# UGENT-LT3 SCATE Submission for WMT16 Shared Task on Quality Estimation

**Arda Tezcan** **Véronique Hoste** **Lieve Macken**
LT3, Language and Translation Technology Team
Department of Translation, Interpreting and Communication
Ghent University
Groot-Brittannielaan 45, 9000 Ghent, Belgium
{arda.tezcan, veronique.hoste, lieve.macken}@ugent.be

## Abstract

This paper describes the submission of the UGENT-LT3 SCATE system to the WMT16 Shared Task on Quality Estimation (QE), viz. English-German word and sentence-level QE. Based on the observation that the data set is homogeneous (all sentences belong to the IT domain), we performed bilingual terminology extraction and added features derived from the resulting term list to the well-performing features of the word-level QE task of last year. For sentence-level QE, we analyzed the importance of the features and based on those insights extended the feature set of last year. We also experimented with different learning methods and ensembles. We present our observations from the different experiments we conducted and our submissions for both tasks.

## 1 Introduction

Machine Translation (MT) Quality Estimation (QE) is the task of providing a quality indicator for unseen automatically translated sentences without relying on reference translations (Gandrabur and Foster, 2003; Blatz et al., 2004). The WMT16 QE shared task proposes three evaluation tasks: (1) scoring and ranking sentences according to predicted post-editing effort given a source sentence and its translation; predicting the individual (2a) words and (2b) phrases (segmented by the Statistical Machine Translation (SMT) decoder) that require post-editing; and (3) predicting the quality at document level. In this paper, we describe the UGENT-LT3 SCATE submissions to task 1 (sentence-level QE) and task 2a (word-level QE). By conceiving the QE as a supervised Machine Learning (ML) problem for both tasks, we extended the features that we extracted for our last year's submission (Tezcan et al., 2015), which try to capture the accuracy and fluency errors in MT output. While accuracy is concerned with how much of the meaning expressed in the source is also expressed in the target text, fluency is concerned with to what extent the translation is well-formed. This distinction between accuracy and fluency was suggested to break down human translation quality judgments into separate and smaller units (White, 1995) and is well known in quality assessment schemes for MT (White, 1995; Secară, 2005; Lommel et al., 2014). Similarly, we use the same distinction to break down the QE task into separate units. In addition to the features that try to capture accuracy and fluency errors, given the specialized domain of this year's data set (IT), for word-level QE, we extracted features that try to capture terminological problems. For both tasks, we experimented with different learning methods. For word-level QE we also built ensemble systems that are based on majority voting and bagging (random forests), in which multiple decision trees are constructed using bootstrapped training sets and the predictions of these trees are averaged.

The rest of this paper is organized as follows. Section 2 and Section 3 give an overview of the shared task on word-level QE and sentence-level QE respectively and describe the extracted features, the additional language resources that were used for feature extraction, the learning methods and the experiments that were conducted. Section 4 concludes by discussing the results and observations that were made.

## 2 Word-level Quality Estimation

Similar to the previous year, the word-level QE task in WMT16 is conceived as a binary classification task. The goal is to label translation er-



*Proceedings of the First Conference on Machine Translation*, Volume 2: Shared Task Papers, pages 843–850,
Berlin, Germany, August 11-12, 2016. ©2016 Association for Computational Linguistics

rors at word level by marking words either as OK or BAD. In WMT16, submissions are evaluated in terms of classification performance via the multiplication of F1-scores for the OK and BAD classes against the original labels due to the fact that the F1-score for the BAD class, which has been used as a primary metric in previous years, is biased towards 'pessimistic' labeling. In contrast, the multiplication of F1-OK and F1-BAD has two components and is more balanced.

The organizers provided a data set of English source sentences with the corresponding German MT output, generated by a statistical MT system and the post-edited MT output. This data set consists of a training set of 12,000 sentences, a development set of 1,000 sentences and a test set of 2,000 sentences. As in previous years, the MT output in the training and development data are automatically annotated for errors with binary word-level labels by using the alignments provided by the TER tool (Snover et al., 2006). The distribution of the binary labels and the average sentence length for the training and development sets (in number of tokens) are given in Table 1.

| | # Words | OK | BAD | Length |
|---|---|---|---|---|
| Train | 210958 | 78.5% | 21.5% | 17.57 |
| Dev | 19487 | 80.5% | 19.5% | 19.48 |

Table 1: Number of words, distribution of the binary labels and the average sentence length, on the training and development set.

### 2.1 Features and Language Resources

To characterize each target word of the MT output, in addition to the provided baseline features, which were described in the WMT15 QE shared task (Bojar et al., 2015), we extracted the features[1] we used for our last year's submission, for which detailed descriptions can be found in Tezcan et al. (2015).

Technical texts, like in the IT domain, express concepts in a concise and consistent form and leave little room for data redundancy. This is often achieved with the use a specialized terminology (Rinaldi et al., 2004). As a result, in professional translation services, correct and consistent handling of terminology becomes an important indicator of translation quality (Pinnis, 2015). Given that the data set for the QE-tasks in WMT16 is in the IT domain, we designed three binary features based on the use of terminology, which indicate whether:

- the target word (tw) is part of a term in our bilingual term list;
- the source alignment (sw) of the tw is part of a term in our bilingual term list, given the alignments in the baseline feature set;
- the left or right context word of the sw is part of a term in our bilingual term list, given the alignments in the baseline feature set.

To be able to define these features, we used the bilingual terminology extraction tool TExSIS (Macken et al., 2013) to automatically extract a bilingual term list from the training corpus. Besides additional statistics, TExSIS output provides a frequency ratio for each extracted bilingual term pair, which corresponds to the source/target term frequency in the given data set. We filtered out the bilingual terms with a frequency ratio of less than 0.8 to focus only on the most reliable term pairs. The resulting bilingual term list includes 4198 entries. Examples of the extracted terms are provided in Table 2.

| Source Term (EN) | Target Term (DE) |
|---|---|
| dialog box | Dialogfeld |
| SWF file | SWF-Datei |
| pop-up note | Popup-Notiz |
| export | exportieren |
| exported image | exportierten Bilds |
| cross-references | Querverweise |

Table 2: Examples of bilingual terminology automatically extracted by TExSIS.

Based on this bilingual term list, we marked all entries, starting with the longest term found, in the training, development and test sets and extracted the three binary features mentioned before for each target word in the MT output.

Even though we only used the training set for extracting features relating to terminology, we used additional language resources for the other additional features we extracted (see Tezcan et al. 2015 for more details). These features are based on a surface Language Model (LM) and a Part-of-speech (PoS) LM of the target language, and a

[1] All features that are described in Tezcan et al. (2015) except the features based on named entities and simplified Part-of-Speech (PoS) tags.

Phrase Table consisting of phrase alignments and translation probabilities between the source and target languages. As bilingual data, we used the provided training set, the Autodesk Post-Editing Data[2] and a collection of corpora from OPUS (Tiedemann, 2012) in the IT domain. The number of sentence pairs collected from each corpus is presented in Table 3.

| Corpus | # Segments | # Words (EN-DE) | |
|---|---|---|---|
| WMT16 | 12000 | 201505 | 228549 |
| Autodesk | 124486 | 1411351 | 1382342 |
| Gnome | 28439 | 201634 | 183958 |
| KDE4 | 224035 | 1745841 | 1671591 |
| PHP | 39707 | 228549 | 228434 |
| Ubuntu | 12992 | 70136 | 66348 |
| TOTAL | 441659 | 3859016 | 3747393 |

Table 3: Additional language resources that were used to extract features and the number of segments in each data set.

We used the Moses Toolkit (Koehn et al., 2007) to obtain phrase alignments from the collected data. The phrase alignments were pruned to exclude entries with a direct alignment probability $P(t|s) < 0.01$. We built the LM and PoS LM on the target side of the collected bilingual data. The following preprocessing steps have been applied on the data prior to building the LM and the phrase table: normalization of digits, tokenization and lowercasing. The surface form LM has been built using KenLM (Heafield, 2011). For building the PoS LM, we used TreeTagger (Schmid, 1995) to obtain the PoS tags on the target (DE) data and IRSTLM (Federico et al., 2008) for building the LM. As smoothing technique we used Witten-Bell as the modified Kneser-Ney smoothing, which is used by KenLM, is not well defined when there are no singletons (Chen and Goodman, 1996) and leads to modeling issues on the PoS data. The resulting LMs and phrase table were stored in databases and indexed to speed up lookup operations.

### 2.2 Learning Methods

By combining different learning methods into ensemble systems based on majority voting, we were able to increase the word-level QE performance of individual systems in the past (Tezcan et al., 2015). This has motivated us to experiment with different learning methods and ensembles. In our experiments we used 6 different learning methods: Logistic Regression (LR), Perceptron (PE), Random Forest (RF) and Linear Support Vector Classification (SVC) using the Scikit-learn module in Python (Pedregosa et al., 2011), Conditional Random Fields (CRFs) using the CRF++ Toolkit (Kudo, 2005) and Memory-Based Learning (MBL) using TiMBL (Daelemans et al., 2004). For the algorithms that did not accept categorical features in the Scikit-learn module (such as LR and RF), one-hot encoding was applied to transform the feature sets prior to training.

### 2.3 Experiments

We carried out experiments with the six ML methods and combinations of three different feature sets, namely the baseline features (*b*), the SCATE features we used for WMT15 (*s*) and the new features we extracted, which identify words that appear in the bilingual term list (*t*). We applied hyper-parameter optimization for the ML algorithms (when applicable) using 10-fold cross validation on the training set and tested the classification performance on the development set. All the features were scaled to the [0, 1] range prior to training. The classification performance of different algorithms and feature sets, with respect to F1 scores for the BAD class, the OK class and the multiplication of the two (MLT), are provided in Table 4.

| | | LR | PE | **RF** | SVC | CRF | MBL |
|---|---|---|---|---|---|---|---|
| *b* | BAD | 0.33 | 0.37 | 0.23 | 0.24 | 0.30 | 0.29 |
| | OK | 0.87 | 0.81 | 0.90 | 0.88 | 0.89 | 0.88 |
| | MLT | 0.29 | 0.30 | 0.20 | 0.21 | 0.26 | 0.25 |
| *b+s* | BAD | 0.41 | 0.40 | 0.45 | 0.42 | 0.38 | 0.38 |
| | OK | 0.83 | 0.83 | 0.85 | 0.83 | 0.80 | 0.80 |
| | MLT | 0.34 | **0.33** | 0.38 | **0.35** | 0.30 | 0.30 |
| ***b+s +t*** | BAD | 0.45 | 0.37 | 0.45 | 0.43 | 0.44 | 0.39 |
| | OK | 0.83 | 0.85 | 0.86 | 0.82 | 0.82 | 0.81 |
| | MLT | **0.37** | 0.31 | **0.39** | **0.35** | **0.36** | **0.32** |

Table 4: The performance of different ML algorithms and feature sets on the development set. The plus sign '+' indicates the combined feature sets.

Under the hypothesis that different learners make different types of errors, we first analyzed the amount of disagreement by comparing the output of each system using the overall best feature

[2]https://autodesk.app.box.com/v/autodesk-postediting

set '*b+s+t*'.

| | CRF | LR | PE | SVC | RF |
|---|---|---|---|---|---|
| MBL | 21% | 19% | 29% | 20% | 21% |
| CRF | | 5% | 19% | 3% | 19% |
| LR | | | 21% | 4% | 18% |
| PE | | | | 19% | 30% |
| SVC | | | | | 18% |

Table 5: The disagreement ratios between the predicted labels by different algorithms (feature set '*b+s+t*').

Based on the disagreement ratios between the different ML systems given in Table 5, we built two ensemble systems by combining individual ML systems with high disagreement ratios (low correlation) that vote for the final output, which is defined by the majority vote. The two ensemble systems and their performances on word-level QE are provided in Table 6. In this table, we provide the MLT scores for these two ensemble systems. For the second system, which combines an even number of algorithms, we consider the both possible output types (OK or BAD) in case of ties.

| | MLT |
|---|---|
| MBL+PE+RF | 0.35 |
| MBL+PE+RF+LR (Ties OK) | 0.35 |
| **MBL+PE+RF+LR (Ties BAD)** | **0.37** |

Table 6: The MLT scores for the two ensemble systems. The plus sign '+' indicates the combined algorithms.

Based on the results we obtained from these experiments, we selected the following systems for the submission of this year's shared task on word-level QE:

- *RF*: The RF system, which uses the '*b+s+t*' feature set (best scoring system)
- *ENS*: The ensemble system indicated as: MBL+PE+RF+LR (Ties BAD)

These two systems obtained MLT scores on the test set of respectively 0.41 and 0.38 and were ranked third and fourth on the word-level QE task.

## 3 Sentence-level Quality Estimation

The aim of sentence-level QE is to predict Human mediated Translation Edit Rate (HTER) (Snover et al., 2006) scores that are obtained by comparing the MT output to its post-edited version. The ranking variant of this task is defined as ranking the MT output (per segment) from best to worst.

### 3.1 Features and Language Resources

In our experiments we initially used two feature sets: The baseline features (17) and the additional features (17) we used for our last year's submission. These additional features rely on the surface LM, PoS LM and the phrase table as well as the output of the best word-level QE system (RF) for each MT output. Detailed descriptions of these features can be found in Tezcan et al. (2015). Based on the observations we made during our experiments (see Section 3.3 for details) we designed two extra features that use additional information from the surface LM.

### 3.2 Learning Methods

We experimented with Support Vector Machines (SVMs), Linear Regression (LR) and Random Forests (RF) using the Scikit-learn module in Python to build regression models.

### 3.3 Experiments

In the first round of our experiments, we used two feature sets, namely the baseline features (*b*) and the additional features (*a*) that are described in Tezcan et al. (2015). We applied hyper-parameter optimization for the ML algorithms (when applicable) using 10-fold cross validation on the training set and tested the regression performance on the development set. The performance of the different ML algorithms and the different feature sets, with respect to Pearson's correlation (r), Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are provided in Table 7.

| | | **SVM** | RF | LR |
|---|---|---|---|---|
| *b* | r | 0.38 | 0.34 | 0.36 |
| | MAE | 13.87 | 14.66 | 14.29 |
| | RMSE | 19.52 | 19.43 | 19.29 |
| ***b+a*** | r | 0.42 | 0.39 | 0.42 |
| | MAE | 21.55 | 22.89 | 21.52 |
| | RMSE | 26.30 | 27.62 | 25.86 |

Table 7: The performance of different ML algorithms and feature sets on the development set. The plus sign '+' indicates the combined feature sets.

We analyzed the RF system to rank the features for their informativeness using the Scikit-learn module, which implements *gini importance* as described in Breiman et al. (1984). Gini importance, whose values are positive and sum to 1, provides information about the sum of impurity decrease for each variable, over all nodes in all decision trees. Based on this analysis, we list the top five features and corresponding importance scores in Table 8.

| | **Feature** | **Score** |
|---|---|---|
| 1 | % of 5-grams that appear in the LM at least once (*a*) | **0.73** |
| 2 | % of words that are marked as BAD by the best WL QE system (*a*) | 0.08 |
| 3 | LM probability of the source sentence (*b*) | 0.01 |
| 4 | Average source token length (*b*) | 0.01 |
| 5 | % of 4-grams that appear in the LM at least once (*a*) | 0.01 |

Table 8: The top five features for the RF system, with respect to gini importance scores, which uses the *b+a* feature set. Each feature is marked in brackets with the feature set that it comes from.

Considering the fact that the surface LM features were found to be extremely informative by the RF system (especially the % of 5-grams that appear in the LM at least once), we extended this feature set with n-grams of size 6 and 7 and named them as *lm6* and *lm7*[3]. We provide the performances of the different systems using the extended feature sets in Table 9.

| | | **SVM** | RF | LR |
|---|---|---|---|---|
| *b+a* *+lm6* | r | **0.42** | 0.39 | **0.42** |
| | MAE | 21.55 | 22.89 | 21.45 |
| | RMSE | 26.30 | 27.62 | 25.74 |
| ***b+a*** ***+lm6*** ***+lm7*** | r | **0.42** | 0.39 | **0.42** |
| | MAE | **21.46** | 22.93 | 21.48 |
| | RMSE | 26.21 | 27.66 | 25.80 |

Table 9: The performance of different ML algorithms and feature sets on the development set. The plus sign '+' indicates the combined feature sets.

The effectiveness of using the word-level QE predictions as features for sentence-level QE systems has been shown in previous years (de Souza et al., 2014; Tezcan et al., 2015). Moreover, a single feature based on the word-level predictions was able to perform better than the baseline features in previous year's shared task on QE (Tezcan et al., 2015). To confirm these results on a new language pair and domain, we performed a final experiment. In table 10, we can see the differences in the performances of the different systems using features sets that include and exclude the word-level feature (wl) (% of words that are marked as BAD by the best WL QE system).

| | | **SVM** | RF | LR |
|---|---|---|---|---|
| *wl* | r | 0.41 | 0.39 | 0.39 |
| | MAE | 18.24 | 19.33 | 19.79 |
| | RMSE | 23.07 | 23.97 | 25.37 |
| ***a-wl+b*** ***+lm6*** ***+lm7*** | r | 37.58 | 36.00 | 37.53 |
| | MAE | 21.47 | 22.75 | 27.17 |
| | RMSE | 26.03 | 27.17 | 25.40 |

Table 10: The performance of different ML algorithms and feature sets on the development set. While the plus sign '+' indicates inclusion, the minus sign '-' indicates the exclusion of a particular feature(s).

Based on the results we obtained from these experiments, for the scoring variant of the sentence-level QE task, we selected the following systems:

- *SVM1*: The SVM system, which uses the *a-wl+b+lm6+lm7* feature set
- *SVM2*: The SVM system, which uses the *a+b+lm6+lm7* feature set (best scoring system)

For the ranking variant of the sentence-level QE task, we used the output of these two systems to rank the sentences from best to worst. These two systems obtained *r* scores on the test set of respectively 0.36 and 0.41 and were ranked ninth and sixth on the sentence-level QE task.

## 4 Results and Discussion

For the word-level QE task, in addition to the baseline features, we extracted additional features based on accuracy and fluency of translations and features that utilize an automatically extracted bilingual terminology list. The results showed that

[3]No extension has been made to the features obtained from the PoS LMs.

all additional features were found to be informative by all the six ML algorithms we experimented with. Additionally, the best scores for five of these systems were obtained by including the features that are based on the bilingual terminology list. For the shared task, we worked with a small automatically extracted term list, but we assume that either a manually verified term list or a (larger) client-specific term list will further improve QE system performance, especially for the technical domain. Random forest, an ensemble of decision trees, was the best performing algorithm on the word-level QE, which utilized all the extracted features.

For sentence-level QE, we used different ML algorithms to train systems using the feature sets from our last year's submission. We extended this feature set based on a feature importance analysis we performed on the random forest system and added two new features (% of 6- and 7-grams that appear in the LM at least once). Including these features however showed only minor improvements on regression performance. This observation can be attributed to the high correlation between the features that all use the *n-gram* information on the target language, for different values of *n*.

Another interesting observation can be made for all three ML algorithms with respect to the baseline (*b*) and the merged feature sets (*b+a*). While the additional features improved the Pearson's correlation in all systems, they reduced the performance in terms of MAE and RMSE. To analyze this difference further, we plotted the errors made by the SVM system, using the two different feature sets, as shown in Figure 1.

The linear trend lines, provided in Figure 1, show that the slope of the equation *SVM(b+a) TL* (-0.67) is a better fit to the gold standard HTER scores ($y = 0$) than the slope of the equation *SVM(b) TL* (-0.82), which can explain the better correlation obtained with the *b+a* feature set, compared to *b*. On the other hand, the intercept of the equation *SVM(b+a) TL* (34) is further from the origin than the intercept of the equation *SVM(b) TL* (17), which can explain the lower MAE and RMSE scores obtained by the feature set *b*. A further analysis of the descriptive statistics for the HTER predictions coming from both systems and the gold standard HTER scores can be seen in Table 11.

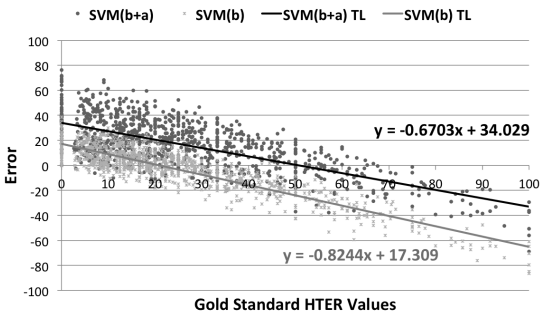


Figure 1: Errors made by the two SVM systems using the two different feature sets, sorted along the x-axis by their gold standard HTER scores. The equations for the linear trend lines (TL) for each data set are additionally provided.

| | Mean | Std. Dev. | Max. |
|---|---|---|---|
| *SVM(b)* | 21.82 | 9.28 | 48.22 |
| *SVM(b+a)* | 42.49 | 16.18 | 77.79 |
| Gold Std. | 25.69 | 20.37 | 100 |

Table 11: The mean, standard deviation and maximum values for each data set consisting of predicted and gold standard HTER scores.

Combining the information presented in Figure 1 and Table 11, we can see that the *SVM(b)* system has a smaller error margin on the lower end of the scale with respect to the HTER scores. This greatly influences the MAE and RMSE scores, given the fact that the gold standard HTER scores are skewed towards the lower end of the scale, centered around a mean of 25.69. In fact, the trend line *SVM(b) TL* corresponds to a smaller error margin between the gold standard HTER scores of 0 to 34.34 than the trend line *SVM(a+b) TL*[4]. The error margin for the former equation becomes greater than the latter starting from the HTER score of 34.34 (up to 100). The higher error margin on the high end of the scale can also be explained by the max. HTER predictions of the *SVM(b)* system (48.22). The additional features that are used in the *SVM(a+b)* system enable it to predict higher HTER values (max. 77.79), which seems to contribute to the higher correlation scores. Finally, we confirmed our observations from last year by showing that a sentence-level QE system, which uses a single feature based

[4]Based on Figure 1, solving the following equation for x gives us the gold standard HTER score, to which both equations are equidistant: $0 = -0.67x + 34 - 0.82x + 17$

on the word-level predictions of the best system, was able to beat the system trained on the baseline feature set. The performances of the sentence-level QE systems were further improved by combining this single feature with the baseline and the additional feature sets.

## Acknowledgments

This research has been carried out in the framework of the SCATE[5] project funded by the Flemish government agency IWT.

## References

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. In *Proceedings of the 20th international conference on Computational Linguistics*, page 315. Association for Computational Linguistics.

Ondrej Bojar, Rajen Chatterjeeand Christian Federmannand Barry Haddowand Matthias Huckand Chris Hokampand Philipp Koehnand Varvara Logachevaand Christof Monzand Matteo Negriand Matt Postand Carolina Scartonand Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 workshop on statistical machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46.

Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. 1984. *Classification and regression trees*. CRC press.

Stanley F Chen and Joshua Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 310–318. Association for Computational Linguistics.

Walter Daelemans, Jakub Zavrel, Kurt van der Sloot, and Antal Van den Bosch. 2004. Timbl: Tilburg memory-based learner. *Tilburg University*.

José GC de Souza, U Politecnica de Valencia, Christian Buck, Marco Turchi, and Matteo Negri. 2014. Fbk-upv-uedin participation in the wmt14 quality estimation shared-task. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 322–328.

Marcello Federico, Nicola Bertoldi, and Mauro Cettolo. 2008. Irstlm: an open source toolkit for handling large scale language models. In *Interspeech*, pages 1618–1621.

[5]http://www.ccl.kuleuven.be/scate

Simona Gandrabur and George Foster. 2003. Confidence estimation for translation prediction. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 95–102. Association for Computational Linguistics.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Taku Kudo. 2005. Crf++: Yet another crf toolkit. *Software available at http://crfpp. sourceforge. net*.

Arle Lommel, Aljoscha Burchardt, Maja Popovic, Kim Harris, Eleftherios Avramidis, and Hans Uszkoreit. 2014. Using a new analytic measure for the annotation and analysis of mt errors on real data. *EAMT-2014*, pages 165–172.

Lieve Macken, Els Lefever, and Veronique Hoste. 2013. Texsis: bilingual terminology extraction from parallel corpora using chunk-based alignment. *Terminology*, 19(1):1–30.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830.

Marcis Pinnis. 2015. Dynamic terminology integration methods in statistical machine translation. In *Proceedings of the Eighteenth Annual Conference of the European Association for Machine Translation (EAMT 2015)*.

Fabio Rinaldi, Michael Hess, James Dowdall, Diego Mollá Aliod, and Rolf Schwitter. 2004. Question answering in terminology-rich technical domains. In *New directions in question answering*, pages 71–86.

Helmut Schmid. 1995. Improvements in part-of-speech tagging with an application to german. In *In Proceedings of the ACL SIGDAT-Workshop*. Citeseer.

Alina Secară. 2005. Translation evaluation–a state of the art survey. In *Proceedings of the eCoLoRe/MeLLANGE Workshop, Leeds*, pages 39–44. Citeseer.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, pages 223–231.

Arda Tezcan, Véronique Hoste, Bart Desmet, and Lieve Macken. 2015. Ugent-lt3 scate system for machine translation quality estimation. In *Tenth Workshop on Statistical Machine Translation*.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *LREC*, pages 2214–2218.

John S White. 1995. Approaches to black box mt evaluation. In *Proceedings of Machine Translation Summit V*, page 10.