Improving the Confidence of Machine Translation Quality Estimates

Lucia Specia and **Craig Saunders**
Xerox Research Centre Europe
Meylan, 38240, France
lucia.specia@xrce.xerox.com
craig.saunders@xrce.xerox.com

Marco Turchi
Department of Engineering Mathematics
University of Bristol
Bristol, BS8 1TR, UK
Marco.Turchi@bristol.ac.uk

Zhuoran Wang and **John Shawe-Taylor**
Centre for Computational Statistics and Machine Learning
University College London
London, WC1E 6BT, UK
jst@cs.ucl.ac.uk
z.wang@cs.ucl.ac.uk

Abstract

We investigate the problem of estimating the quality of the output of machine translation systems at the sentence level when reference translations are not available. The focus is on automatically identifying a threshold to map a continuous predicted score into "good" / "bad" categories for filtering out bad-quality cases in a translation post-edition task. We use the theory of Inductive Confidence Machines (ICM) to identify this threshold according to a confidence level that is expected for a given task. Experiments show that this approach gives improved estimates when compared to those based on classification or regression algorithms without ICM.

1 Introduction

Computer-aided translation (CAT) tools like translation memories and electronic dictionaries have long been used to improve productivity of professional translators. On the other hand, machine translation (MT) systems, and particularly statistical machine translation (SMT), only recently have started to attract language-service providers' and translators' attention. As any other CAT tool, MT is seen as an instrument to save translators' time. As with translation memories, the usual workflow is to apply an MT system and then manually post-edit the translation to correct mistakes.

It is nowadays easy to set-up an SMT system from existing tools and parallel data. Moreover, improvements in the average quality of such systems have been observed in the last years (Callison-Burch et al., 2008). However, there is no guarantee that a given translated segment will be good enough for post-edition. Human translators need to read the segment many times to find out that it is better to delete it and start from scratch. The time spent to read a translation and attempt to post-edit it before dropping it out may be even longer than the time to translate the source sentence from scratch. The lack of information about the quality of an SMT system's output is certainly one of the reasons hampering the use of these systems. The research area addressing this problem is referred to as Confidence Estimation (Blatz et al., 2003).

Our target scenario is that of professional translators post-editing MT segments. In that scenario, the simplest and possibly most effective form of a segment-level quality estimate is a binary "good" or "bad" score, where translations judged as "bad" are not suggested for post-edition. We propose a score that is estimated using a machine learning technique from a collection of information sources and translations annotated according to 1-4 quality scores. A regression algorithm produces a continuous score, which is then thresholded into the two classes to filter out "bad" translations. Differently from previous work, we define this threshold dynamically by estab-

lishing a confidence level that is expected from the models. This is done using the theory of Inductive Confidence Machines (Papadopoulos et al., 2002) to introduce an extra layer of confidence verification in the models. This verification allows tuning the threshold according to the translators' needs. For example, for very experienced and fast translators, usually only very good-quality translations are better than translating from scratch, while medium-quality translations could already be helpful for other translators. We show that this yields better results than thresholding the continuous scores according to the true quality score or using binary or multi-class classifiers to directly estimate discrete "bad" / "good" or 1-4 scores.

In the remainder of this paper we first discuss the previous work on CE for MT (Section 2), then describe our experimental setting (Section 3), the method used to estimate the CE scores (Section 4) and the method threshold them based on an expected confidence level (Section 5). We finally present and discuss the results obtained (Sections 6).

2 Related Work

The task of Confidence Estimation (CE) for MT is concerned with predicting the quality (e.g., fluency or adequacy, post-editing requirements, etc.) of a system's output for a given input, without any information about the expected output. We distinguish, therefore, the task of CE from that of automatic MT evaluation by the need, in the latter, of reference translations.

Although not directly comparable, some of the metrics proposed for sentence-level MT evaluation also exploit learning algorithms and sometimes similar features to those used in CE. Kulesza and Shieber (2004) use a classifier with n-gram precision and other reference-based features to predict if a sentence is produced by a human translator (presumably good) or by an MT system (presumably bad). Albrecht and Hwa (2007a; 2007b) rely on regression algorithms and (pseudo-)reference-based features to measure the quality of sentences. *Pseudo-references* are produced by alternative MT systems, instead of humans, but this scenario with multiple MT systems is different from that of CE envisaged in our work.

Blatz et al. (2004) train regressors and classifiers for CE on features extracted for translations tagged according to MT metrics like NIST (Doddington, 2002). NIST scores are thresholded to label the 5th or 30th percentile of the examples as "good". However, there is no reason to believe that exactly the top 5% or 30% of translations are good.

Quirk (2004) uses classifiers and a pre-defined threshold for "bad" / "good" translations considering a small set of translations manually labelled for quality (350 sentences). Models trained on this dataset outperform those trained on a larger set of automatically labelled data.

Gamon et al. (2005) train a classifier using linguistic features extracted from machine and human translations to distinguish between these two types of translations. The predictions obtained have very low correlation with human judgements, which is an indication, as discussed by (Albrecht and Hwa, 2007a), that high human-likeness does not necessarily imply good MT quality.

Our work differs from previous approaches in several respects, including the addition of new features that were found to be very relevant, the exploitation of multiple datasets of translations from different MT systems, through the use of resource-independent features and the definition of system-independent features, and the use of a feature selection procedure that enables identifying relevant features in a systematic way. More importantly, the main contribution of this paper is the use of Inductive Confidence Machines to dynamically define the threshold to filter out bad translations under a certain expected level of confidence.

3 Experimental Setting

3.1 Features

A number of features have been used in previous work for CE (see (Blatz et al., 2003) for a list). In this paper we focus on features that do not depend on any aspect of the translation process, that is, which can be extracted from any MT system, given only the input (source) and translation (target) sentences, and possibly monolingual or parallel corpora. We call these "black-box" features.

The decision to use only black-box features aims to allow performing the task of CE across different MT systems, which may use different frameworks,

to which we may not have access. It was also motivated by our observation, in previous work, that for the language pair addressed in this paper, more elaborated (and computationally more costly) features do not yield significant gains in performance (Specia et al., 2009).

We extract all linguistic resource- and MT system- independent features that have been proposed in previous work, and also some new features. In what follows, we describe our 77 features, grouped for space reasons. A '*' is used to indicate new features with respect to previous work on CE.

- source & target sentence lengths and their ratios
- source & target sentence 3-gram language model probability & perplexity
- source & target sentence type/token ratio
- source sentence 1 to 3-gram frequency statistics in a given frequency quartile of a monolingual corpus
- alignment score for source and target and percentage of different types of word alignment, as given by GIZA++.
- percentages and mismatches of many superficial constructions between the source and target sentences (brackets, quotes and other punctuation symbols, numbers, etc.)
- *average number of translations per source word in the sentence (as given by probabilistic dictionaries), unweighted or weighted by the (inverse) frequency of the words
- *Levenshtein edit distance between the source sentence and sentences in the corpus used to train the SMT system
- *source & target percentages of numbers, content-words and non-content words
- *POS-tag target language model, based on the target side of the corpus used to train the SMT system.

3.2 Data

We use translation data produced by three phrase-based SMT systems: Matrax (Simard et al., 2005), Portage (Johnson et al., 2006) and Sinuhe (Kaariainen, 2009). The systems are trained on approximately 1 million sentence pairs from the Europarl English-Spanish parallel corpus provided by WMT-08 (Callison-Burch et al., 2008) and used to translate 4K Europarl sentences from the development and test sets also provided by WMT-08.

Metric	Matrax	Portage	Sinuhe
Human	2.5081	2.8345	2.5581
BLEU	0.3241	0.3880	0.3521
NIST	8.4041	8.8586	8.3985
TER	49.543	47.090	49.624
METEOR	0.2397	0.2824	0.2528

Table 1: Average sentence-level human score and corpus-based MT evaluation metrics for all datasets

For each system, translations are manually annotated by professional translators with 1-4 quality scores, which are commonly used by them to indicate the quality of translations with respect to the need for post-edition:

1. requires complete retranslation
2. post editing quicker than retranslation
3. little post editing needed
4. fit for purpose

Table 1 shows the overall quality of the translations in such datasets, as given by the average human annotation (i.e., scores 1-4) and common evaluation MT metrics: BLEU (Papineni et al., 2002), NIST (Doddington, 2002), TER (Snover et al., 2006) and METEOR (using the lemmas of the words) (Lavie and Agarwal, 2007).

The feature vector for each dataset is randomly subsampled five times in training (40%), validation (40%) and test (20%) using a uniform distribution.

We also performed experiments in which the quality score used to annotate sentences was the time spent by a profession translator post-editing a sentence. However, we found a very large variability in the post-edition time for sentences with similar sizes and quality, even by the same translator. Therefore, this type of annotation did not yield reliable results.

3.3 Partial Least Squares Regression

In order to predict the sentence-level 1-4 scores, we use Partial Least Squares (PLS) (Wold et al., 1984). PLS projects the original data onto a different space of latent variables (or "components"). It can be defined as an ordinary multiple regression problem,

i.e., $Y = XB_w + F$, where X is a matrix of input variables, Y is a vector of response variable, B_w is the regression matrix, F is the residual matrix, but B_w is computed directly using an optimal number of components. When X is standardized, an element of B_w with large absolute value indicates an important X-variable.

To evaluate the performance of the approach, we compute the average error in the estimation of the human scores by means of the Root Mean Squared Prediction Error (RMSPE) metric:

$$\sqrt{\frac{1}{N}\sum_{j=1}^{N}(y_j - \widehat{y}_j)^2}$$

where N is the number of test cases, $\widehat{y}$ is the prediction obtained by the regressor and y is the true score of the test case. RMSPE quantifies the amount by which the estimator differs from the true score.

4 Predicting the CE Score

We use the technique of PLS, as described in Section 3.3, to estimate the CE score in [1,4]. We take advantage of a property of PLS, which is the ordering of the features of X in B_w according to their relevance, to select subsets of discriminative features. The method to perform regression supported by an embedded feature selection procedure consists of the following steps:

1. Given each possible number of components (from 1 to the maximum number of features), we run PLS to compute the B_w matrix on 50% of the validation data, generating a list L of features ranked in decreasing order of importance.
2. Given the list L produced for a certain number of components, we re-train the regression algorithm on 50% of the validation data, adding features from L one by one. We test the models on the remaining validation data and plot learning curves. By analyzing the learning curves, we select the first n features that minimize the error of the models.
3. Given the selected n features and the number of components that minimizes the error in the validation data, we train PLS on the training dataset and test the performance of the regressor using these features on the test dataset, computing the corresponding RMSPE.

This is repeated five times for each of the subsamples of the original dataset, and the average error is computed.

As we have shown in previous work (Specia et al., 2009), the use of PLS with feature selection to estimate a continuous score allows considerable gain in performance as compared to PLS or other regression methods without this step.

5 Controlling the Acceptance Threshold via Conformal Prediction

The user scenario investigated in this paper requires distinguishing between only two classes of translations: "good" and "bad". The main problem addressed here is thus how to choose a threshold to categorize the regression predictions in such classes. In this particular scenario, one might want to prioritize precision or recall, depending on whether it is preferable to select a small set of good quality translations for post-edition or a larger set of doubtful translations. We define a mechanism to control the threshold for the two classes which is based on the expected confidence level of the predictions. This results in a mechanism to control the precision of the CE models, and different choices yield a natural trade-off between precision and recall.

The theory of conformal prediction (Vovk et al., 2005), whose models are also referred to as *confidence machines*, is adopted to deal with this problem. We are especially interested in the inductive versions of the confidence machines as introduced by Papadopoulos et al. (2002).

The proposed approach is to search for a threshold value such that $1 - \delta$ ($0 < \delta < 1$) of the examples whose predicted scores that are equal to or greater than the threshold are indeed acceptable, i.e. their true scores are greater than or equal to a pre-fixed value, e.g. $y \geq 3$. Here, δ is called the *significance level*, while $1 - \delta$ is called the *confidence level*, for example, $\delta = 0.1$ corresponds to a 90% confidence level. Such a threshold value can be obtained via a binary search among the regression predictions for examples in a calibration dataset, as shown in Algorithm 1. We call this threshold the *confidence threshold* and denote it by ρ, whilst the prefixed true score to identify a good translation is called *acceptance threshold* and is denoted by τ.

Algorithm 1: Search for confidence threshold

1 **input:** $\hat{\mathbf{y}}$ regression predictions
$\mathbf{y}$ true scores
τ acceptance threshold
δ significance level
2 $L \leftarrow \min(\hat{\mathbf{y}}), U \leftarrow \max(\hat{\mathbf{y}})$;
3 $\mathbf{s} \leftarrow \{i | L \leq \hat{y}_i \leq U\}$;
4 $\rho \leftarrow \text{median}(\hat{\mathbf{y}}_\mathbf{s})$;
5 $\hat{\delta} \leftarrow \frac{|\{i|\hat{y}_i \geq \rho,\ y_i < \tau\}|}{|\{j|\hat{y}_j \geq \rho\}|}$;
6 **if** $\hat{\delta} = \delta$ **or** $L = U$
7 **return** ρ;
8 **else if** $\hat{\delta} < \delta$
9 $L \leftarrow \rho$, **goto** 3;
10 **else**
11 $U \leftarrow \rho$, **goto** 3;

The theory of inductive confidence machines shows that the confidence threshold guarantees the confidence level on unseen data.

5.1 Inductive Confidence Machines

In order to search for the confidence threshold ρ for a given acceptance threshold τ, with a significance level δ, we use a training set $S := \{(x_i, y_i)|i = 1, \ldots, l\}$, which is split into two sets: a proper *training set* $S_T := \{(x_i, y_i)|i = 1, \ldots, m\}$ with $m < l$ examples and a *calibration set* $S_C := \{(x_{m+i}, y_{m+i})|i = 1, \ldots, k\}$ with $k := l - m$ examples. The original regression model is trained on the proper training set, and tested on the calibration set to obtain calibration predictions $\hat{y}_{m+i}$ for $i = 1, \ldots, k$.

A strangeness measure function $\alpha(\hat{y}, y)$ is then defined to associate a correctness score for every prediction $\hat{y}$, obtaining the p-value expression for a new example, denoted as (x_{l+1}, y_{l+1}):

$$p(y) := \frac{|\{i|1 \leq i \leq k+1,\ \alpha_i \geq \alpha_{k+1}\}|}{k+1} \quad (1)$$

where we use α_i to represent $\alpha(\hat{y}_{m+i}, y_{m+i})$ for short.

For any threshold value ρ, we only consider those examples whose $\hat{y} \geq \rho$. Hence, we redefine the "active" calibration set to be $S_C^* := \{(x_i^*, y_i^*)|1 \leq i \leq n\} := \{(x_{m+i}, y_{m+i})|1 \leq i \leq k,\ f(x_i) \geq \rho\}$, where we assume the regression problem to be expressed by $\hat{y} = f(x)$. We then define our strangeness measure α to be:

$$\alpha(\hat{y}^*, y^*) := \text{sgn}(\tau - y^*) \cdot (\hat{y}^* - \rho) \quad (2)$$

where $\text{sgn}(z)$ returns $+1$ if $z \geq 0$, and -1 otherwise. Computing the p-value according to Eq. (1) based on the α defined above implies Line 5 in Algorithm 1. For new examples, we also only consider those (x^*, y^*) that have $f(x^*) \geq \rho$.

5.2 Validity of the P-value

To prove the validity of the p-value in Eq. (1), we assume that the calibration examples and new examples are independently and identically distributed (i.i.d.) according to a fixed distribution P. Accordingly, we will have the active validation examples $(x_1^*, y_1^*), \ldots, (x_n^*, y_n^*)$ and a new example (x_{n+1}^*, y_{n+1}^*) i.i.d. drawn from the conditional distribution $P^* := P\{(x, y)|f(x) \geq \rho\}$.

For any distribution P and any significance level δ, a valid p-value satisfies $P\{p(y) \leq \delta\} \leq \delta$. The methodology used by Papadopoulos et al. (2002) can be employed to prove the validity of our p-value. We assume that the sequence $(x_1^*, y_1^*), \ldots, (x_{n+1}^*, y_{n+1}^*)$ is generated from a bag, i.e., an unordered set $\lbag (x_1^*, y_1^*), \ldots, (x_{n+1}^*, y_{n+1}^*) \rbag$, by assigning a permutation to it. There will be $(n+1)!$ possible permutations.

The probability of the very example (x_{n+1}^*, y_{n+1}^*) being selected as the $(n+1)$th (i.e. the new) example is $\frac{n!}{(n+1)!} = \frac{1}{n+1}$. As $p(y_{n+1}^*) \leq \delta$ if and only if $\alpha(\hat{y}_{n+1}^*, y_{n+1}^*)$ is among the $\lfloor \delta(n+1) \rfloor$ largest $\alpha(\hat{y}_i^*, y_i^*)$, the probability $P\{p(y_{n+1}^*) \leq \delta\} = \frac{1}{n+1}\lfloor \delta(n+1) \rfloor \leq \delta$, since all the $(n+1)!$ permutations are equally probable.

6 Results

6.1 PLS Regression

Table 2 shows the performance obtained by PLS (without ICM) for all datasets annotated with 1-4 scores. The figures for the subsets of features consistently outperform those for using all features and are also more stable (lower standard deviations). Using only the selected features, predictions deviate on average ~ 0.618-0.68 from the true score. Although it is not possible to compare these results to previous

Dataset	RMSPE	RMSPE all features
Matrax	0.680 ± 0.007	1.261 ± 0.760
Portage	0.618 ± 0.016	0.719 ± 0.094
Sinuhe	0.669 ± 0.016	1.203 ± 0.839

Table 2: RMSPE for all datasets

studies, since different datasets are used, we consider them to be satisfactory if the predictions are to be used as such, to provide the end user with an idea about the quality of the translations. In that case, the observed error would yield on average crossing one adjacent category in the 1-4 ranking.

By looking at the features selected by PLS, we can highlight the following features appearing as top in all datasets:

- source & target sentence 3-gram language model probabilities;
- source & target sentence lengths and their ratio;
- *percentages of cardinalities of word alignments (1-1, 1-n, etc.);
- *percentage and mismatch in the numbers and punctuation symbols in the source and target;
- *ratio of percentage of a-z tokens in the source and target;
- percentage of unigrams seen in the corpus.

In general, the top features indicate the difficulty of translating the source sentence (because it is long, ambiguous, or not commonplace, for example) or some form of mismatch between source and target sentences. Many of these features had not been used before for CE (marked with '*' here). Interestingly, apart from the target length and language model, these features are not part of the set used in standard SMT models.

6.2 Inductive Confidence Machines

As previously mentioned, the models produced for different datasets using PLS deviate ~ 0.618-0.68 points when predicting sentence-level 1-4 scores, which can be considered a satisfactory deviation if the scores are used as such, but may have a strong negative impact if the scores are thresholded for filtering out the bad translations, as necessary in the scenario we address in this paper. Since this is an average error, there might be cases where a sentence that should be scored as "requires complete retranslation" (score 1) will be predicted as "a little post editing needed" (score 3) and sent for post-edition, delaying the translation process. We applied the method described in Section 5 to minimize the number of such cases.

We use the predictions found by PLS for the validation dataset as the ICM "calibration" set to find a good confidence threshold and then apply it on the test set to split the test cases into "bad" and "good" translations. The first step is to establish the true acceptance threshold (τ) for a given task, that is, the true score under which translations should be considered "bad". This threshold is usually based on the language-pair, text domain and possibly the level of experience of translators, e.g., the more experienced the translator, the higher the acceptance threshold for a translation to be useful. Based on this threshold, the resulting performance can be quantified in terms of the "good" cases kept for post-edition (that is, cases scored above the true acceptance threshold), using *precision* (number of cases correctly classified as "good" divided by number of cases classified as "good"), *recall* (number of cases correctly classified as "good" divided by total number of "good" cases) and *f-measure* (harmonic mean of precision and recall).

Figure 1 shows the precision and f-score of the three MT systems on a test set of 800 examples, considering τ = 3 and expected confidence levels from 98% to 80% (i.e., δ = 0.02 to 0.2). This figure shows that by using ICM it is possible to control the required CE precision for a particular task by setting different expected confidence levels: the precision obtained is linear to that confidence level. It also shows that by decreasing the confidence level, it is possible to significantly improve the f-score (as a consequence of improving recall). The curves show the same behavior for the three MT systems, and their precision is comparable. However, recall (and consequently f-measure) is significantly higher for Portage, followed by Sinuhe. This may be explained by the differences in the overall quality of these MT systems. According to human and automatic MT evaluation metrics, the ranking of the systems for the datasets used here is the same as in this figure: Portage, Sinuhe and Matrax. In fact, we believe the quality of the MT system may influence the quality

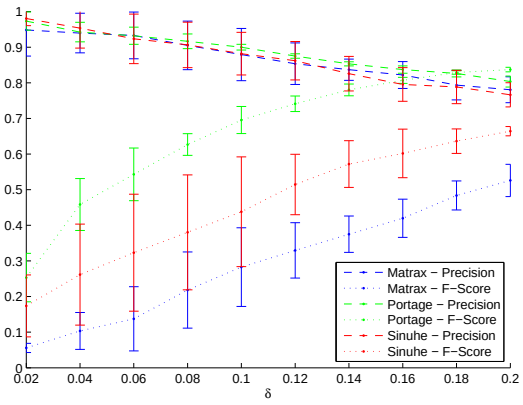

Figure 1: Precision and F-Score of PLS + ICM for τ = 3 and different confidence levels

τ	Model	Precision	Recall
2	PLS	92.86±0.81	91.84±0.81
	ICM δ = 0.05	95.31±0.77	73.01±5.67
	ICM δ = 0.1	91.37±0.97	99.43±1.13
3	PLS	91.45±1.81	25.01±3.54
	ICM δ = 0.05	94.22±7.10	17.46±1.23
	ICM δ = 0.1	88.18±5.97	30.68±1.27

Table 3: Comparison between performance of PLS and PLS+ICM in Sinuhe (confidence levels = 95% and 90%)

of the CE estimates: the CE task becomes easier for an MT system which produces very good (or very bad) translations most of the times.

With τ = 2, results are similar, but the precision and recall curves flatten from δ above 0.1, with f-measure of approximately 95% for all systems. This behavior is observed because for τ = 2 the CE problem becomes "easier", since the vast majority of the translations are scored equal or above 2.

Table 3 compares the performance obtained by PLS and the combination of PLS with ICM for one of the datasets (Sinuhe), with confidence levels of 95% and 90% and the acceptance thresholds (τ) of interest in this paper (2 and 3). It shows that for higher expected confidence levels (δ = 0.05) for both acceptance thresholds, ICM guarantees higher precision, while recall drops, as compared to PLS alone. On the other hand, when lower confidence levels are expected (δ = 0.1), ICM guarantees higher recall, but has lower precision. The difference between PLS and PLS+ICM for the other datasets is comparable.

The guaranteed confidence levels are higher for lower τs, since these make the problem easier. For higher τs, in order to try to guarantee the expected confidence level, a larger proportion of positive examples need to discarded. This is in line with the post-edition scenario that we are targeting in this paper, where a higher τ and higher confidence levels are aimed at more experienced translators. In fact, for most professional translators, translations are only expected to be useful for post-edition if they require little retranslation (scored equal or above 3).

τ	SVM 4-classes	SVM binary	PLS+ICM
2	91.60±0.62	90.99±0.95	90.88±1.26
3	50.09±1.61	69.19±0.97	87.94±7.33

Table 4: Comparison between precision of SVM classifiers and PLS+ICM for Matrax (confidence level = 90%)

Focusing on such experienced translators, in Table 4 we compare the precision of the PLS+ICM against using two versions of a Support Vector Machines (SVM) classifier (Joachims, 1999) to predict a discrete score:

- A binary version to directly classify the test cases into "good" or "bad", by considering scores equal or above 2 as "good" for comparison with τ = 2, and equal or above 3 as "good" for comparison with τ = 3.
- A multi-class implementation of SVM to predict the 1-4 categories which are then thresholded for comparison with τ = 2 or τ = 3.

For τ = 2, both SVM and PLS+ICM have similar precisions, but τ = 3, PLS+ICM guarantees a much higher level of confidence. Results for other datasets are comparable to these.

7 Discussion and conclusions

We proposed a method for further improving the quality estimates produced for machine translations at the sentence level. Focusing on a scenario where a binary score is necessary for filtering out "bad" translations, we applied the theory of Inductive Confidence Machines to allow controlling the expected level of confidence (precision) of the scores predicted using a regression algorithm. This was done by dynamically establishing a threshold to categorize translations into "bad" or "good" classes based

on such confidence level. With translation datasets produced by different MT systems, we showed that this method improves results over regression and classification algorithms, allowing for better precision or recall, depending on the translation quality required. The method allows control the expected precision (and as a consequence, recall) according to the needs of a certain translation task, that is, whether it is better to keep a smaller number of very likely to be good translations for post-edition or a larger number of possibly doubtful translations.

We plan now to train ICMs with multiple parameters by reformulating it to an optimization problem that guarantees an expected confidence level while maximizing the number of accepted cases, that is, improving precision and recall at the same time.

Acknowledgments

This work was supported by the European Commission under the IST Project SMART (FP6-033917).

References

J. Albrecht and R. Hwa. 2007a. A re-examination of machine learning approaches for sentence-level mt evaluation. In *45th Meeting of the Association for Computational Linguistics*, pages 880–887, Prague.

J. Albrecht and R. Hwa. 2007b. Regression for sentence-level mt evaluation with pseudo references. In *45th Meeting of the Association for Computational Linguistics*, pages 296–303, Prague.

J. Blatz, E. Fitzgerald, G. Foster, S. Gandrabur, C. Goutte, A. Kulesza, A. Sanchis, and N. Ueffing. 2003. Confidence estimation for machine translation. Technical report, Johns Hopkins University, Baltimore.

J. Blatz, E. Fitzgerald, G. Foster, S. Gandrabur, C. Goutte, A. Kulesza, A. Sanchis, and N. Ueffing. 2004. Confidence estimation for machine translation. In *20th Coling*, pages 315–321, Geneva.

C. Callison-Burch, C. Fordyce, P. Koehn, C. Monz, and J. Schroeder. 2008. Further meta-evaluation of machine translation. In *3rd Workshop on Statistical Machine Translation*, pages 70–106, Columbus.

G. Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *2nd Human Language Technology Research*, pages 138–145, San Diego.

M. Gamon, A. Aue, and M. Smets. 2005. Sentence-level mt evaluation without reference translations: beyond language modeling. In *10th Meeting of the European Association for Machine Translation*, Budapest.

T. Joachims, 1999. *Advances in Kernel Methods - Support Vector Learning*, chapter Making large-scale SVM learning practical. MIT Press.

H. Johnson, F. Sadat, G. Foster, R. Kuhn, M. Simard, E. Joanis, and S. Larkin. 2006. Portage: with smoothed phrase tables and segment choice models. In *Workshop on Statistical Machine Translation*, pages 134–137, New York.

M. T. Kaariainen. 2009. Sinuhe: Statistical machine translation with a globally trained conditional exponential family translation model. In *EAMT Workshop on Statistical Multilingual Analysis for Retrieval and Translation*, Barcelona.

A. Kulesza and A. Shieber. 2004. A learning approach to improving sentence-level mt evaluation. In *10th International Conference on Theoretical and Methodological Issues in Machine Translation*, Baltimore.

A. Lavie and A. Agarwal. 2007. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *2nd Workshop on Statistical Machine Translation*, pages 228–231, Prague, Czech Republic.

H. Papadopoulos, K. Proedrou, V. Vovk, and A. Gammerman. 2002. Inductive confidence machines for regression. In *13th European Conference on Machine Learning*, Helsinki.

K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *40th Meeting of the Association for Computational Linguistics*, pages 311–318, Morristown.

C. B. Quirk. 2004. Training a sentence-level machine translation confidence measure. In *4th Language Resources and Evaluation*, pages 825–828, Lisbon.

M. Simard, N. Cancedda, B. Cavestro, M. Dymetman, E. Gaussier, C. Goutte, and K. Yamada. 2005. Translating with non-contiguous phrases. In *Empirical Methods in Natural Language*, pages 755–762, Vancouver.

M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *7th Association for Machine Translation in the America*, pages 223–231, Cambridge, MA.

L. Specia, M. Turchi, N. Cancedda, M. Dymetman, and N. Cristianini. 2009. Estimating the sentence-level quality of machine translation systems. In *13th Meeting of the European Association for Machine Translation*, Barcelona.

V. Vovk, A. Gammerman, and G. Shafer. 2005. *Algorithmic Learning in a Random World*. Springer.

S. Wold, A. Ruhe, H. Wold, and W. J. Dunn. 1984. The covariance problem in linear regression. the partial least squares (pls) approach to generalized inverses. *SIAM Journal on Scientific Computing*, 5:735–743.