Towards a post-editing recommendation system for Spanish–Basque machine translation

Nora Aranberri
IXA research group
University of the Basque Country UPV/EHU
nora.aranberri@ehu.eus

Jose A. Pascual
School of Computer Science
The University of Manchester
jose.pascual@manchester.ac.uk

Abstract

The overall machine translation quality available for professional translators working with the Spanish–Basque pair is rather poor, which is a deterrent for its adoption. This work investigates the plausibility of building a comprehensive recommendation system to speed up decision time between post-editing or translation from scratch using the very limited training data available. First, we build a set of regression models that predict the post-editing effort in terms of overall quality, time and edits. Secondly, we build classification models that recommend the most efficient editing approach using post-editing effort features on top of linguistic features. Results show high correlations between the predictions of the regression models and the expected HTER, time and edit number values. Similarly, the results for the classifiers show that they are able to predict with high accuracy whether it is more efficient to translate or to post-edit a new segment.

1 Introduction

Although machine translation (MT) quality is getting better every day, neither regular users nor professional translators can blindly trust the correctness of a translation. Therefore, providing them with information about the quality together with the actual translation seems sensible. We could argue that this is relevant for regular users, who might not necessarily have a native-like command of the source and target languages. But it is of no lesser importance for professional translators who, being able to assess the quality themselves, might be able to speed up this process.

In this paper, we specifically focus on the case of Spanish–to–Basque professional translators. Note that MT quality for this language pair can be considered relatively poor (Aranberri et al., 2014; Aranberri et al., 2017) - at least that provided by freely accessible systems such as *itzultzailea*[1] or *Google Translate*[2]- and as a result, MT in the professional domain is very rarely used (Garmendia et al., 2017). In this context, we investigate whether we could build estimation models that may prove informative for translators and help with the integration of MT technology in this sector.

To that end, we use a small set of data collected in a post-editing workshop, where post-editing seems to benefit productivity at times. We first aim at providing professional translators with indicators of estimated work to guide their decision whether to post-edit or translate from scratch. For this, we build a set of regression models to estimate indicators of post-editing effort (overall MT quality, time and edits) which we obtain from data solely consisting of post-editing work. Results show high correlations over 0.70 between real and estimated indicators.

Nevertheless, it is undeniable that a recommendation model that suggests the most efficient editing approach would be a more direct way to help in such process. The recommendation could be used either to opt for the most efficient approach during editing or to filter out MT output before the editing phase starts. Thus, we build classification

[1] http://www.itzultzailea.euskadi.eus
[2] https://translate.google.com

Pérez-Ortiz, Sánchez-Martínez, Esplà-Gomis, Popović, Rico, Martins, Van den Bogaert, Forcada (eds.)
Proceedings of the 21st Annual Conference of the European Association for Machine Translation, p. 21–30
Alacant, Spain, May 2018.

models using linguistic features to recommend the editing approach that increases the productivity the most. However, given the low accuracy of the classifiers, we try to improve them by adding specific post-editing effort features. As this information is only available once the editing is completed, we estimate it using the above-mentioned regression models. Results show a large increase in the capacity of the classifiers to provide the correct editing approach even considering the loss of accuracy introduced by the regression models.

The remaining of the paper is structured as follows. A short overview of related work is presented in Section 2. In Section 3 we describe the data sets and features used to train the models while the experimental set-up is outlined in Section 4. Section 5 and Section 6 present the results for the regression and classification models, respectively. Finally, Section 7 summarizes the main conclusions and possible lines of future work.

2 Background

In this section we present an overview of the quality and post-editing effort indicators studied in the literature. In 2004, Blatz et al. (2004) brought confidence estimation techniques, mainly used in speech recognition until then, to the area of MT as they considered that these could help in filtering translations for post-editing, among other tasks. They built models for sentence-level annotation by training regressors and classifiers to predict NIST and WER values. Whereas the tasks themselves proved interesting, experiments revealed that estimated automatic metrics did not match human annotations of quality or post-editing effort.

Similar results were reported by Specia et al. (2009), who used a number of MT system-independent and MT system-dependent features to train a regression algorithm to estimate both NIST and human scores. The models performed well for human annotations, but once again, correlations with automatic metrics were not as successful. From then on, Specia and Farzindar (2010) tested the use of TER and HTER (Snover et al., 2006), which supposedly consider the actual post-editing work translators perform more closely, to build the estimation models. This time, the models correlated well with human annotations of post-editing effort. For that reason, HTER was established as the global quality indicator in quality estimation (QE) tasks and remains so today, despite attempts at looking for alternative ways of measuring quality (Specia et al., 2011).

Since then, a number of authors have worked on building models to provide translators with useful information. Some have tried to describe post-editing time (Specia, 2011) whereas others have focused on selecting the best MT output from a pool of candidates (Avramidis et al., 2011), or on recommending whether a source segment should be tackled using a MT candidate or a translation memory candidate (He et al., 2010). However, it could be argued that the main bulk of research in quality estimation has been shaped by the yearly QE Shared Task, in place since 2012. In its first year, participants focused on correlating estimation models with manual annotations of quality defined as 5 levels of post-editing effort (Moreau and Vogel, 2012; Hardmeier et al., 2012). In 2013 and 2014, the goals were broadened and tasks involved predicting HTER and post-editing time and ranking MT candidates (Beck et al., 2014; Bicici and Way, 2014). Since then, however, efforts have mainly addressed HTER and even if submissions for other indicators such as post-editing time and keystrokes have been welcome, no results have been published on these aspects.

In order to provide professionals with a wider set of pointers that guides the translation task, in this paper we expand the post-editing effort indicators. Specifically, we propose to create a recommendation system that (1) estimates the quality of the MT output as defined by HTER, (2) predicts post-editing effort according to time, and the type and number of edits, and (3) recommends the editing approach for a particular segment by classifying it for either post-editing or translation from scratch.

3 Data Collection and Processing

Unlike for other mainstream language pairs, no readily-available data exists to train quality estimation models for the Spanish–Basque pair. For this purpose, therefore, we adapted post-editing data collected in a workshop for professional translators run in 2015. In this section we describe the data and the linguistic features that we used to build the estimation models.

3.1 Data Sets

In the above-mentioned workshop, translators worked on a series of post-editing tasks and pro-

Task Number	Task Type	Translators	Text	MT System	Sentences	Source Words
1–4	post-editing	10	1	itzultzailea	60	1,467
5	productivity	10	1	itzultzailea	21	495
6–9	post-editing	10	1	itzultzailea	81	1,958
10	productivity	9	1	itzultzailea	16	506
11–14	post-editing	8	1	itzultzailea	82	2,043
15	productivity	8	1	itzultzailea	22	366
16–19	post-editing	8	1	itzultzailea	80	1,964
20	productivity	8	1	itzultzailea	29	516
21–24	post-editing	8	1	itzultzailea	138	2,045
25	productivity	8	1	itzultzailea	26	515
26–29	post-editing	6	1	Google Translate	121	2,082
30	productivity	6	1	Google Translate	24	508
31–34	post-editing	5	2	itzultzailea	187	2,012
35	productivity	5	2	itzultzailea	60	486

Table 1: List of total tasks performed by professional translators.

ductivity tests over a period of seven weeks (See Table 1). For the productivity tests, translators alternately post-edited and translated source sentences. We divided translators in two groups who performed the opposing editing approach for each segment. Throughout the workshop they translated a report by the Basque Institute of Women about Sexism in toys advertising (Text 1) and two short user guides for a mobile phone and a washing machine (Text 2). The original Spanish texts were translated using *itzultzailea*, the MT system made publicly available by the Basque Government and powered by Lucy. The overall MT output was of relative low quality (~ 50.7 HTER) and translators introduced a significant number of edits to turn the segments into acceptable translations.

Task	Avg. PE time/word	Avg. TR time/word
5	4576.73	**4353.46**
10	3058.86	3882.97
15	2920.31	4400.37
20	3454.05	4224.66
25	3174.79	3520.80
30	3523.23	**2974.36**
35	3054.51	291.58

Table 2: Average post-editing and translation time (ms) per word for each productivity task performed by translators.

For our experiments (see Section 4.1), we divided the data collected in the tasks into two sets, namely, the post-editing (PE) set and the productivity (PR) set. The former includes all the segments from the post-editing tasks whereas the latter includes those from the productivity tests. We discarded all tasks performed during the first week (tasks 1–5) as this was the first contact translators had with post-editing and therefore their work was deemed unreliable. Also, we decided to discard tasks 26–30, as they were performed using a different MT system, with which the translating time appears to be lower than the post-editing time (See Table 2). As a result, we collected work for 568 source segments (10,022 words) from the post-editing task and 153 segments (2,389 words) from the productivity test. Note that because all translators were asked to perform the same tasks, our sets include information about several final translations for each of the source segments.

Finally, we added the information required to train the models which is not present in the original data to both sets. Firstly, in order to build models to predict the post-editing indicators, we added HTER scores, the number of each edit-type and the total number of edits to the PE set. Editing times were already present. Secondly, for the classification models, we added to the PR set labels referring to the editing approach that benefits each source segment the most. As opposed to the method used in the 2012 QE Task where manual annotation of perceived post-editing effort was performed by professional translators according to a 5-level scale, our strategy to assign the labels mainly relied on the time gain introduced by the fastest approach. To this end, we used the productivity ratio (translation time/post-editing time). In our case, we calculated the ratio for each source segment with the averaged editing times of the different translators to account for translator variability. Scores above 1 indicate that post-editing is more productive whereas scores below 1 indicate the extent to which translation is faster.

We used three sets of labels, L2, L3 and L5 which involve two, three and five labels, respectively. L2 directly assigns a *post-edit* label to all ratios above 1 and a *translate* label to all ratios below 1. L3 considers that, given the editing variability among translators, scores close to 1 may not reliably predict the most effective approach nor indicate much time difference between them. Therefore, ratios ranging between 0.90–1.10 are assigned the *any approach* label. Finally, L5 adds two extra labels to the L3 set which identify those segments that are clearly more efficient to either post-edit (above 1.30) or translate (below 0.70).

3.2 Features

We extracted the same set of 17 baseline features provided by the WMT12-17 QE Tasks using *Quest++* (Specia et al., 2015). They are black-box features, that is, shallow MT system-independent features. Most of them rely on the comparison of the sentences against a large training corpus, e.g. language model probabilities, n-gram frequencies and translation options per word.

The monolingual Spanish and Basque corpora we used to this end consist of 38 and 44 million segments, respectively. The Spanish corpus includes data released for the WMT tasks (Europarl corpus, UN corpus, News Commentary corpus, etc.). The Basque one comprises texts from different sources such as the Basque newspaper *egunkaria* and radio–television *EITB*, the Elhuyar Web Corpus and administrative translation memories. The bilingual corpus used to train GIZA++ is a considerably smaller set of 7.8 million segments. Overall, the corpora, and specially the monolingual sections, are of a good size to model the relevant languages. However, the domain of our data sets is not represented in them, which could significantly harm the accuracy of the features.

For this reason, we tried to overcome this drawback by adding linguistic information directly extracted from the segments in the data sets. We want to remark, however, that it is not the aim of this work to do feature ingeneering as in Specia and Felice (2012) and Avramidis (2012). For Spanish, we processed the text using *ixa-pipes* tools (Agerri et al., 2014) and for Basque, we used *ixaKat* (Otegi et al., 2016). We collected POS frequencies, tags for morphological features and dependency relations for both source and target segments. Therefore, we added a feature for each POS, morphological feature and dependency relation, whose value was the number of times it appeared in the segment (10, 185 and 42 features for Spanish, respectively, and 10, 316 and 28 for Basque). However, preliminary tests showed that no improvement was coming from the morphological features so we decided to discard them.

For the experiments, we therefore use four different data sets. PE-17 and PR-17 include the baseline features only and PE-107 and PR-107 also use the additional linguistic features.

4 Experimental Set-up

In this section we explain the experiments carried out to predict the MT quality and the post-editing effort required to transform the MT output into the desired quality standard.

4.1 Experiments

We divided the experiments into three distinct parts. In the first part we evaluate the ability of five regression algorithms to learn a number of models to predict indicators of post-editing effort. The indicators are as follows:

- **HTER:** This metric is used as a global quality measure for the professional translator. It is an edit-distance metric that considers the number of edits to be made to a MT segment to transform it into the desired final translation normalized by the number of words in the reference sentence.

- **Post-editing time:** This indicator accounts for the time required by a professional translator to transform the MT output into the desired final text. We give the estimates in milliseconds per segment.

- **Edit types:** This indicator provides individual information for each type of edit, i.e., insertions, deletions, substitutions and shifts, to be introduced to the MT output as computed by HTER. Although the mathematical approach used by the HTER metric to calculate the edits often differs from the linguistically-motivated instinct of translators, they might prove useful in gauging the complexity of the expected post-editing effort.

- **Number of edits:** This is a raw indicator of the number of edits to be made to the MT output to reach the desired quality as computed by HTER. Whereas the edit types are more informative, this provides a rawer measurement of the overall changes.

The second set of experiments is devoted to building and measuring the capacity of classification models to suggest whether a source segment should be translated or post-edited.

However, given the limited data available, we expected these models to have low accuracy. For this reason, we also proposed and evaluated a third set of experiments in which the features of the second data set are incremented with indicators of post-editing effort. To do so, we train the models with real post-editing effort indicators even if these are not available for new segments. Then we apply the regression models described in the first set of experiments to predict these additional features for the new segments before testing (See Figure 1). We expect that the accuracy of the classifiers will increase with these additional features.

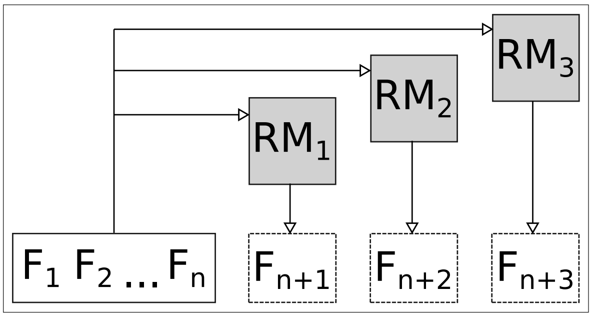

Figure 1: Representation of the extension of the number of features from n to $n+3$ using three previously trained regression models.

In all the experiments the learning and testing process was carried out using 10-fold cross-validation over the PE and PR data sets. The accuracy of the regression models was measured using the correlation coefficient (ρ) which measures the strength and the direction of a linear relationship between two variables. On the other hand, the accuracy of the classifiers was measured using the area under the curve ROC. Each experiment was repeated 10 times and we report the average (μ_ρ and μ_{ROC}) and the standard deviation (σ_ρ and σ_{ROC}). We also performed a paired t-test ($p < 0.05$) to check statistical significance of the results of the algorithms (in bold). We also checked, for each algorithm, if the addition of linguistic features is significant (with the symbol †) .

4.2 Regression and Classification Algorithms

There are countless machine learning algorithms to train regression and classification models. As the purpose of our experiments is to explore the ability of these algorithms to train the recommendation system, we selected six of the most used ones to have an insight into their individual performance.

- Linear regression (LR): It is used for *regression* and it works by estimating coefficients for a line or hyperplane that best fits the training data. It is fast to train and can have great performance if the output is a linear combination of the inputs.

- Logistic regression (LG): It is a regression model (used for *classification*) that estimates the probability of class membership as a multi-linear function of the features.

- k-Nearest Neighbors (k-NN): This algorithm supports both *classification* and *regression*. It works by storing the training dataset and locating the k most similar training patterns to perform a prediction.

- Classification And Regression Trees (CART): They work by creating a tree to evaluate an instance of data, starting at the root of the tree and moving down to the leaves until a prediction can be made. They support both *classification* and *regression*.

- Support Vector Machine (SVM): This is an algorithm for *classification* which finds a line that best separates the training data into classes. The adaptation of SVM for *regression* is called Support Vector Regression (SVR) and works by finding a line that minimizes the error of a cost function. In both cases we use a polynomial kernel.

- Multi-Layer Perceptron (MLP): This algorithm supports both *regression* and *classification* problems using neural networks.

We want to remark that this is a first attempt to measure the quality of the predictions leaving as future work the fine tuning of these algorithms.

5 Results of Regression Models for Quality and Post-editing Work

In this section we present the regression models that aim to predict the post-editing effort. We report the results for each indicator, namely, overall quality, time and edits, separately using both the PE–17 and PE–107 data sets.

5.1 Overall Quality with HTER

Let us start by analyzing the results to estimate segment quality (HTER) by focusing on the PE–17 data set (see Table 3). The results show that the correlation coefficient obtained by k-NN is the highest at 0.71, closely followed by CART. LR and SVR obtain the poorest results with a notably lower correlation coefficient of 0.35 and 0.32, respectively. This suggests that neither LR nor SVR are able to model the relation between the features and the HTER values. In order to confirm this, we performed a test to measure the correlation between the features and HTER, which showed that except for three cases, correlations were lower than 0.1. Indeed, these algorithms are best fitted to capture liner relations and a quick test using a non-linear kernel in SVR revealed an increase of the average correlation to 0.68±0.04 in PE–17 and to 0.70±0.04 in PE–107.

	PE–17		PE–107	
Alg	μ_ρ	σ_ρ	μ_ρ	σ_ρ
LR	0.3499	0.0399	0.4509†	0.0373
k-NN	**0.7146**	**0.0220**	**0.7144**	**0.0218**
CART	0.6704	0.0367	0.6685	0.0347
SVR	0.3211	0.0415	0.4126†	0.0335
MLP	0.4704	0.0517	0.5870†	0.0456

Table 3: Regression results for the HTER model.

If we compare these results with those obtained using the PE–107 data set to analyze the impact of the linguistic features in the learning process, we observe that for the best performing algorithms in PE–17, k-NN and CART, the contribution of the new features is non-existent. However, the remaining three algorithms do benefit from the addition of the new features significantly. This suggests a stronger linear relation between the features and the HTER values. This was confirmed by testing this relation, which showed that the number of features with a correlation higher than 0.1 with the HTER values had increased to 25.

5.2 Post-editing Time

Previous attempts at estimating time have shown that it is quite an objective indicator for post-editing effort. Looking at the results for the PE-17 data set (see Table 4) we see that, unlike for HTER, all the algorithms perform very similarly (differences not statistically significant) and obtain a correlation coefficient of around 0.71. In this case, the correlation between the features and time is higher than 0.2 for 8 of the features, which explains the good behavior of LR and SVR.

	PE–17		PE–107	
Alg	μ_ρ	σ_ρ	μ_ρ	σ_ρ
LR	0.7137	0.0402	**0.7238**	**0.0372**
k-NN	0.7106	0.0362	0.7131	0.0366
CART	0.7081	0.0392	0.7092	0.0383
SVR	0.7135	0.0405	**0.7265**	**0.0388**
MLP	0.7122	0.0380	0.6955	0.0436

Table 4: Regression results for the time model.

If we examine the results for PE-107, we notice that the contribution of the new linguistic features is not significant for any of the algorithms. We observe, however, that LR and SVR benefit the most from them and obtain the highest results, which are statistically significant in this data set. An analysis of the relation between the features and time showed a correlation higher than 0.2 for 47 features and higher than 0.1 for another 21.

5.3 Edit Types and Total Number

Not much has been published on estimating the different types of edits required to transform the MT output into the desired final version. Avramidis (2014; 2017) trained models for each edit type to then combined them, to try to obtain a higher accuracy HTER model. However, the potential value of the individual models was not considered. If we look at the results, we see that, given their accuracy, we could in fact include them in the recommendation system as part of the information about post-editing effort provided to translators.

Let us consider the different edit types in PE–17 first (see Table 5). We observe that for all the models, k-NN is the best performing algorithm. However, the level of accuracy of the models varies considerably for the different edit types. The model for substitutions is by far the best performing one, with a correlation coefficient of 0.80. Shifts and insertions also get good results. But the

		INSERTIONS		DELETIONS		SUBSTITUTIONS		SHIFTS		TOTAL EDITS	
Dataset	Alg	μ_ρ	σ_ρ	μ_ρ	σ_ρ	μ_ρ	σ_ρ	μ_ρ	σ_ρ	μ_ρ	σ_ρ
PE–17	LR	0.5685	0.0427	0.4537	0.0421	0.7336	0.0180	0.6167	0.0266	0.8029	0.0180
	k-NN	**0.7011**	**0.0687**	**0.5214**	**0.0435**	**0.8035**	**0.0182**	**0.7422**	**0.0238**	**0.8660**	**0.0168**
	CART	0.6556	0.0930	0.4896	0.0459	0.7876	0.0187	0.7164	0.0252	0.8550	0.0176
	SVR	0.5625	0.0273	0.4517	0.0415	0.7325	0.0179	0.6147	0.0274	0.8020	0.0187
	MLP	0.6633	0.0740	0.4731	0.0488	0.7404	0.0205	0.6344	0.0268	0.8125	0.0198
PE–107	LR	0.6167	0.0610	0.4840	0.0394	0.7566	0.0187	$0.6710^{\dagger}$	0.0257	0.8247	0.0175
	k-NN	**0.7010**	**0.0688**	**0.5214**	**0.0435**	**0.8035**	**0.0182**	**0.7423**	**0.0238**	**0.8660**	**0.0167**
	CART	0.6571	0.0893	0.4874	0.0452	0.7872	0.0190	0.7184	0.0249	0.8558	0.0182
	SVR	0.5750	0.0306	0.4723	0.0373	0.7505	0.0190	0.6567	0.0274	0.8193	0.0179
	MLP	0.6664	0.0758	0.4746	0.0466	0.7364	0.0332	0.6674	0.0370	0.8210	0.0268

Table 5: Regression results for the individual edit type and total edits models.

coefficient score for deletions is low at 0.52, showing that the algorithms are not able to capture the relation between the features and this indicator.

If we take a look at the total number of edits, irrespective of their type, we observe that the prediction models perform very well. Again, k-NN is the best performing algorithm with a correlation coefficient of 0.86 but all five score above 0.80.

As with previous regression models, we notice that the new linguistic features added in PE–107 make no or only a marginal contribution to the learning process and in no case improve the results of the best performing algorithm.

5.4 Summary

In summary, we see that we are able to train models that predict HTER and time with a relatively high accuracy and within the range reported by other research despite the limited training data. It is true that the overall performance should be improved, and the models trained and tested on additional data sets before these indicators are provided to translators. However, the results are very promising as there is ample room for tuning.

In reference to edits, regression models perform well in general, although there is strong variation across types. Room from tuning aside, it is worth considering that not all edit types may have the same weight for translators when assessing the work involved during post-editing. Insertions and substitutions require intensive work where translators either add missing information or replace incorrect MT output. Shifts are lower intensity edits, where the correct translation is present, just not in the correct place. These three edit types achieve correlation coefficients of over 0.70 and we could provide them with confidence after additional tuning tests. Deletions, however, score poorly but these could be viewed as very low intensity edit types where translators would easily identify the incorrect elements to eliminate. Therefore, they might not be the edit type that represents the most laborious aspect of post-editing. It remains to be tested which of the types translators find most informative regarding post-editing effort.

Predicting the total number of edits has been much more successful. It may not be as informative as having predictions for the different types but considering the distinct nature of the approach to editing used by humans and machines, it might prove a good compromise that measures the effort in terms of raw changes.

What is interesting to see is the difference in performance between the HTER and the total edit number models, as the latter is based on HTER information. For some reason, the regression models and features seem to be better suited for predicting the errors without considering the length of the final translations. The significantly higher scores obtained for the total edits makes us consider whether providing HTER scores as indication of post-editing effort is appropriate or whether providing raw edit numbers together with sentence lengths would be more accurate and informative.

Finally, it is worth noting that features accounting for the frequencies of POS and dependency relations only contribute to the learning process in a few cases further than the 17 baseline features.

6 Results of Classifiers for Editing Approach

In this section we present the results for the classifiers that aim to predict the editing approach, post-editing or translation, a translator should follow when addressing a new segment. We report results with and without additional linguistic features and also analyse the impact of using post-editing effort indicators as features. We predict label-sets of varying numbers of classes.

6.1 Baseline classification models

We first present the results for the baseline classification models. We trained the models with all available segments in the productivity data set.

We check the results obtained for the 2 label task (2L) first (see Table 6). For both the PR–17 and PR–107 sets, all algorithms perform very poorly with μ_{ROC} below 0.60, with SVR lagging behind (statistical difference). However, if we consider the PR–107 data set, we see that thanks to the additional linguistic features SVR has caught up with the other algorithms (statistical significance between PR–17 and PR–107).

	PR–17 (2L)		PR–107 (2L)	
Alg	μ_{ROC}	σ_{ROC}	μ_{ROC}	σ_{ROC}
LG	**0.56**	**0.15**	0.60	0.15
k-NN	**0.58**	**0.11**	0.56	0.11
CART	**0.51**	**0.11**	0.49	0.09
SVR	0.50	0.02	0.57†	0.11
MLP	**0.59**	**0.13**	0.58	0.17

Table 6: Results of the classification algorithms for 2 labels.

Let us now take a look at the results for 3 labels (3L) (see Table 7). In this case there is no statistically significant difference between the algorithms. Same as before, adding linguistic features only benefits SVR (statistical significance).

	PR–17 (3L)		PR–107 (3L)	
Alg	μ_{ROC}	σ_{ROC}	μ_{ROC}	σ_{ROC}
LG	0.53	0.17	0.50	0.17
k-NN	0.56	0.11	0.55	0.13
CART	0.50	0.11	0.49	0.10
SVR	0.48	0.06	0.57†	0.13
MLP	0.58	0.14	0.56	0.16

Table 7: Results of the classification algorithms for 3 labels.

We observe the same trend of poor results for the 5 label task (5L) (see Table 8). However, in this task, additional linguistic features do not bring any improvement. In fact, the only statistical significance is the setback for LG.

Overall, we conclude that the performance of the classification models is far from being accurate enough to prove useful in a real set-up. Even with room for tuning, we believe that the current features do not properly inform the algorithms for the classification task.

	PR–17 (5L)		PR–107 (5L)	
Alg	μ_{ROC}	σ_{ROC}	μ_{ROC}	σ_{ROC}
LG	0.57†	0.25	0.40	0.24
k-NN	0.57	0.15	**0.56**	**0.15**
CART	0.54	0.13	**0.55**	**0.13**
SVR	0.57	0.22	**0.52**	**0.22**
MLP	0.69	0.22	**0.55**	**0.22**

Table 8: Results of the classification algorithms for 5 labels.

6.2 Classification models using predictions for post-editing work

In an attempt to improve the performance of the classification models, we propose to use indicators of post-editing effort as features for training. We believe that these indicators reflect more closely the reasons why a translator would choose one editing approach over the other. For that, we first analyse whether the previous classifiers perform better by adding original HTER, total edits and time as features to the PR set. Secondly, as these three features are not available until translation is completed, we test new classifiers with predicted post-editing features (see Section 4.1).

We summarize the results of adding the three post-editing effort indicators as features in Table 9. Results are given by μ_{ROC}, which corresponds to the average results of the training set. For the sake of space we omit the standard deviations of the learning process. We can see that in the PR–20 set the accuracy of the models varies from fair to excellent regardless of the number of labels. This is a large improvement over the baseline classifiers that reveals the potential of adding HTER, time and edit number as features. Whereas k-NN, CART and MLP are the best performing algorithms across all sets, LG and SVR are the worst scoring for PR-20. However, as in the regression

	PR–20			PR–110		
Alg	2L	3L	5L	2L	3L	5L
LG	0.76	0.74	0.80	**0.99**†	**1.00**†	**0.99**†
k-NN	**0.99**	**0.99**	**0.98**	**1.00**†	**1.00**†	**1.00**†
CART	**0.98**	**0.99**	**1.00**	**0.99**	**1.00**	**1.00**
SVR	0.64	0.63	0.77	0.91†	0.93†	0.96†
MLP	**0.97**	**0.96**	**0.95**	**1.00**	**0.99**	0.96

Table 9: Results of the classification algorithms using the post-editing effort features given by μ_{ROC}.

experiments, we see that when new linguistic features are added (PR-110), the performance of LG and SVR improves (statistical significance). Interestingly, the performance of k-NN also benefits from the linguistic features.

Given the promising results obatined when adding the post-editing effort indicators, we test this approach using a scenario viable for deployment. We divide the productivity data set into a training and a test set. Out of the 153 unique source segments, we randomly include 80% in the training set and 20% in the test set. The training set includes all the available data for each of the unique segments. The test set, in turn, only includes one instance of each unique segment with HTER, time and total edits as predicted by the best-performing models in Section 6 on top of the initial features (PR–20 and PR–110 after adding the 3 new features). Results, given by T_{ROC}, the ROC value obtained after applying one of the learnt models to the test set, are summarized in Table 10.

	PR–20			PR–110		
Alg	2L	3L	5L	2L	3L	5L
LG	0.66	0.69	0.75	0.70	0.73	0.75
k-NN	**0.89**	0.89	0.90	**0.89**	**0.99**	0.90
CART	0.88	**0.90**	0.89	**0.89**	0.90	0.90
SVR	0.56	0.55	0.60	0.83	0.85	0.87
MLP	0.88	0.83	**0.92**	0.88	0.90	**0.99**

Table 10: Results of the classification algorithms using the post-editing effort features given by T_{ROC}.

With this set-up, the best scoring models range between good and excellent for both PR–20 and PR–110 sets and for all the label sets. Notice that our predictions carry over the margin of error of the regression models, but still their level of accuracy is very high. In the PR–20 set, MLP is the best performing algorithm and LG and SVR perform very poorly. In the PR–110 set, the same best performers remain on top but k-NN and CART perform particularly well for 2 and 3 labels, and MLP for 5 labels. It is also worth noting the improvement of SVR in this data set. Overall we can argue that the results for the classification model are promising and could be useful in a real setting. Even more, we expect further improvement from tuning the classifiers and from obtaining more accurate predictions from the regression models.

7 Conclusions and Future Work

In this paper we tested the feasibility of training a number of estimation models that go beyond the usual MT quality level to build a recommendation system that helps speed up the decision time of professional translators to decide whether to post-edit or translate. In particular, we studied if reasonable results could be obtained for the Spanish–Basque pair, for which MT quality is low, and thus not widely used within the professional sphere.

We trained regression models to predict HTER, time, types and total number of edits as indicators of post-editing effort using a limited data set. We show that relatively high correlation coefficients can be achieved for almost all indicators. The total edit number seems the easiest to predict whereas accuracy is lower for each of the individual types, particularly for deletions. Results also reveal that adding POS and dependency relation frequencies as features did not generally improve the majority of our models. k-NN was the best performing algorithm, with the best results for HTER and all the models involving edits and with no contribution from the new linguistic features. For the time model, LR and SVR performed best, obtaining marginal gains with the new features.

Besides providing post-editing effort indicators, we also trained a classification model that would recommend translators the editing approach to take. Whereas the baseline models performed poorly, we showed that including post-editing effort indicators as features largely improves the results. As this information is not available for new segments, we successfully used previously trained regression models to add these features in new test sentences. k-NN, CART and MLP consistently show the best performance across all the data sets.

Given the good results achieved, the next step would involve tuning and testing the models in further data sets. Our aim is to investigate to what extent HTER, post-editing time and edit types are valuable indicators for professionals translators.

Acknowledgements: The research leading to this work was partially funded by the *TIN2015-70214-P* project (MINECO-FEDER) and the *KK-2017/00094* project (Basque Government).

References

Agerri, Rodrigo, Josu Bermudez and German Rigau. 2014. IXA pipeline: Efficient and Ready to Use

Multilingual NLP tools. *LREC2014, 9th Language Resources and Evaluation Conference*, Reykjavik, Iceland. 3823–3828.

Aranberri, Nora, Gorka Labaka, Arantza Diaz de Ilarraza, and Kepa Sarasola. 2014. Comparison of post-editing productivity between professional translators and lay users. *Third Workshop on Post-editing Technology and Practice*, Vancouver, Canada. 20–33.

Aranberri, Nora. 2017. What Do Professional Translators Do when Post-Editing for the First Time? First Insight into the Spanish-Basque Language Pair. *HERMES-Journal of Language and Communication in Business*, (56):89–110.

Aranberri, Nora, Gorka Labaka, Arantza Diaz de Ilarraza, and Kepa Sarasola. 2017. Ebaluatoia: crowd evaluation for English-Basque machine translation. *Language Resources and Evaluation*, 51(4):1053–1084.

Avramidis, Eleftherios. 2017. Sentence-level quality estimation by predicting HTER as a multi-component metric. *WMT-2017, Conference on Machine Translation*, Copenhagen, Denmark. 534–539.

Avramidis, Eleftherios. 2014. Efforts on Machine Learning over Human-mediated Translation Edit Rate. *WMT-2014, 9th Workshop on Statistical Machine Translation*, Baltimore, Maryland. 302-306.

Avramidis, Eleftherios. 2012. Quality estimation for machine translation output using linguistic analysis and decoding features. *WMT-2012, Seventh workshop on statistical machine translation*, Montreal, Canada. 84–90.

Avramidis, Eleftherios, Maja Popovic, David Vilar, and Aljoscha Burchardt. 2011. Evaluate with confidence estimation: machine ranking of translation outputs using grammatical features. *WMT-2011, Sixth Workshop on Statistical Machine Translation*, Edinburgh, Scotland. 65–70.

Beck, Daniel, Kashif Shah and Lucia Specia. 2014. SHEF-Lite 2.0: Sparse Multi-task Gaussian Processes for Translation Quality Estimation. *WMT-2014, Ninth Workshop on Statistical Machine Translation*, Baltimore, Maryland. 307-312.

Bicici, Ergun and Andy Way. 2014. Referential Translation Machines for Predicting Translation Quality. *WMT-2014, Ninth Workshop on Statistical Machine Translation*, Baltimore, Maryland. 313-321.

Blatz, John, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence Estimation for Machine Translation. *ACL-2004, 42th Annual Meeting of the Association for Computational Linguistics*, Geneva, Switzerland. 315–321.

Garmendia, Lierni, Naroa Lasarte and Maialen Pinar. 2017. Situación actual y viabilidad de la TA en euskera: posedición y análisis de los resultados de un motor de TABR español-euskara. *Master Thesis*, Universitat Autònoma de Barcelona.

Hardmeier, Christian, Joakim Nivre and Jorg Tiedemann. 2012. Tree kernels for machine translation quality estimation. *WMT-2012, 7th Workshop on Statistical Machine Translation*, Montreal, Canada. 109–113.

He, Yifan, Yanjun Ma, Josef van Genabith, and Andy Way. 2010. Bridging SMT and TM with Translation Recommendation. *ACL-2010, 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden. 622–630.

Moreau, Erwan and Carl Vogel. 2012. Quality estimation: an experimental study using unsupervised similarity measures. *WMT-2012, 7th Workshop on Statistical Machine Translation*, Montreal, Canada. 120–126.

Otegi, Arantxa, Nerea Ezeiza, Iakes Goenaga, and Gorka Labaka. 2016. A Modular Chain of NLP Tools for Basque. *TSD 2016, 19th International Conference on Text, Speech and Dialogue*, Brno, Czech Republic. 93–100.

Snover, Matthew, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. *AMTA-2006, 7th Conference of the Association for Machine Translation in the Americas*, Cambridge, Massachusetts. 223–231.

Specia, Lucia, Marco Turchi, Nicola Cancedda, Marc Dymetman, and Nello Cristianini. 2009. Estimating the Sentence-Level Quality of Machine Translation Systems. *EAMT-2009, 13th Conference of the European Association for Machine Translation*, Barcelona, Spain. 28–37.

Specia, Lucia, and Atefeh Farzindar. 2010. Estimating Machine Translation Post-Editing Effort with HTER. *AMTA-2010, Workshop Bringing MT to the User: MT Research and the Translation Industry*, Denver, Colorado. 33–41.

Specia, Lucia, Najeh Hajlaoui, Catalina Hallett and Wilker Aziz. 2011. Predicting machine translation adequacy. *Machine Translation Summit XIII*, Xiamen, China. 19–23.

Specia, Lucia. 2011. Exploiting objective annotations for measuring translation post-editing effort. *EAMT-2011, 15th Conference of the European Association for Machine Translation*, Leuven, Belgium. 73–80.

Specia, Lucia and Mariano Felice. 2012. Linguistic features for quality estimation. *WMT-2012, Seventh workshop on statistical machine translation*, Montreal, Canada. 96–103.

Specia, Lucia, Gustavo Henrique Paetzold and Carolina Scarton. 2015. Multi-level Translation Quality Prediction with QuEst++. *ACL-IJCNLP 2015 System Demonstrations*, Beijing, China. 115–120.