Recommending Machine Translation Output to Translators by Estimating Translation Effort: A Case Study

Prashant Mathur, Nick Ruiz, and Marcello Federico

***Abstract*—In this paper we use the statistics provided by a field experiment to explore the utility of supplying machine translation suggestions in a computer-assisted translation (CAT) environment. Regression models are trained for each user in order to estimate the time to edit (TTE) for the current translation segment. We use a combination of features from the current segment and aggregated features from formerly translated segments selected with content-based filtering approaches commonly used in recommendation systems. We present and evaluate decision function heuristics to determine if machine translation output will be useful for the translator in the given segment. We find that our regression models do a reasonable job for some users in predicting TTE given only a small number of training examples; although noise in the actual TTE for seemingly similar segments yields large error margins. We propose to include the estimation of TTE in CAT recommendation systems as a well-correlated metric for translation quality.**

***Index Terms*—Machine translation, computer-assisted translation, quality estimation, recommender systems.**

I. Introduction

RECENT advances in Statistical Machine Translation (SMT) have been due to the large availability of parallel corpora. A natural goal is to apply machine translation to the computer-assisted translation (CAT) domain to increase translator productivity. CAT tools typically consist of a Translation Memory (TM) which stores segments that have been translated before by the users. When a translator is translating a new sentence, the sentence is first looked up in the TM and if there is a fuzzy match (a partial match score above a given threshold) the CAT tool suggests the translation in the TM to the translator. In the common scenario where the TM does not provide any suggestions to the translator, the translator must translate the segment from scratch.

The aim of CAT tools is to improve the productivity of translator and to ensure consistency. Integrating SMT system in a CAT tool has been shown to speed up the translation process [1]. Machine translation output provides excellent coverage that can help overcome sparsity in the TM; however, MT output can also add additional noise on the screen that can distract the translator from her task. Ideally, the goal for SMT in the CAT scenario is to provide machine translation suggestions only when their quality in order to guarantee an increase in the user's productivity for a given segment.

The aim of the EU-funded MateCat[1] project is to increase translator productivity by providing self-tuning, user-adaptive and informative MT in the CAT scenario. In this paper we propose the integration of a recommendation system framework using content-based filtering to suggest when MT output should be presented on the translator's screen, based on the difficulty of the current segment and his previous behavior on similar segments. We do so by estimating the amount of effort, as a function of time, to translate the current segment, given a MT suggestion. Given the estimated time to translate a sentence, we attempt to define a decision function to determine if the MT output will increase the productivity of the translator.

This paper is organized as follows. In Section II we describe related work in the field of quality estimation. Section III outlines the conditions of a preliminary field test conducted at the beginning of the MateCat project. Section IV describes the methodology used to estimate the time to edit a sentence (TTE). Our experiment and results using the field test dataset are described in Sections V and VI, respectively. Finally, we discuss the results of the experiment and provide suggestions for future work.

II. Previous Work

Automatic Quality Estimation (QE) for Machine Translation (MT) seeks to predict the usefulness of MT outputs without observing reference translations. QE can be cast as a machine learning problem whose goal is to predict a quality score based on one or more metrics, including human post-editing effort.

A baseline regression system was constructed in [2] for the WMT 2012 Quality Estimation shared task that uses MT and surface-level features derived from a training set to predict a user's perceived level of post-editing effort for newly translated sentences. Levenshtein features are used in [3] which measure the distance between the current sentence with each of the closest entry in the training corpus. Low edit

Manuscript received on December 7, 2012; accepted for publication on January 11, 2013.

Prashant Mathur and Nick Ruiz are with University of Trento and FBK, Italy.

Marcello Federico is with FBK, Italy.

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

distances imply that the current sentence is close to the training set and thus its quality is expected to be high.

Sentences in a development set are ranked by their sentence-level BLEU scores [4] and divided the development set into quartiles in [5]. Inspired by *TrustRank* [6], additional regression features are added to measure the distance between the current sentence and the high or low quality quartile sets in the development data. The distance is measured via n-gram matches through a modified BLEU score and is evaluated in both the source and the target language directions.

The prediction of word- and sentence-level MT errors are treated as a sequence labeling task in [7], using alignment, agreement, and POS-based features. Each token is labeled by the type of error occurring (e.g. insertion, deletion, substitution, or shift).

Quality estimation for MT is treated as a binary classification problem for computer-assisted translation in [8]. They predict the usefulness of MT output over translation memory recommendations in terms of the number of words edited to match a reference translation. This metric is known as the translation error rate (TER) [9]. The system also provides confidence scores based on posterior classification probabilities. A MT output is recommended if its corresponding TER score is lower than that of a TM suggestion.

Due to the nature of the field test, we treat the SMT system as a black box and cannot use SMT-based features in our model. However, we consider the reverse translation fuzzy match score as a feature in our model.

III. Field Test

The EU-funded project MateCat was launched in early 2012 with the aim to integrate Statistical Machine Translation systems such as Moses [10] with a state-of-the-art CAT tool to improve translator productivity. The goal of MateCat is to seamlessly integrate a MT engine as a back-end process inside the CAT tool. Translators will receive translations either from TM matches or from the MT engine. One of the aims of the project is to recommend MT outputs if the machine translation requires less post-editing than the translation memory.

A feasibility study was conducted in [1] as a field test which integrated a production-quality MT Engine (namely, Google Translate[2]) in SDL Trados Studio[3]. Twelve professional translators worked on real translation projects covering the information technology and legal domains. Documents were translated both from English to German and English to Italian. The experiment was held in two parts. In the first part, a baseline was established by providing translators with only TM matches for suggestions. We refer to this baseline as TM experiments. In the second part MT outputs were viable alongside the TM matches. We refer to these as MT experiments. To measure the productivity of translators two indicators were used: the *post editing speed* which is the average number of words processed by translator in one hour, and the *post editing effort* which is the average percentage of word changes applied by the translator on each suggestion. The results from the experiment show that providing MT recommendations significantly increased productivity across all users.

[2]http://translate.google.com

[3]http://www.trados.com/en/

IV. Methodology

Modern recommendation systems typically use two mechanisms: content-based and collaborative filtering. In content-based filtering an item is recommended to a user based on its similarity between items she previously observed. The user's judgments on previously seen items are stored in a user profile, which may also contain additional user information. An item profile is constructed based on a set of characteristics or attributes describing it. The user's profile is combined with item profiles to find new items that a user may prefer. In a CAT scenario, "item profiles" of previously translated segments can be aggregated and used to predict a user's judgment on the quality of a machine translation in a future segment.

In collaborative filtering, a given user is compared against a collection of other users with similar profiles to provide recommendations for new items, even if the items recommended are dissimilar to those preferred by the user in the past. Such filtering increases the pool of items that can be recommended to a user. In a CAT scenario, translation recommendations could be provided based on the previous translations of other users. Unfortunately, such an approach is not useful in a professional translation scenario, where translators must maintain consistency within their own projects. Additionally, a careful look at the data provided in the MateCat field test shows that the translators behave quite differently. However, such an approach might be useful for crowd-sourced or community-based translations with a large number of amateur translators.

CAT systems aim at increasing the productivity of the translators by providing translation suggestions, usually in the form of a TM. It is a necessity that the translations recommended from the MT system are good. If the cost of post-editing a translation is higher than translating the segment from scratch then it decreases the productivity of the user. An ideal recommendation system suggests a translation only when the cost of post-editing is low. Thus, we only use content-based filtering, treating each translation segment as an item to be compared against segments previously translated by the same user. We combine features drawn from the item profile of the current segment along with a collection of item profiles of similar segments to predict the time required to translate the segment. We call this the *time to edit* (TTE). TTE is one of several indicators for translation effort. If additionally providing MT outputs does not significantly improve the translation effort over scenarios where only the TM is available, then it is not useful to recommend the MT output to the translator.

In [8], most of the features for translation recommendation come directly from the SMT models (e.g. phrase translation scores, language model scores). These features are combined with system independent features such as language model perplexity scores on target side, fuzzy match scores and lexical translation scores from a word alignment model [11]. However, in this work we are restricted to the features used in the field test. The underlying MT engine and language models used by Google Translate are unavailable for analysis. Instead, we used features in the MateCat field test, such as *word count*, *TTE* on a sentence, *match percent* (percent match between a TM and the current segment), and the *after match* score (how close the suggestion and translation are).

However, there were several limitations in the original field test. The translators worked remotely on a client system at their respective locations and connected to a centralized server which made it hard to capture the user/translator focused features such as keystrokes, the actual time to translate a sentence, and whether the translator actually used the MT output or translated directly from scratch. In the absence of such features we use a number of aggregated features obtained from the similar segments to the current one being translated (see section V-B). We borrow the idea of a pseudo fuzzy match score from [8]. However, instead of Levenshtein distance we use TER. There are two kinds of features, one representing the statistics of current segment (current features) and the one representing the statistics of the similar segments (aggregated features). Both sets of features are taken from the TM and MT outputs.

Table I provides a list of the features considered in this paper. Features such as the source word count (WC_{SR}) and the TER of the reverse-translated MT output against the source text (TER_{SR}) are computed on the current segment. Several other features are computed on the collection of similar TM and MT segments extracted by means of content-based filtering. Aggregated features provide summary statistics on the TTE and time for word (TFW, computed as TTE divided by segment word count), as well as an estimate on the translation error rate (TER_{TG}) and word counts on the MT and post-edited outputs (WC_{MT} and WC_{PE}, respectively). Each summary statistic yields three features, one for the respective TM and MT aggregates and an additional feature for their difference.

TABLE I
FEATURES USED IN THIS EXPERIMENT. CERTAIN FEATURES ARE DERIVED FROM THE CURRENT SEGMENT, WHILE OTHERS ARE COMPUTED AS AN AVERAGE OVER SIMILAR SEGMENTS. WORD COUNT (WC) FOR SOURCE (SR), MT, AND POST-EDITED (PE) SEGMENTS; TTE AND TFW, AND TER ON TARGET (TG) AND SOURCE (SR) SEGMENTS.

Segment	Current	Similar
WC_{SR}	Y	Y
WC_{MT}	N	Y
WC_{PE}	N	Y
TFW	N	Y
TTE	N	Y
TER_{TG}	N	Y
TER_{SR}	Y	N

TABLE II
SEGMENTS PER USER IN THE EN>DE IT DOMAIN CAT SCENARIO. THE DATASET CONSISTS OF 3670 SEGMENTS.

User	TM Segments	MT Segments
User 1	487	486
User 2	498	481
User 3	332	486
User 4	490	410

V. EXPERIMENT

We perform our experiment on the English to German translation of segments in the information technology domain.

A. Preprocessing

As in [1], segments where the processing time per word was less than 0.5 seconds or greater than 30 seconds were removed as outliers. Perfect (100%) TM matches were removed. After filtering, the data set contains a total of 3670 segments, subdivided among four users, as listed in Table II.

Data is split as follows: the first 10% of each split is reserved as a burn-in to accumulate user statistics for extracting similar segments. The remainder of the data is split into 11 folds in a round-robin fashion to minimize the effects of immediately translated segments (i.e., $seg_1 \rightarrow fold_1, seg_2 \rightarrow fold_2, ..., seg_{12} \rightarrow fold_1$, etc.). Ten folds are used for cross-validation purposes to evaluate the utility of the regression model. The folds are later combined and used to evaluate the final held-out set to provide a recommendation to the user.

B. Extracting similar segments

In order to gather statistics on features not observable in the current segment (see Table I), we compute aggregated features from the statistics of similar segments, both in our TM and MT experiments. We rely on the popular cosine similarity metric using unigram features from the source text for identifying similar segments. Given source language bag-of-word features for segments A and B, cosine similarity is defined as:

$$\text{similarity} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}} \quad (1)$$

In order to simulate the computer-assisted translation scenario, we only compute similarity scores on segments that would appear in a user's "translation cache" – e.g., segments that were already translated by the user at a given point in the TM or MT field test experiments. Since the purpose of the experiment is to evaluate the utility of MT suggestions, the cosine similarity is always calculated from a MT segment to any previous segments.

In the cross-validation scenario, each segment in the training fold draws similar segments from the burn-in set and the previous segments in the training set. Formally, for $fold_i$, similar segments are drawn from the previous segments in a candidate pool defined by $\{burn\text{-}in, fold_{-i}\}$ where $-i$ refers to the cross-validated folds not labeled $fold_i$. The test set draws its similar segments from the candidate pool $\{burn\text{-}in, fold_i\}$.

Average similarity scores per user under cross-validation fold 3 are listed in Table III. Given that the average similarity score (sim_{avg}) per user is low, we select candidate segments with a similarity score higher than sim_{avg}, as an arbitrary segment is semantically unrelated to a given segment. From the remaining candidates, we establish a heuristic that selects the candidates within 10% of the segment with the highest similarity (sel_{max}), or those candidates whose score is higher than the average of the selected candidates (sel_{avg}). In other words, we aggregate the similarity scores of segments whose scores are greater than $\max(\text{sel}_{max} - 0.1, \text{sel}_{avg})$ from the candidate pool.

The aggregated features described in the previous section are computed by averaging the feature values of the similar segments within the TM and MT experiments, respectively. In some cases, no candidates are selected for the TM or MT aggregation. In these cases, we substitute with average statistics for the particular user across all segments in the training set.

C. Predicting time to edit

We train linear regression models, combining features of the current segment with statistics on the user's behavior on similar segments and call this the *Aggregated* regression model. Similarly, we build baseline regression models just with features from current segment (e.g. WC_{SR}, TER_{SR}). Using a 10-fold cross validation strategy, we train 10 regression models for each user based on the training set. Each model is designed to predict the TTE for a given segment. We use the linear regression classes provided by the Weka open-source machine learning toolkit [12] to orchestrate our experiment. Our linear regression models use M5 attribute selection [13] which incrementally removes features with the smallest coefficient. A ridge regularization parameter is fixed to 1.0×10^{-8}. The regression results are averaged across each fold and reported in Section VI. For the final MT output recommendation, we train a model using all the training folds and evaluate on our held-out test set.

D. Suggesting MT output

After predicting the TTE values on a held-out test set, we recommend whether or not to present MT output to the user by comparing the TTE of the current segment against the TTE of segments with similar word counts in the MT and TM experiments. Given a segment s and its predicted TTE x, we evaluate the number of standard deviations of x from $\hat{\mu}_{MT}$ and $\hat{\mu}_{TM}$, the bootstrap mean of the TTE values in the MT and TM experiments for segments with a source-side word count in the range $[WC_{SR}(s) - 1, WC_{SR}(s) + 1]$. Thus, the following criterion (inspired by the Z-score) is defined for recommending MT output:

$$f(x) = \begin{cases} 1: & |(x - \hat{\mu}_{MT}) / \hat{\sigma}_{MT}| < |(x - \hat{\mu}_{TM}) / \hat{\sigma}_{TM}| \\ 0: & \text{otherwise} \end{cases} \quad (2)$$

$\hat{\sigma}_*$ is the standard deviation of segments in the corresponding sample. We use the bootstrap mean as a robust mean estimate to account for outliers.

TABLE IV
AVERAGE PERFORMANCE OF AGGREGATED AND BASELINE SYSTEMS AFTER 10-FOLD CROSS-VALIDATION FOR LINEAR REGRESSION MODELS TRAINED FOR EACH USER. AVERAGE NUMBER OF INSTANCES IN EACH FOLD IS GIVEN BY *Instances*. MEAN ABSOLUTE ERROR (MAE) AND ROOT MEAN SQUARE ERROR (RMSE) ARE IN SECONDS.

		Baseline			Aggregated		
User	Instances	Corr.	MAE	RMSE	Corr.	MAE	RMSE
1	38.9	0.6128	31.94	45.70	**0.6300**	31.16	44.87
2	38.6	**0.6371**	25.79	47.35	0.6362	25.82	48.47
3	38.6	**0.5102**	18.39	38.58	0.4672	19.56	39.47
4	33.0	**0.5293**	13.34	25.72	0.4947	14.29	26.70

VI. RESULTS

The results for cross-validation and the final evaluation are reported below.

A. Regression results

1) Cross validated models: Average regression results along with baseline results per user are reported in Table IV. While Users 3 and 4 have a lower correlation coefficient, the Root Mean Square Error (RMSE) remains relatively low with respect to Users 1 and 2. In particular, User 4's regression curve implies that many of the model features do not contribute to the predictive power of the model. In fact, a baseline system using only two features performs better than an aggregated system for all the users except User 1. This is likely due to the fact that there are few samples in our data set available for aggregation. The following features have a significant contribution to the regression model: current WC_{SR}, δWC_{SR} for similar TM and MT segments, and δWC_{TG}.

2) Final models: Regression results of the final test per user are reported in Table V and regression coefficients for each attribute are reported in Table VI. The correlation coefficients are higher than those of the cross-validation experiment.

Here, the aggregated system performs better than baseline only for Users 1 and 3. The majority of errors occur in segments with high word counts, or "outlier" cases where either an identical match appears in the translation memory or the user took an abnormally long amount of time to translate a segment.

TABLE III
SIMILAR SEGMENT EXTRACTION FOR TWO RANDOMLY SELECTED SEGMENTS UNDER CROSS-VALIDATION FOLD 3. CANDIDATES MUST HAVE A SIMILARITY SCORE ABOVE THE USER'S AVERAGE SIMILARITY SCORE. TO BE SELECTED, CANDIDATES MUST BE WITHIN 0.1 OF THE MOST SIMILAR SEGMENT IN THE POOL SEL_{max} OR THE CANDIDATE AVERAGE SEL_{avg} − WHICHEVER IS GREATER.

User	Seg.	Exp.	Candidates	Selected	sel_{avg}	sim_{avg}	sel_{min}	sel_{max}
User 3	11710	MT	47	3	0.327	0.227	0.236	0.530
User 3	11710	TM	65	21	0.307	0.218	0.221	0.447
User 4	13421	MT	49	5	0.347	0.227	0.230	0.542
User 4	13421	TM	261	14	0.369	0.229	0.230	0.607

TABLE VI
COEFFICIENTS FOR THE LINEAR REGRESSION MODEL FEATURES LISTED IN TABLE I. FEATURES FOR THE CURRENT SEGMENT ARE LABELED "CURR". FEATURES COMPUTING THE DIFFERENCE BETWEEN THE MT AND TM AGGREGATIONS ARE LABELED "DIFF".

User	WC_{SR}				WC_{TG}			WC_{PE}		
	Curr	TM	MT	Diff	TM	MT	Diff	TM	MT	Diff
1	5.485	-3.580	-2.232	-2.850	-1.419	3.869	0.000	4.584	-2.908	4.646
2	5.398	8.582	-7.832	11.014	-5.526	3.136	-5.515	-2.570	3.949	-3.382
3	3.516	1.427	-5.051	0.000	-1.114	1.849	0.000	-1.634	3.793	0.000
4	2.233	0.000	0.000	-3.156	0.000	0.000	0.000	0.000	0.000	3.061
User		TFW			TTE			TER_{TG}		TER_{SR}
	Intercept	TM	MT	Diff	TM	MT	Diff	TM	MT	Curr
1	-25.742	0.000	1.574	0.000	-0.134	0.314	0.242	0.158	0.000	0.144
2	-21.001	0.000	2.580	0.000	-0.387	0.412	0.483	0.000	0.000	0.000
3	-16.399	0.000	1.570	1.210	0.000	-0.167	-0.223	0.134	0.000	0.000
4	5.263	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

TABLE V
PERFORMANCE OF BASELINE VERSUS AGGREGATED SYSTEM AFTER FULL TRAINING OF LINEAR REGRESSION MODELS FOR EACH USER, EVALUATED ON THE TEST SET. CORRELATION (CORR.), MEAN ABSOLUTE ERROR (MAE), ROOT MEAN SQUARE ERROR (RMSE).

		Baseline			Aggregated		
User	Instances	Corr.	MAE	RMSE	Corr.	MAE	RMSE
1	40	0.6551	29.54	47.19	**0.6682**	28.16	46.31
2	38	**0.7359**	23.57	33.20	0.7290	23.13	33.81
3	40	0.7001	13.01	17.19	**0.7466**	12.53	16.92
4	34	**0.6702**	11.81	15.81	0.6270	11.95	16.49

a) Important features.: For all users, the average time-for-word (TFW) feature for similar TM segments was not useful. In addition, the average TER for MT segments and the average TER difference were not used. Naturally, the word count of the current segment was proportional to the TTE. Likewise, the average word counts and their computed difference were significant features. The reverse TER score (TER_{SR}) was not significant. All other features were significant.

b) Example predictions.: Table VII shows time-to-edit (TTE) prediction results for User 3. While IDs #15, #18, #19 have a relatively good error bound, outliers such as #32 drive up the RMSE. Incidentally, the MT output provided for the 5-word source segment in #32 was "Die andere MLV ausgewählt wird.", which only requires one word swap operation.

While the MT output aligns closely with the user's draft translation in ID #21, the regression model provided an much higher estimate. In this example, a total of 170 MT candidates with similarity scores above the user average (22.65%) were available, providing a group average of 38.46%. However, only one segment was selected (94.28% similar). No other candidates are within 10% of the highest candidate; thus, the MT statistics were unreliable. We further note that of 1488 training instances, 512 segments had only one similar MT segment in the aggregation; 75 segments had none (using global user averages instead).

B. Recommendation based on time to edit

Given the TTE predictions from our regression models, we provide recommendations on whether to provide MT suggestions to the translator. Figure 1 lists the confusion matrices for each user's regression model and Table VIII lists the precision, recall, and F-measure for each user.

	T_a	F_a
T_e	21	6
F_e	8	5

(a) User 1

	T_a	F_a
T_e	14	8
F_e	6	10

(b) User 2

	T_a	F_a
T_e	25	7
F_e	5	3

(c) User 3

	T_a	F_a
T_e	9	10
F_e	6	9

(d) User 4

Fig. 1. Confusion matrices for each user. T_a and F_a correspond to the actual values. T_e and F_e correspond to the estimated values.

TABLE VIII
PRECISION, RECALL, F-MEASURE FOR EACH USER

User	Precision	Recall	F-Measure
1	77.77	72.41	74.99
2	63.63	70.00	77.13
3	78.25	83.33	80.71
4	47.36	60.00	52.93

Understandably, due to the lack of useful features in User 4's regression model, the overall MT suggestion results are

TABLE VII
SAMPLE TIME-TO-EDIT A SENTENCE (TTE) PREDICTIONS FOR USER 3. "MT?" REFERS TO THE ACTUAL/ESTIMATED RESULT OF THE CLASSIFICATION MODEL AFTER APPLYING THE DECISION FUNCTION DEFINED IN (2).

ID	Segment	Translation	WC	TTE	Est TTE	Error	MT?
15	In the License Allocations tab you can explicitly add a new eQube-BI TcRA context that can connect to the current license server.	In der Registerkarte Lizenzzuweisung können Sie explizit einen neuen eQube - BI TCRA Kontext hinzufügen, der sich mit dem aktuellen Lizenzserver verbinden kann.	22	66	70.12	4.12	1/1
18	You need to assign roles to users so that they can perform certain operations.	Sie müssen Rollen Benutzern zuweisen, sodass sie bestimmte Operationen durchfähren können.	14	12	10.48	-1.52	1/1
19	Add Role	Rolle hinzufügen	2	11	10.99	-0.01	0/0
20	Click Submit, if you want to submit the details you entered.	Klicken Sie auf Senden, wenn Sie die Details übermitteln möchten, die Sie eingegeben haben.	11	20	26.97	6.97	1/0
21	Click the role on the left side of the screen, to which you want to assign operations.	Klicken Sie auf die Rolle auf der linken Seite des Bildschirms, der Sie Operationen zuweisen möchten.	17	17	36.94	19.94	1/1
32	The other MLV gets selected.	Die andere MLV wird ausgewählt.	5	66	18.64	-47.36	0/1

low. For the other three users, we report higher F-measures that suggest a correlation between the estimated and actual TTE scores in terms of our goodness measure. However, what does this say about our goodness measure? The right-most column of Table VII lists the actual vs. estimated predictions for the example segments translated by User 3. ID #20 consists of 11 words having a TTE difference of approximately 7 seconds. Looking more closely, for User 3, $\hat{\mu}_{TM} < \hat{\mu}_{MT}$, which implies that MT is not useful for any segments of this length ($\hat{\mu}_{TM} = 2.47$, $\hat{\sigma}_{TM} = 1.51$, $\hat{\mu}_{MT} = 2.69$, $\hat{\sigma}_{TM} = 3.80$). This underlies the importance of significance testing as one of the missing components in our decision function.

CONCLUSION AND FUTURE WORK

In conclusion, we address the problem of quality estimation for machine translation in the CAT scenario by constructing regression models tailored to each translator in order to estimate the productivity of a user. We estimate user productivity in terms of the time taken to edit a translation (TTE). We combine features from the current segment with aggregated features from similar segments in two field test experiments. We find that a trained regression model predicts TTE reasonably well given a limited data set drawn from the preliminary MateCat field test outlined in [1], with exceptions explained by user inconsistencies and limitations in the data captured.

The estimated TTE values for each segment in our test set is compared against the mean TTE values for similarly long segments in the TM and MT sub-experiments. Segments whose TTE is closer to the MT experiment's mean than its TM counterpart are judged to indicate that suggesting machine translation output will improve user productivity. We evaluate this decision function against the actual TTE values to measure the consistency of the regression model. After careful evaluation, we see that this heuristic is deficient in accurately suggesting MT output to the translator in cases where the population means of similarly long TM and MT segments are close. As such, the choice of a decision function should be revisited.

One potential source of problems in the regression model is that each segment contains a limited number of content words. In practice, the content words are the biggest determiners of coherence in a text. Thus, we propose to add additional features based on the content words to our regression model. Additionally, we propose to add the average similarity scores and the average word length between the current segment and the aggregated TM and MT segments.

ACKNOWLEDGMENTS

This work is partially funded by the European Commission under the FP7 project MateCat, Grant 287688. The authors wish to thank Georgia Koutrika for her valuable suggestions in this experiment.

REFERENCES

[1] M. Federico, A. Cattelan, and M. Trombetti, "Measuring User Productivity in Machine Translation Enhanced Computer Assisted Translation ," in *AMTA 2012*, San Diego, California, October 2012.

[2] L. Specia, M. Turchi, Z. Wang, J. Shawe-Taylor, and C. Saunders, "Improving the confidence of machine translation quality estimates," in *Machine Translation Summit XII*, Ottawa, Canada, 2009.

[3] C. Buck, "Black box features for the WMT 2012 quality estimation shared task," in *Proceedings of the Seventh Workshop on Statistical Machine Translation*. Montreal, Canada: Association for Computational Linguistics, June 2012.

[4] C.-Y. Lin and F. J. Och, "Orange: a method for evaluating automatic evaluation metrics for machine translation," in *Proceedings of Coling 2004*. Geneva, Switzerland: COLING, Aug 23–Aug 27 2004, pp. 501–507.

[5] R. Soricut, N. Bach, and Z. Wang, "The SDL Language Weaver Systems in the WMT12 Quality Estimation Shared Task," in *Proceedings of the Seventh Workshop on Statistical Machine Translation*. Montréal, Canada: Association for Computational Linguistics, June 2012, pp. 145–151. [Online]. Available: http://www.aclweb.org/anthology/W12-3118

[6] R. Soricut and A. Echihabi, "TrustRank: Inducing Trust in Automatic Translations via Ranking," in *ACL*, 2010, pp. 612–621.

[7] N. Bach, F. Huang, and Y. Al-Onaizan, "Goodness: a method for measuring machine translation confidence," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, ser. HLT '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 211–219. [Online]. Available: http://dl.acm.org/citation.cfm?id=2002472.2002500

[8] Y. He, Y. Ma, J. van Genabith, and A. Way, "Bridging SMT and TM with Translation Recommendation," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Uppsala, Sweden: Association for Computational Linguistics, July 2010, pp. 622–630. [Online]. Available: http://www.aclweb.org/anthology/P10-1064

[9] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A study of translation edit rate with targeted human annotation," in *In Proceedings of Association for Machine Translation in the Americas*, 2006, pp. 223–231.

[10] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *ACL*, 2007.

[11] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer, "The mathematics of statistical machine translation: Parameter estimation," *Computational Linguistics*, vol. 19, no. 2, pp. 263–312, 1993. [Online]. Available: http://aclweb.org/anthology-new/J/J93/J93-2003.pdf

[12] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, Nov. 2009. [Online]. Available: http://doi.acm.org/10.1145/1656274.1656278

[13] J. R. Quinlan, "Learning with continuous classes," in *Proceedings of the 5th Australian Joint Conference on Artificial Intelligence*. World Scientific, 1992, pp. 343–348.