nicola Cancedda, Marc Dymetman, Marco Turchi, and Nello Cristianini. 2009. Estimating the Sentence-Level Quality of Machine Translation Systems. In *Proceedings of the 13th Annual Conference of the European Association for Machine Translation (EAMT'09)*, pages 28–35, Barcelona, Spain.

Lucia Specia, Dhwaj Raj, and Marco Turchi. 2010. Machine Translation Evaluation versus Quality Estimation. *Machine translation*, 24(1):39–50.

Lucia Specia, Najeh Hajlaoui, Catalina Hallett, and Wilker Aziz. 2011. Predicting machine translation adequacy. In *Proceedings of the 13th Machine Translation Summit*, pages 513–520, Xiamen, China, September.

Lucia Specia. 2011. Exploiting Objective Annotations for Measuring Translation Post-editing Effort. pages 73–80.

Marco Turchi and Matteo Negri. 2013. ALTN: Word Alignment Features for Cross-Lingual Textual Entailment. *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013)*.