Natural Language Engineering (2020), **26**, pp. 73–94
doi:10.1017/S1351324919000111

CAMBRIDGE UNIVERSITY PRESS

ARTICLE

Estimating word-level quality of statistical machine translation output using monolingual information alone

Arda Tezcan*, Véronique Hoste and Lieve Macken

LT[3], Language and Translation Technology Team, Department of Translation, Interpreting and Communication, Ghent University, Ghent, Belgium
*Corresponding author. Email: arda.tezcan@ugent.be

(Received 7 September 2017; revised 4 February 2019; accepted 6 February 2019; first published online 27 March 2019)

Abstract

Various studies show that statistical machine translation (SMT) systems suffer from fluency errors, especially in the form of grammatical errors and errors related to idiomatic word choices. In this study, we investigate the effectiveness of using monolingual information contained in the machine-translated text to estimate word-level quality of SMT output. We propose a recurrent neural network architecture which uses morpho-syntactic features and word embeddings as word representations within surface and syntactic *n*-grams. We test the proposed method on two language pairs and for two tasks, namely detecting fluency errors and predicting overall post-editing effort. Our results show that this method is effective for capturing all types of fluency errors at once. Moreover, on the task of predicting post-editing effort, while solely relying on monolingual information, it achieves on-par results with the state-of-the-art quality estimation systems which use both bilingual and monolingual information.

Keywords: Machine translation; Quality estimation; Neural networks

1. Introduction

Human translation and machine translation (MT) quality are often linked to two main concepts, namely adequacy (also known as accuracy) and fluency, which represent the transfer and re-encoding tasks involved in the translation process, respectively (White 1995; Toury 2000; Lommel, Uszkoreit, and Burchardt 2014). Whereas adequacy is concerned with how much of the source content and meaning is preserved in the target text, fluency is concerned with the well-formedness of the produced translation, regardless of the transferred meaning from the source sentence.

The field of MT has seen promising improvements in recent years with the introduction of neural MT (NMT) as a new paradigm. Even though NMT systems have pushed the state of the art by outperforming statistical MT (SMT) systems on different language pairs and with respect to different evaluation methods (Bahdanau, Cho, and Bengio 2014; Bojar *et al.* 2016; Bentivogli *et al.* 2016), especially customized SMT systems are still widely used, given that they deliver better results in certain scenarios (Castilho *et al.* 2017).

In recent studies, scholars also gained a better understanding of the error profiles of the different MT paradigms. While the quality of any translated text can only be accurately assessed by both its accuracy and fluency, fluency seems to play an important role in defining the quality of SMT output (Bentivogli *et al.* 2016; Tezcan, Hoste, and Macken 2017*b*). In this study, we investigate to what extent the monolingual properties of the SMT output can be used to estimate its quality. With the goal of locating MT errors on subsegment level using monolingual information

© Cambridge University Press 2019

https://doi.org/10.1017/S1351324919000111 Published online by Cambridge University Press

alone, we focus on two word-level quality estimation (QE) tasks, namely detecting fluency errors and predicting overall post-editing effort. While the first task consists in detecting fluency errors, which are obtained in the form of manual error annotations, the second task consists in detecting words in the MT output that require post-editing due to translation errors, in terms of both accuracy and fluency. On the one hand, successful detection of fluency errors can be used to assist post-editors and MT developers to highlight monolingual errors in machine-translated text and to develop better automatic post-editing (APE) systems. Besides informing users about the types and location of errors, error detection systems can serve as a backbone for sentence-level QE systems. The output obtained from different error detection systems, which focus on accuracy and fluency errors, can be combined and used as an input for sentence-level QE systems. On the other hand, accurate prediction of overall post-editing effort, by using the monolingual information present in the SMT output alone, can reduce the effort required for building QE systems by eliminating the need for source language analysis. This can especially be beneficial in scenarios when translations are done from multiple source languages to a single target language or when limited resources and/or natural language processing (NLP) tools are available to process the source language.

For estimating word-level quality of SMT output, we propose a recurrent neural network (RNN) architecture which takes morpho-syntactic and distributed word representations as separate inputs and uses them within surface and syntactic context windows. Firstly, we test the proposed method on the detection of fluency errors for the English–Dutch (EN–NL) language pair. Secondly, we utilize this method to estimate the overall post-editing effort for both the EN–NL and English–German (EN–DE) language pairs. Furthermore, we investigate whether combining morpho-syntactic features with word-embedding models learned from lemmas and by using in-domain data improves QE performance, compared to using models that are learned from surface forms of words and by using general-domain data.

This work contributes beyond the existing work by demonstrating the informativeness of the monolingual information obtained from a given MT output and by combining morpho-syntactic information with word-embeddings in an RNN architecture. Our results indicate that the monolingual information contained in the SMT output alone has high predictive power for building competitive word-level QE systems for both types of QE tasks and both language pairs we are studying.

The rest of this paper is organized as follows: In Section 2, we describe related research. Sections 3 and 4 explain the morpho-syntactic and distributed word representations and the proposed RNN architecture, respectively. In Section 5, we evaluate the proposed approach on two different QE tasks. We conduct an error analysis in Section 6 and finish with concluding remarks in Section 7.

2. Related work

For any individual NLP system, such as an MT system, evaluation can be approached from various points of view with varied implications for the conduct of any evaluation and the lessons to be drawn from it (Jones and Galliers 1995). While the quality of an MT system can simply be characterized by the errors it makes, MT quality can have different interpretations depending on the purpose of the assessment. In literature, we can see various studies, in which the quality of MT output is assessed for its adequacy and fluency (White 1995; Toury 2000; Graham *et al.* 2014; Bojar *et al.* 2016, 2017). Drawing a distinction between the different properties of machine-translated text has proved fundamental to error analysis. In a bid to study the strengths and weaknesses of MT systems in detail, fine-grained error analysis has generated interest and different taxonomies have been proposed to identify MT errors (Vilar *et al.* 2006; Daems, Macken, and Vandepitte 2014; Lommel *et al.* 2014; Costa *et al.* 2015; Tezcan *et al.* 2017*b*). With the recent improvements in the field of MT, post-editing, which can simply be defined as correcting errors in the MT output, has

https://doi.org/10.1017/S1351324919000111 Published online by Cambridge University Press

become a common practice in translation workflows. Given that MT quality is not uniform and certain machine-translated segments may contain more errors than others, the quality of MT systems has been increasingly measured in terms of the overall post-editing effort that is required to correct machine-translated text (Blatz *et al.* 2004; Specia *et al.* 2009; Koponen *et al.* 2012; Bojar *et al.* 2016, 2017).

QE is the task of providing a quality indicator for machine-translated text without relying on reference translations (Gandrabur and Foster 2003). Like the different interpretations of MT quality, QE approaches also vary. Early work on QE for MT focused on estimating the quality at the word level with the aim of suggesting translations of the source text within interactive MT scenarios (Ueffing and Ney 2005). Word-level QE has regained attention in the recent years thanks to its ability to identify words that require post-editing (Bojar *et al.* 2015, 2016). Sentence-level QE, on the other hand, aims to estimate an overall quality score, which has often been measured in terms of human quality judgments or post-editing effort (Specia *et al.* 2009, 2013; Bojar *et al.* 2015, 2016).

With the aim of capturing overall quality of SMT output, state-of-the-art QE systems often analyze both the bilingual and monolingual properties of a source text and MT output (Bojar *et al.* 2016, 2017). This is a natural choice as monolingual information obtained on the target text alone is not enough to capture all types of translation errors, such as errors that correspond to incorrect meaning transfer from source to target text. On the other hand, with the assumption that the quality of a human translation is much less affected by features of the source text than factors such as time pressure and familiarity with the domain, Hokamp *et al.* (2014) demonstrated that target-centric features are highly informative on the task of sentence-level QE of human translation. Moreover, in a recent study, Avramidis (2017) found out that grammatical and fluency-related features made a solid contribution to the performance of sentence-level QE systems, which were trained for English–German and German–English language pairs using various MT architectures, excluding NMT.

In recent years, with the advances in NMT, a number of studies have revealed the error profiles of different MT paradigms. Multiple studies showed that, compared to NMT, SMT systems make more fluency errors in the form of grammar and lexical choice (Bentivogli *et al.* 2016; Klubička, Toral, and Sánchez-Cartagena 2017). Moreover, in different studies it has been observed that fluency errors make up the majority of translation errors made by SMT systems (Tezcan *et al.* 2017*b*; Daems *et al.* 2017). These studies suggest that while the bilingual aspects of MT text cannot be neglected, the monolingual properties of SMT output play an important role in assessing its overall quality.

In the literature, we see a variety of studies that focus on monolingual information to detect grammatical errors in SMT output (Stymne and Ahrenberg 2010; Ma and McKeown 2012; Tezcan, Hoste, and Macken 2017*a*). In one of these studies, Tezcan *et al.* (2017*a*) showed that the morpho-syntactic features used for detecting grammatical errors can also learn meaningful patterns to estimate overall post-editing effort at the word level. However, morpho-syntactic features lack the information about the semantic properties of words, and additional information is needed to detect all fluency errors, including semantic errors, and to predict overall post-editing effort.

One way of capturing both syntactic and semantic properties in MT output is to use word embeddings, which are real-valued vectors whose dimensions represent a latent feature of a given word (Li *et al.* 2016). In the last decade, word embeddings have been successfully applied to variety of NLP tasks, including named entity recognition (Turian, Ratinov, and Bengio 2010), parsing (Socher *et al.* 2011*a*), sentiment analysis (Socher *et al.* 2011*b*) and MT (Cho *et al.* 2014). In recent years, word embeddings have also been used to estimate overall post-editing effort within neural network (NN) architectures (Kreutzer, Schamoni, and Riezler 2015; Abdelsalam, Bojar and El-Beltagy 2016; Kim and Lee 2016; Patel and Sasikumar 2016; Martins *et al.* 2016; Scarton *et al.* 2016; Kim, Lee, and Na 2017; Hokamp 2017; Martins, Kepler, and Monteiro 2017).

https://doi.org/10.1017/S1351324919000111 Published online by Cambridge University Press

For word-level QE, Kreutzer *et al.* (2015) proposed a system combination approach which combined a multilayer perceptron trained with the word-embedding representations of the context of the target word and the context around its source alignment, with a linear classifier trained with the baseline features of Workshop on Statistical Machine Translation 2015 (WMT'15) QE task (Bojar *et al.* 2015), which are based on information such as language model scores, the context words of a given target word and its source alignment, and part-of-speech (PoS) labels. Similarly, Patel and Sasikumar (2016) used the context information obtained on the target and source words in an RNN-based system for the same task. In a different study, Kim and Lee (2016) proposed a two-stage RNN-based word-level QE model consisting of a neural word prediction model (i.e. word predictor), which uses both source and target text information from large-scale parallel corpora and a neural QE model (i.e. quality estimator) to predict the quality label for each target word. Kim *et al.* (2017) advanced this model by applying a stack propagation method by allowing backpropagation from the quality estimator to the word predictor and achieved improvements in QE performance. Martins, Kepler, and Monteiro (2017) proposed a word-level QE system which consists of a neural model stacked into a linear feature-rich classifier. In this system, they use a number of features that depend on the context of a given target word and its aligned source word, as well as syntactic features that provide information about the dependency structure of a given sentence. Moreover, Martins *et al.* (2017) proposed using an APE system for predicting word-level quality labels, which relies on a large amount of artificially generated MT–PE sentence pairs, also referred to as "roundtrip translations." The idea of using such artificial data was a major contribution toward improving APE systems (Junczys-Dowmunt and Grundkiewicz 2016). Hokamp (2017) achieved the state-of-the-art QE performance by using a similar APE approach proposed by Martins *et al.* (2017), by building ensembles of specialized NMT systems for the APE and word-level QE tasks while additionally incorporating syntactic features as "factors" in the input to NMT systems. Similar to the work of Martins *et al.* (2017), this system relies on roundtrip translations for generating additional training data.

The proposed NN-based word-level QE system in this work differs from the above-mentioned studies in three ways. The most important difference is that, unlike many of the related work on QE, we propose a QE system that uses only monolingual information extracted from the MT output. We elaborate on the work of Tezcan *et al.* (2017*a*), which introduced novel morpho-syntactic word representations, which we now combine with word embeddings within surface and syntactic context windows in a single RNN-based QE system and apply this model on two different tasks, namely detecting fluency errors and predicting technical post-editing effort on the word level. The third novelty in this study is that we investigate the impact of combining morpho-syntactic word representations with word embedding models learned from lemmas and the impact of learning word embedding models from in-domain data on word-level QE estimation performance.

3. Word representations: Morphology, syntax, and semantics

As a form of written text, MT output can be studied by the three main subdomains of linguistics, viz. morphology, syntax, and semantics. While morphology studies the rules that govern the way morphemes are used in a language, syntax studies the rules that pertain to the ways in which words can be combined to form grammatical sentences in a language. Finally, semantics studies the meaning of words and combinations of words in a language. Even though features that represent the properties of source text can additionally be beneficial to detect fluency errors, our expectation is that each of these three domains will provide the necessary information obtained on target text alone to identify the fluency errors made by MT systems to a great extent. Based on this idea, we consider two types of word representations relevant for monolingual QE in SMT: morpho-syntactic word representations (Tezcan *et al.* 2017*a*), which include information from both morphology and syntax, and word embeddings, which capture semantic and syntactic

https://doi.org/10.1017/S1351324919000111 Published online by Cambridge University Press

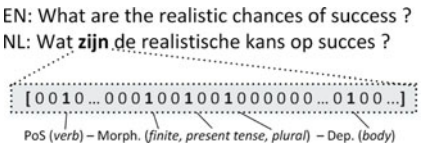


Figure 1. Binary vector for "*zijn (are)*" consisting of 1s for its PoS, morphology, and dependency features and 0s for the remaining items in the vocabulary.

similarities between words (Mikolov *et al.* 2013). Tezcan *et al.* (2017*a*) defined morpho-syntactic word representations as binary, multi-hot vectors consisting of PoS, morphology, and dependency information about a given word. In this method, each word w is mapped to a binary vector y_w with one dimension for each possible PoS, morphology, and dependency label, where $y_w^i = 1$ and $y_w^j = 0, \forall j \neq i$, with i representing the morpho-syntactic features of the given word w. Figure 1 shows an example source sentence (EN), its machine-translated version (NL), and the morpho-syntactic representation for the word "zijn (are)".

One difficulty of using dependency parsing on MT output is that the generated parse trees can be unreliable when the MT output itself contains fluency errors. On the other hand, multiple studies demonstrated that parse trees obtained on MT output nevertheless provide useful information in terms of MT quality. Tezcan, Hoste, and Macken (2016) showed that parsers were useful to detect grammatical errors in the MT output due to the unusual dependency structures they produce. The same year, Martins *et al.* (2016) included syntactic features extracted from the parse trees obtained on the SMT output in their system and achieved the best results on the WMT'16 shared task on word-level QE. Dependency parsing on SMT output has also been applied to the task of quality evaluation in the past. Owczarzak, van Genabith, and Way (2007) presented an MT evaluation metric which achieved highest correlation with human evaluation of translation fluency by using labeled dependencies on the MT output.

Unlike morpho-syntactic features, word embedding models capture both the syntactic and the semantic relationships between words (Turian *et al.* 2010). As word embeddings also model semantic information, our hypothesis is that morpho-syntactic word representations and word embeddings can complement each other and achieve better estimation performance than when they are used in isolation, on the task of word-level QE of SMT output.

In a recent study, Avraham and Goldberg (2017) argue that lemmas are responsible for semantic aspects of the similarities between words and have the ability to improve semantic similarity models when used in isolation, while affixes are mostly responsible for morphological similarity. When word embeddings are used in combination with accurate morpho-syntactic features, the question also arises whether the additional morphological information stored in the affixes provides any valuable information. We investigate whether using lemmas for learning word embedding models is a better alternative to using surface forms, when they are combined with accurate morpho-syntactic features.

The positive effect of learning translation and language models from in-domain data on the quality of SMT systems has been shown in various studies in the past (Xu *et al.* 2007; Bertoldi and Federico 2009; Axelrod, He, and Gao 2011). Even though the ability to train on very large data sets allows word embedding models to learn complex word relationships (Kusner *et al.* 2015), Anastasakos, Kim, and Deoras (2014) showed that word embeddings learned from domain-specific data performed better over those obtained from general-domain data on the task of spoken language understanding. Furthermore, this positive effect was observed even when the word embedding models were learned from much smaller domain-specific data compared to general-domain data (Anastasakos *et al.* 2014). In a different study, Blain, Scarton, and Specia (2017) used word embeddings that are obtained from in-domain data (for the IT and medical domains) on word-, phrase-, and sentence-level QE tasks. Based on the reasonably good results they achieved on sentence-level QE compared to similar systems in literature, which used out-of-domain word embeddings, the authors argue that in-domain word embeddings encoded useful information for QE tasks. To our knowledge, the impact of the word-embedding domain on

https://doi.org/10.1017/S1351324919000111 Published online by Cambridge University Press

the QE-task has never been studied before. In this study, we also analyze the impact on QE performance of learning word embedding models from different domains with different data sizes.

4. NN architecture

We propose an NN architecture which uses both morpho-syntactic word representations and word embeddings as input vectors. Four fixed-sized context windows are created using each type of word representation. The four context windows consist of the following surface and syntactic n-grams, as described by Tezcan *et al.* (2017*a*)[a]:

- **Surface *n*-gram** (n): Sequence of words as they appear in MT output, centered around the target word ($n = 5$)
- **Syntactic *n*-grams** (sn):
  - **Parents** (sn_p): Vertical sequence of parent nodes in a given dependency tree for a given target node ($n = 3$)
  - **Siblings** (sn_s): Horizontal sequence of sibling nodes sharing the same parent in a given dependency tree, centered around the target node ($n = 5$)
  - **Children** (sn_c): Sequence of children nodes for a given target node (depth 1), containing the target node in the center ($n = 5$)

We include additional placeholder tokens in the vocabulary of the morpho-syntactic features to indicate boundaries (namely "<s>" to indicate a sentence boundary, "[ROOT]" to indicate the root of the dependency tree, and "[NA]" to indicate horizontal boundaries in the sub-trees). Moreover, as described in Tezcan *et al.* (2017*a*), we retain the original word order in each syntactic n-gram, with a view to capturing word-ordering errors in machine-translated texts.

The context windows built for the two types of word representations are provided as input to eight Gated Recurrent Unit (GRU) (Cho *et al.* 2014) layers, respectively. GRU is a variant of RNNs and aims to capture dependencies of different time scales. Despite their similar performances on different tasks, unlike long short-term memory (LSTM) units, GRUs feature gating units without the separate memory cell. GRUs have outperformed LSTM units on different NLP tasks, including QE, both in terms of convergence in CPU time and parameter updates and generalization (Chung *et al.* 2014; Patel and Sasikumar 2016). Assuming that each time step t has an input x_t and a hidden state h_t, GRU is defined as

$$r_t = sigm(W_{xr}x_t + W_{hr}h_{t-1} + b_r) \quad (reset\ gate), \tag{1}$$

$$z_t = sigm(W_{xz}x_t + W_{hz}h_{t-1} + b_z) \quad (update\ gate), \tag{2}$$

$$\widetilde{h_t} = tanh\big(W_{xh}x_t + W_{hh}(r_t \odot h_{t-1}) + b_h\big) \quad (memory\ content), \tag{3}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \widetilde{h_t} \quad (activation), \tag{4}$$

where $\odot$ is an element-wise product, W_* the weight matrices, b_* the bias vectors, r the reset gates, z the update gates, and *sigm* and *tanh* the sigmoid function and the hyperbolic tangent function, respectively.

The outputs of the eight GRU layers are concatenated before they are connected to the output layer, which consists of two units. The softmax over the activation of these two units is taken as the

[a]By analyzing the dependency trees obtained on the sentences in the SCATE corpus of MT errors, Tezcan *et al.* (2017*a*) first selected a range of n values as candidates for the optimal size of syntactic and surface n-grams on the task of detecting grammatical errors. From this set of candidates, specific n values were found to maximize QE performance. To build surface and syntactic and surface n-grams in this study, we used the best n values observed in Tezcan *et al.* (2017*a*).

https://doi.org/10.1017/S1351324919000111 Published online by Cambridge University Press

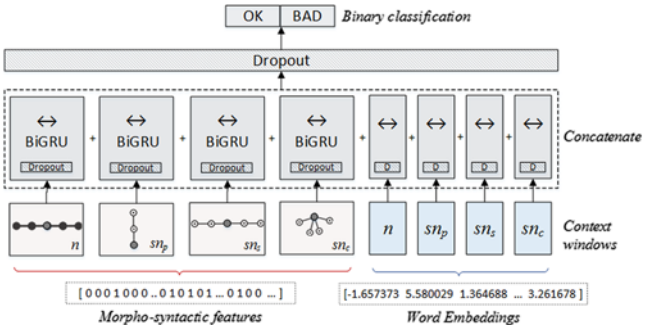


Figure 2. The proposed NN architecture.

score for the two classes OK and BAD which represent the correct and erroneous words, respectively. To reduce overfitting, we apply dropout (Srivastava et al. 2014) within the GRU layers (for the input gates and the recurrent connections) and after the concatenated hidden units. Figure 2 illustrates the proposed NN architecture.

In all of our experiments, we use Gaussian initialization scaled by fan_in + fan_out (Glorot and Bengio 2010) to initialize the weights, binary cross-entropy as loss function, and RMSProp (Tieleman and Hinton 2012) as optimizer. We fix the mini-batch size to 50 and train each model for 50 epochs. We implement all models using the TensorFlow framework (Abadi *et al.* 2016). We adjust the sizes of the GRU layers according to the sizes of the two different data sets we use in our experiments, which is further explained in Section 5.

5. Experiments

We test the proposed method on two different word-level QE tasks. In the first experiment, the task is to detect fluency errors in the SMT output for the EN–NL language pair. As a binary classification task, we consider each word in a given SMT output either erroneous or correct and we extract the error labels by utilizing a corpus that contains manually annotated errors. Furthermore, we investigate whether the information provided by the morpho-syntactic features and word embeddings complement each other and lead to better estimation performance. Our hypothesis is that, unlike morpho-syntactic features, word embeddings provide additional useful information about the semantic similarities between words and complement the information provided by morpho-syntactic features on the task of QE. In a recent study, Avraham and Goldberg (2017) showed that lemmas were able to improve semantic similarity models compared to using surface forms of words. Encouraged by their findings, we test whether QE performance can be maximized when lemmas are used to train word embedding models instead of surface forms of words. Moreover, based on the positive results obtained in other tasks (as outlined in Section 2), we study the impact of using smaller but in-domain data to train word embedding models compared to using large, general-domain data.

In the second experiment, we test the proposed method on the task of predicting overall post-editing effort, by detecting the words that require post-editing. In this binary classification task, the error labels are automatically extracted for each word by comparing the SMT output to its post-edited version. As post-editing is the task of correcting all types of translation errors, including accuracy and fluency errors, the extracted error labels in this experiment do not draw a distinction between error types. They only indicate that a post-editing operation has been performed on a given word. This experiment seeks to analyze whether monolingual information alone is effective to estimate the overall post-editing effort for a given SMT output. Similar to the task of detecting fluency errors, we analyze the impact of different NN configurations and word embedding models on estimation performance. To test the applicability of the proposed method across languages, we conduct our experiments on two language pairs (EN–NL and EN–DE).

https://doi.org/10.1017/S1351324919000111 Published online by Cambridge University Press

Table 1. Number of words annotated as fluency errors in the SCATE corpus of SMT errors

Error type	No. of erroneous words
Grammar	11,073
Multiple errors	2055
Lexicon	1224
Orthography	2055

5.1 Detecting fluency errors

In the first experiment, we evaluate the performance of the proposed method on the detection of fluency errors for the EN–NL language pair. For this experiment, we use the SCATE corpus of SMT errors (Tezcan *et al.* 2017*b*), which consists of 2967 sentence pairs with manually annotated, fine-grained error annotations. The source sentences in this data set are extracted from three different text types of the Dutch Parallel Corpus (Macken, De Clercq, and Paulussen 2011), namely external communication, nonfictional literature, and journalistic texts. The translations in this data set are obtained from Google Translate,[b] which was based on SMT technology. The SCATE MT error taxonomy, proposed by Tezcan *et al.* (2017*b*), is hierarchical in nature and is based upon the familiar notions of accuracy and fluency. In this taxonomy, any error that can be detected in the target text alone is referred to as a fluency error. Fluency errors make up 68% of all error annotations in this corpus. The SCATE error taxonomy is further divided into four main error categories: *grammar*, *lexicon*, *orthography,* and *multiple errors*. "Grammar" errors include all errors against the grammatical rules of the target language (in our case Dutch) with respect to word order, word form, missing or extra words, and multi-word syntax. "Lexicon" errors are concerned with words that either do not exist in the target lexicon or words that belong to the target lexicon but do not fit in a given context. "Orthography" errors include spelling, capitalization, and punctuation errors. Multiple errors refer to a combination of fluency errors which make these difficult to annotate separately. In the SCATE error taxonomy, any error that can only be detected by analyzing both the source and target text is referred to as accuracy error. For the details on the SCATE MT error taxonomy, we refer the readers to Tezcan *et al.* (2017*b*).

Despite the availability of a similar error taxonomy, namely Multidimensional Quality Metrics (MQM) (Lommel *et al.* 2014) and a data set annotated with MT errors using the MQM taxonomy, namely the WMT'14 data set (Bojar *et al.* 2014), in this study, we opted for the SCATE corpus of MT errors due to the differences in the annotation process. MQM uses a decision tree to guide the error annotation process and prioritizes annotating accuracy errors over fluency errors. This approach results in data sets in which fluency errors can remain invisible due to overlaps with accuracy errors. Unlike MQM, the SCATE error annotation guidelines allow for the annotation of multiple error types on the same text span independently, which allows us to analyze all fluency error annotations in isolation.

As the aim of the proposed NN architecture is to capture all types of fluency errors at once, for our experiments, we use the subset of fluency error annotations from the SCATE corpus of SMT errors. We mark all words that are annotated by any type of fluency error as BAD and the words that are not annotated with fluency errors as OK. The resulting data set consists of 58,002 words, 16,275 of which are annotated as fluency errors, with an OK to BAD ratio of approx. 2.56:1. We provide the number of words annotated with different subtypes of fluency errors in Table 1.[c]

[b] `http://translate.google.com` (June, 2014)

[c] As the SCATE error taxonomy allows multiple error annotations in the same text span, each word in the MT output can be subject to annotations of multiple error types.

https://doi.org/10.1017/S1351324919000111 Published online by Cambridge University Press

As Logacheva, Lukasik, and Specia (2016*b*) state, one of the reasons word-level QE is a challenging problem is the fact that OK and BAD labels are not equally important: we are generally more interested in finding incorrect words than in assigning a suitable category to every single word. According to Logacheva *et al.* (2016*b*), for evaluating word-level QE tasks, an ideal metric should be oriented toward the recall for the BAD class. However, they add that the case of F1_BAD score, which is calculated as the harmonic average of precision and recall for the BAD class, shows that this is not the only requirement: in order to be useful, the metric should not favor pessimistic labelings, that is, all or most words labeled as BAD. Logacheva *et al.* (2016*b*) further argue that the weighted average of F1 scores for the two classes is not suitable as it will be dominated by F1_OK due to label imbalance, with the assumption that any reasonable MT system nowadays will generate texts where most words are correct. Therefore, they suggest F1_MULTI, the multiplication of the F1 scores for the two labels, as an appropriate evaluation metric. In this experiment, we follow the work of Logacheva *et al.* (2016*b*) and use F1_MULTI as the primary evaluation metric for evaluating word-level QE systems. F1_MULTI has also been used as the primary evaluation criterion in the WMT shared tasks on QE, in recent years (Bojar *et al.* 2016, 2017). During our experiments, to avoid any biasing toward choice of initial weights, we train each NN three times with random weight initialization and calculate the average scores for each evaluation criterion.

Considering the relatively small number of samples in this data set, we evaluate the systems using the average 10-fold cross-validation scores. During training, we assign class weights that are inversely proportional to their frequency in each training fold using the following formula:

$$W_{ok} = 1 - (C_{ok}/C_{total}), \tag{5}$$

$$W_{bad} = 1 - (C_{bad}/C_{total}), \tag{6}$$

where W_{ok} and W_{bad} refer to the weights and C_{ok} and C_{bad} refer to the counts of the words labeled as OK and BAD, respectively. C_{total} refers to the total count of words in the data set. Setting class weights that are inversely proportional to their frequency aims to handle the imbalance we observe in the data sets we experiment with as it down-weighs the loss for the majority class and up-weighs the loss for the minority class.

We use the Alpino parser (Van Noord 2006) to obtain the PoS, morphology, and dependency information for each given word in the machine-translated output. The morpho-syntactic word vectors that are built with this information consist of 129 features. Similarly, we build word embeddings with 129 dimensions, using 328M words from the SoNaR corpus (Oostdijk *et al.* 2008), which consists of texts extracted from 38 text types, including newspapers and e-mails. To build word embeddings, we implement *word2Vec* (Mikolov *et al.* 2013) using the Python[d] library *gensim* (Řehuek and Sojka 2010). For the EN–NL experiments, we train the NN systems with GRU layer sizes of 25.

First, we analyze the impact on QE performance of morpho-syntactic features and word embeddings within different NN configurations. The performance of the different systems we build is included in Table 2.

As a first baseline, we build a conditional random field (CRF) system which uses the baseline features used in the WMT'16 shared task on QE (22) with one difference. Instead of using all baseline features, to be able to make a fair comparison with the proposed NN architecture that relies on monolingual information, we only used the features that refer to the target text (14). We used the CRF++ toolkit[e] to train QE models and to predict quality labels, and the Marmot

[d] https://www.python.org
[e] http://taku910.github.io/crfpp/

https://doi.org/10.1017/S1351324919000111 Published online by Cambridge University Press

Table 2. The impact of the machine learning algorithm and NN architecture

System	F1_BAD	F1_OK	F1_MULTI
CRF (WMT'16)	0.3254	0.8396	0.2734
UniGRU-MS (Tezcan *et al.* 2017)	0.4903	0.8203	0.4021
UniGRU-W2V	0.4686	0.7851	0.3679
UniGRU-MS/W2V	0.5038	0.8014	0.4037
UniGRU-MS+W2V	0.5050	0.8206	0.4144
BiGRU-MS+W2V (proposed architecture)	**0.5157**	0.8161	**0.4209**

tool (Logacheva, Hokamp, and Specia 2016*a*) to extract the features, which are further outlined in Bojar *et al.* (2016).[f]

As a second baseline, we build the system proposed by Tezcan *et al.* (2017*a*), which provides morpho-syntactic features as input to the four context windows, as described in Section 4, and which uses unidirectional GRU layers (UniGRU-MS). The estimation performance of these two systems shows us that the NN system, which is based on morpho-syntactic information alone, is able to outperform the CRF system, which is based on the features that are used to build the baseline system in WMT'16 shared task on QE, with relative improvements of 53% and 48% in F1_BAD and F1_MULTI scores, respectively.

We use the same NN architecture to build two additional systems, a first one which uses only word embeddings (UniGRU-W2V) and a second system using a concatenation of morpho-syntactic features and word embeddings as a single vector (UniGRU-MS/W2V). By their performance, we can observe that morpho-syntactic features outperform word embeddings on this task. The informativeness of morpho-syntactic features over word embeddings can be associated with the high number of grammatical errors present in the data set used in this study, which is shown in Table 1. We also see that when the two types of word vectors are concatenated as input vectors (UniGRU-MS/W2V), the results are slightly better (F1_MULTI = 0.4037).

An alternative approach for combining morpho-syntactic features and word embeddings is to provide two separate vectors as input to a total of eight context windows, as described in Section 4 (UniGRU-MS+W2V). We see that this alternative architecture is able to reflect the added benefit of combining the two types of inputs, as it improves the performance up to a F1_MULTI score of 0.4144. Splitting the morpho-syntactic information and distributed representation of words into separate vectors has a positive effect on estimation performance. Using bidirectional GRU layers (BiGRU-MS+W2V), as proposed in Section 4, further increases the performance and achieves the best results with respect to all evaluation criteria, with an F1_MULTI score of 0.4209.

Secondly, we analyze the impact of the domain of the corpus and the types of information, from which word embeddings are learned. In addition to using the (out-of-domain) SoNaR corpus, as described above, we additionally build word embeddings using 2.9M words from the (in-domain) Dutch Parallel Corpus[g] and on the merged data set which consists of all sentences from both corpora. Furthermore, we build alternative systems by using base forms of words (lemmas) instead of surface forms. In all cases, we build the proposed architecture as already described in Section 4, which provides separate word vectors as input, consisting of morpho-syntactic features and word embeddings. The performance of the different systems is provided in Table 3.

In the upper section of Table 3, we see that word embedding models learned from relatively small but in-domain data lead to better QE performance on the task of detecting fluency errors

[f]We used 2.9M words from the (in-domain) Dutch Parallel Corpus to build a language model, which is necessary to extract some of the WMT baseline features. This data set did not contain any sentences that are in the SCATE corpus of SMT errors.

[g]We remove the overlapping sentences with the SCATE corpus of SMT errors prior to building word embedding models.

https://doi.org/10.1017/S1351324919000111 Published online by Cambridge University Press

Table 3. The impact of word embedding model on estimation performance. Word embedding models learned from surface forms of words are indicated with "(S)" and models learned from lemmas with "(L)"

System	F1_BAD	F1_OK	F1_MULTI
BiGRU-MS+W2V_SONAR (S)	0.5185	0.8213	0.4259
BiGRU-MS+W2V_DPC (S)	0.5192	0.8238	0.4278
BiGRU-MS+W2V_DPC+SONAR (S)	0.5179	0.8202	0.4247
BiGRU-MS+W2V_SONAR.33M (S)	0.5184	0.8194	0.4247
BiGRU-MS+W2V_SONAR.33M (L)	**0.5195**	0.8194	0.4257
BiGRU-MS+W2V_DPC (L)	0.5187	**0.8297**	**0.4303**
BiGRU-MS+W2V_DPC+SONAR.33M (L)	0.5139	0.8221	0.4224

(BiGRU-MS+W2V_DPC (S)). Using very large, out-of-domain data in this case does not only achieve slightly inferior performance compared to in-domain data when used alone (BiGRU-MS+W2V_SONAR (S)), but also deteriorates the performance slightly when it is added to the in-domain data set (BiGRU-MS+W2V_DPC+SONAR (S)). Although further research would be necessary to make conclusions, the positive effect of the in-domain data might be explained by its ability to learn specific meanings of certain words in limited context. Even when both data sets are combined, learning additional meanings for words in larger, general-domain data might lead to a decrease in the error detection performance.

In the lower section, we see the performance of the systems using word embeddings models that are learned from lemmas instead of surface word forms, with the exception of the system displayed in the first row (BiGRU-MS+W2V_SONAR.33M (S)). Alpino provides lemmas for each word in the parse tree it builds. As parsing brings additional computational cost, we acquire lemmas only for approx. 33M words, which are obtained from 2M randomly selected sentences from the SoNaR corpus (BiGRU-MS+W2V_SONAR.33M (L)). We also combine the same data set with the in-domain data (BiGRU-MS+W2V_DPC+SONAR.33M (L)). To observe the impact of reducing the training data, we use the same 33M words to learn word embeddings from surface word forms (BiGRU-MS+W2V_SONAR.33M (S)). As a result, the lower section of Table 3 shows similar trends as in the upper section and confirms the positive effect of using in-domain data for building word embedding models when lemmas are used. Moreover, we see that combining morpho-syntactic features with word embeddings learned from lemmas achieves slightly better F1_MULTI scores than combining morpho-syntactic features with word embeddings learned from surface forms (BiGRU-MS+W2V_DPC (L) vs. BiGRU-MS+W2V_DPC (S)).

5.2 Predicting post-editing effort

In the second experiment, we evaluate the proposed method on the task of predicting overall post-editing effort, which consists of detecting the words that require post-editing. In this experiment, we do not have manually annotated errors. Instead, the error labels are extracted using the Translation Error Rate (TER) tool (Snover *et al.* 2006), by comparing the SMT output with its post-edited version. The motivation behind this experiment is to investigate the feasibility of using the same NN architecture to predict the technical post-editing effort involved in correcting a given MT output. While detecting all errors in a given MT output, without making a distinction between the error types, can be considered as an alternative to this experimental setup, previous studies show that different post-editors can arrive to different correct translations as a result of post-editing the same MT output and that MT errors are not always good indicators of post-edit

https://doi.org/10.1017/S1351324919000111 Published online by Cambridge University Press

operations (de Almeida 2013; Daems *et al.* 2014). Therefore, we consider predicting technical post-editing effort on the word level as a different task than detecting MT errors.

For extracting the error labels with TER, the following settings are adopted from the WMT'16 shared task on QE:

- Tokenized, case-insensitive, exact matching is performed.
- *Insertions* made during post-editing are not annotated as they cannot be associated with any word in the MT output.
- *Shifts* are disabled, but rather annotated as edits in the form of *deletions* and *insertions*.

As a result, all words that have been subject to post-editing are marked as BAD. All remaining words in the data set are marked as OK. Since post-editing involves correcting all types of translation errors, automatically extracted error labels do not distinguish between the error types but only indicate that a post-editing operation has been performed on a given word.

Tezcan *et al.* (2017*b*) showed that morpho-syntactic features, even though they require different parsing tools and use different PoS and morphology tagsets, can successfully be applied to the task of QE for both EN–NL and EN–DE language pairs. To analyze to what extent overall post-editing effort can be predicted using only monolingual information, similarly, we experiment with both language pairs. Word-level prediction of overall post-editing effort on SMT output has been a part of the WMT shared task on QE in recent years (Bojar *et al.* 2015, 2016). To be able to compare our results with previous studies, we use the WMT'16 data set (Specia, Logacheva, and Scarton 2016) for the EN–DE and the SCATE corpus of SMT errors for the EN–NL language pairs.

The WMT'16 data set consists of 15k source-target sentence pairs (279,976 words in the target language) in the IT-domain, automatically translated from EN into DE by a phrase-based SMT system. The data are partitioned into 12k, 1k, and 2k sentence pairs as training, tuning, and test sets, respectively. Each word in the given data sets is labeled as OK or BAD, with an OK to BAD ratio of approx. 3.70:1, 4.17:1, and 4.25:1 in the training, tuning, and test sets, respectively. We used two data sets to learn word embedding models in German. As in-domain data, given the small size of the WMT'16 training set for learning word embedding models, we combined it with the post-edited SMT segments from the Autodesk Post-Editing Data corpus.[h] This additional data set consists of 107,353 sentences and approx. 1.5M words in total.[i] As out-of-domain data, we used a collection of Wikipedia articles (Wolk and Marasek 2015), consisting of approx. 2.45M sentences and 43.5M words. To build the morpho-syntactic word vectors, we obtained PoS, morphology, and dependency labels for each German word in the MT output, using the Mate tools (Bohnet and Nivre 2012). The morpho-syntactic word vectors that are built for German consisted of 131 features. Similarly, we built word embeddings learned on German data with 131 dimensions. For the EN–DE experiments, we train the NN system with GRU layer sizes of 50 and increase the complexity of the NN, given the relatively bigger data set compared to EN–NL language pair, for which we set the GRU layer sizes to 25.

For the EN–NL language pair, we obtained the post-edited version of the SMT output by a Master student in translation studies. We applied the same label extraction method, as described above, for marking erroneous words.[j] The OK to BAD ratio in this data set is approx. 4.81:1. The word embedding models are learned from the same three data sets described in Section 5.1.

When analyzing the SCATE corpus of SMT errors (EN–NL), we see that the fluency errors make up the majority of errors in this data set (68%). As the SCATE error taxonomy allows

[h] https://autodesk.app.box.com/v/Autodesk-PostEditing

[i] We remove the overlapping sentences with the WMT'16 test set prior to building word embedding models.

[j] The post-edited version of the SMT output we use in this study is the same data used by Tezcan *et al.* (2017*a*).

https://doi.org/10.1017/S1351324919000111 Published online by Cambridge University Press

Table 4. QE results for EN–NL language pair. Word embedding models learned from surface forms of words are indicated with "(S)" and models learned from lemmas with "(L)"

System	F1_BAD	F1_OK	F1_MULTI
UniGRU-MS (Tezcan *et al.*, 2017a)[a]	0.4331	0.8638	0.3741
BiGRU-MS	0.4485	0.8714	0.3986
BiGRU-W2V_SONAR (S)	0.4258	0.8531	0.3631
BiGRU-MS+W2V_SONAR (S)	0.4625	0.8735	0.4040
BiGRU-MS+W2V_DPC (S)	0.4683	0.8734	0.4090
BiGRU-MS+W2V_DPC+SONAR (S)	0.4651	**0.8776**	0.4082
BiGRU-MS+W2V_SONAR.33M (S)	0.4652	0.8743	0.4067
BiGRU-MS+W2V_SONAR.33M (L)	0.4606	0.8723	0.4018
BiGRU-MS+W2V_DPC (L)	**0.4711**	0.8736	**0.4116**
BiGRU-MS+W2V_DPC+SONAR.33M (L)	0.4693	0.8746	0.4104

[a] The performance of this system is slightly worse than the results reported by Tezcan *et al.* (2017a) due to using GRU layers of size 25, instead of 50. Setting the layer sizes to 25 reduced the training times considerably with minimal loss in performance.

annotating multiple errors on the same text span, further analysis reveals that 64% of all accuracy errors in this data set overlap with at least one fluency error. This means that only a small portion of erroneous words in the target text (corresponding to 36% of the accuracy errors and 11.5% of all errors) can be detected only by analyzing the source text together with the SMT output. As a result, the erroneous words corresponding to 88.5% of all errors in this data set can be detected using monolingual information alone. These statistics indicate that monolingual information alone can effectively be used, not only to detect fluency errors made by SMT systems but also to estimate overall post-editing effort. However, the question remains whether similar error statistics can be observed in other data sets, consisting of SMT output obtained for other language pairs and domains. In this study, while we can analyze the error statistics in the EN–NL data set, no explicit information is available for the EN–DE data set.

The experiments performed for both language pairs on the task of predicting overall post-editing effort are similar to the ones we did on the task of detecting fluency errors: Firstly, we analyze the impact of combining morpho-syntactic features and word embeddings on this task. Our hypothesis is that word embeddings provide useful information on top of morpho-syntactic features and can capture nongrammatical errors. We expect the combination of the two types of information to improve QE results on the task of predicting overall post-editing effort. Secondly, we analyze the impact of learning word embedding models from smaller in-domain data compared to larger general-domain data. Finally, we test whether learning word embedding models from lemmas, instead of surface forms of words, maximize the estimation performance.

We provide the results obtained from the QE systems we built for the EN–NL and EN–DE language pairs in Tables 4 and 5, respectively. Similarly to the task of detecting fluency errors, we use F1_MULTI as the main evaluation criterion, while we also report F1_BAD and F1_OK scores. We train each NN three times with random weight initialization and calculate the average scores for each evaluation criterion. Tables 4 and 5 consist of three sections: an upper section which provides the results obtained by using either morpho-syntactic features or word embeddings in isolation, and middle and lower sections which show the impact on QE performance of using different word embedding models in combination with morpho-syntactic features.

From the first row in Table 4, we see that using morpho-syntactic features alone within unidirectional GRU layers, as described by Tezcan *et al.* (2017*a*), results in a strong baseline, with

https://doi.org/10.1017/S1351324919000111 Published online by Cambridge University Press

Table 5. QE results for EN–DE language pair

System	F1_BAD	F1_OK	F1_MULTI
UniGRU-MS (Tezcan *et al.* 2017a)[a]	0.4219	0.8311	0.3506
BiGRU-MS	0.4365	0.8358	0.3648
BiGRU-W2V_WIKI (S)	0.4417	0.8387	0.3704
BiGRU-MS+W2V_WIKI (S)	0.4559	0.8439	0.3847
BiGRU-MS+W2V_WMT/AD (S)	**0.4766**	0.8294	0.3954
BiGRU-MS+W2V_WMT/AD+WIKI (S)	0.4704	0.8396	0.3950
BiGRU-MS+W2V_WIKI (L)	0.4625	0.8218	0.3801
BiGRU-MS+W2V_WMT/AD (L)	0.4679	**0.8503**	**0.3979**
BiGRU-MS+W2V_WMT/AD+WIKI (L)	0.4705	0.8422	0.3962

[a] The performance of this system is slightly worse than the results reported by Tezcan *et al.* (2017*a*) due to using GRU layers of size 50, instead of 100.

an F1_MULTI score of 0.3741. Applying bidirectional GRU layers instead (BiGRU-MS) improves the results further to 0.3986. In the top section, we also see that word embeddings, when used in isolation, are outperformed by morpho-syntactic features on this task for the EN–NL language pair (BiGRU-W2V_SONAR (S)). In Table 1, we notice that the majority of words annotated with fluency errors in the SCATE corpus are associated with grammatical errors. Even though morpho-syntactic features alone are not able to capture semantic properties of words, the dominance of grammatical errors in this data set can explain the benefit of using these features over word embeddings for this task.

In the middle section, we see that combining word embeddings with morpho-syntactic features (BiGRU-MS+W2V_SONAR (S)) achieves better results than using either type of information alone. Similar to our observations on the task of detecting fluency errors, learning word embedding models from a smaller in-domain data set boosts the QE performance further (BiGRU-MS+W2V_DPC (S)). Finally, the lower section shows us that using lemmas is a better choice than using surface forms of words, when learning word embedding models for this task. Again, the highest estimation performance is achieved by the system which learns word embedding models from the lemmas and the in-domain data set (BiGRU-MS+W2V_DPC(L)), with an F1_score of 0.4116.

The QE results obtained on the EN–DE language pair are especially interesting as we do not have statistics about the errors made by the SMT system, which generated the German translations. Similar to the previous results, the upper section in Table 5 shows that using bidirectional GRU layers has a positive impact on performance (BiGRU-MS vs. UniGRU-MS). However, on the German SMT output, unlike the Dutch output, word embeddings outperform morpho-syntactic features for detecting post-editing effort, when both types of information are used in isolation. There are several differences between the two experiments on the EN–NL and EN–DE language pairs, which could explain this difference. The difference in the target languages, domains, and the SMT systems used can all have an impact on the error profiles of the two SMT outputs, and as a result, the amount of specific error types in both data sets can differ. If grammatical errors are not as dominant in the WMT'16 SMT output as in the SCATE corpus of MT errors, it would not be surprising word embeddings to outperform morpho-syntactic features on this data set. On the other hand, given that we extract the morpho-syntactic features on two different languages using different toolkits, the quality of the tags obtained for PoS, morphology, and dependency relations can be different. Nevertheless, combining morpho-syntactic features with word embeddings still provides better results than using either type of information alone on both language pairs, as we can see in the middle section (BiGRU-MS+W2V_WIKI (S)).

https://doi.org/10.1017/S1351324919000111 Published online by Cambridge University Press

Table 6. Performance of the proposed method in comparison to competing systems in the WMT'16 shared task on word-level QE

System	Rank	F1_BAD	F1_OK	F1_MULTI
UNBABEL/ensemble	1	0.5599	0.8845	0.4952
UGENT-LT3_RF	3	0.4917	0.8357	0.4110
BiGRU-MS+W2V_WMT/AD (L)	**–**	**0.4679**	**0.8503**	**0.3979**
UGENT-LT3_ENS	4	0.4642	8212	0.3812
CDACM/RNN	8	0.4192	0.8421	0.3531
Tezcan *et al.*, 2017	–	0.4219	0.8311	0.3506
BASELINE	11	0.3682	0.8800	0.3240
RTM_2.en-de_s4_RTM-GLMd	14	0.3067	0.8883	0.2725

Similar to previous results we obtained on the EN–NL language pair, learning word embeddings from the small but in-domain data set boosts the estimation performance (BiGRU-MS+W2_WMT/AD (S)). Combining this data set with the out-of-domain data deteriorates the results (BiGRU-MS+W2_WMT/AD+WIKI (S)). Finally, learning word embedding models from lemmas seem to improve the results slightly as the best system achieves an F1_MULTI score of 0.3979 (BiGRU-MS+W2_WMT/AD (L)).

The performance of the EN–DE system can be better understood when compared to the participating systems in the WMT'16 shared task on word-level QE. In Table 6, we provide the results for a number of systems from the shared task (Bojar *et al.* 2016) and the ranks they obtained, together with the best system we built using the proposed architecture (MS+W2V_WMT/AD (L)). We also include the results of the system proposed by Tezcan *et al.* (2017*a*), which utilizes morpho-syntactic features in isolation, as an additional baseline.

By using monolingual information given in the SMT output alone, the proposed system outperforms the baseline system in this shared task, which uses features representing monolingual and bilingual properties of each translated text. In fact, the proposed system achieves the 4th best result in the WMT'16 shared task and outperforms 11 participating systems out of 14. From all the systems that were submitted to WMT'16, the NN architecture used in the CDACM/RNN system shows similarities to the proposed NN architecture proposed in this study. Both systems use GRU layers and provide surface *n*-grams around the target word as input, in the form of pre-trained word embeddings. On the other hand, while the CDACM/RNN system combines this information with information obtained from the source text, in the form of word embeddings for the context words centered around the source alignment of the target text, the proposed architecture in this study combines this information with syntactic *n*-grams and morpho-syntactic word representations obtained on the target text. From this perspective, Table 6 confirms the informativeness of monolingual information on predicting post-editing effort of SMT output. More specifically, it shows us that an RNN system that uses only monolingual information, in the form of morpho-syntactic word representations and word embeddings within surface and syntactic *n*-grams, outperforms a similar RNN architecture which uses bilingual information in the form of word embeddings within surface *n*-grams, instead (+0.0448 F1_SCORE).

6. Error analysis

In Sections 5.1 and 5.2, we have observed the effectiveness of using monolingual information alone on estimating word-level quality of SMT output. Despite the improvements achieved over different baselines we considered, the results indicate that word-level QE systems still contain errors, which has also been observed in the past on the task of predicting post-editing effort (Bojar *et al.* 2015 2016).

https://doi.org/10.1017/S1351324919000111 Published online by Cambridge University Press

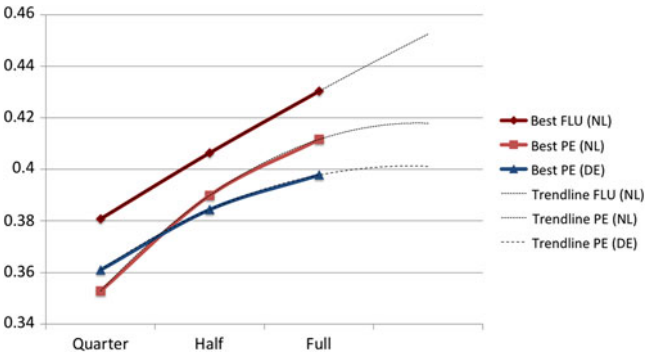


Figure 3. The impact of data size on estimation performance.

A main source of errors in this study is the limited amount of training data we have for the two language pairs, as NNs require large amounts of training data to sufficiently learn the weights between neurons. Although the proposed NN architecture was able to outperform the majority of the submitted systems in the WMT'16 shared task on word-level QE with the provided data sets, the impact of the data size on its performance is yet to be studied. Moreover, the EN–NL data set we used in this study can be considered relatively small, consisting of only 52,202 training samples in each cross-validation fold. In this section, we first analyze the impact of the size of the available training data on estimation performance. For this purpose, we train the best systems for each task and on each language pair by using 25% (every 4th sample) and 50% (every 2nd sample) of the available training data.[k] We provide the results in Figure 3.

The results in Figure 3 confirm that reducing the amount of samples available in training data to 50% and 25% reduces the estimation performance (F1_MULTI score) each time on all tasks and for each language pair. On the other hand, the trend lines drawn for each system indicate that the expected increase in estimation performance is different in each case. While comparing all systems at once is difficult given the differences in the language pairs and the tasks, we can make a pairwise comparison. The best system obtained for the EN–DE language pair on the task of predicting post-editing effort (Best PE (DE)) shows a trend with fewer gains, compared to the system on the EN–NL language pair (Best PE (NL)), where the data size is doubled. This is not surprising as the EN–DE data size is already approximately four times the size of the EN–NL data set we use (consisting of approx. 210k and 52k samples, respectively) and we can expect the positive impact of doubling the data size to decrease for larger data sets.

For the EN–NL data set, when we draw a comparison based on the type of the QE task, we notice that the trend line for the best system obtained for detecting fluency errors (Best FLU (NL)) shows a constant increase, whereas the estimation performance of the best system obtained for predicting post-editing effort (Best PE (DE)) seems to increase with a decreasing rate. This can be explained with the type of translation errors captured by the two QE tasks: it seems that using monolingual information for detecting errors of monolingual nature (fluency errors) benefits more from an increase in data size than predicting post-editing effort, which represents correction of all error types in the SMT output. Even though it is difficult to make a fair comparison between different tasks, we see that the estimation performance of "Best FLU (NL)" is already better than "Best PE (NL)" when the same data sets are used (F1_MULTI scores of 0.4303 and 0.4116, respectively). The trends in Figure 3 further suggest that the difference between their performance will further increase in favor of "Best PE (NL)", with increasing data sizes.

When we analyzed the best systems described above, on the task of predicting post-editing effort, we discovered a common type of mistake especially made by the EN–DE system. As mentioned in Section 5.2, while extracting error labels for this task, inserted words during post-editing are not marked erroneous in the MT output simply because they do not exist in

[k]The context windows we use in the NN architecture for each target word are built prior to reducing the data sizes.

https://doi.org/10.1017/S1351324919000111 Published online by Cambridge University Press

SRC: Available only for monochrome bitmap images .
MT_PRED: Diese Option ist nur für **monochrome Bitmapbilder** .
MT_PE: Diese Option ist nur für monochrome Bitmapbilder verfügbar .

Figure 4. QE system locating the context of missing words, where insertions are made during post-editing.

MT_LABELS: Diese Option ist nur für monochrome **Bitmapbilder .**

Figure 5. Marking the context of missing words, which require an "insertion" in the MT output.

the MT output yet.[l] Therefore, the resulting training set is not able to inform the systems about the missing words in the MT output, that is, the words inserted during post-editing. Interestingly, the EN–DE system[m] we built seemed to successfully locate the context of missing words in many cases, even though no information was available regarding this error type in the training data. In fact, we observed that the ratio of the false positive (predicting a label as BAD whereas the correct label is OK) made on the labels corresponding to the left and right context words of the insertions made during post-editing over all other labels was 1.77:1 and 1.38:1 for EN–DE and EN–NL, respectively. We illustrate this phenomenon in Figure 4.

In Figure 4, we see a source sentence (SRC), the corresponding SMT output containing the predicted errors by the QE system in bold (MT_PRED), and the post-edited version of the SMT output (MT_PE). In the post-edited version, we can see that the word "verfügbar (available)" has been inserted to the SMT output. In this example, the QE system was able to localize the context of the missing word (by marking the words "monochrome Bitmapbilder"), even though it did not encounter error labels for such errors in the training data.

One possible explanation for this system's ability to approximate the location of insertions is that the surface and syntactic context windows enables the NN architecture to learn contextual anomalies regarding the fluency of the SMT output to a certain extent, even when no error labels are provided. To see the effect of this phenomenon on QE performance, we retrained the two best systems we built for predicting post-editing effort on EN–NL and EN–DE with a difference: in the SMT output, we additionally marked the words to the left and right of all the inserted words during post-editing as BAD, which is illustrated in Figure 5.

In Figure 5, even though they are not associated with errors themselves, the word "Bitmapbilder (bitmap images)" and the punctuation token "." have been marked as erroneous to indicate the context of a missing word in the SMT output (MT_LABELS). Marking more words as erroneous introduced 5.9% and 6.7% additional BAD labels in the training data for the EN–NL and EN–DE language pairs, respectively[n]). In Tables 7 and 8, we compare the estimation performance of the retrained systems with the best systems we obtained for both language pairs in Section 5.2. While Table 7 provides the QE performances on the full test sets, Table 8 provides the QE performances only on the labels for the sentences that contain at least one *insertion*. To have a better understanding of the systems' error profiles, in addition to the F1 scores, in both tables, we also provide the precision and recall scores for the BAD label.

What we see in Table 7 is that marking the context of inserted words during post-editing provides both systems additional useful information as the overall performance of both systems improves with respect to F1_MULTI. However, the level of improvement is different for both systems. The EN–DE system (PE-effort DE) seems to benefit from this additional information more compared to the EN–NL system (PE-effort NL) for all evaluation criteria (except F1_OK, which deteriorates in both cases). The improvement, especially for the Recall_BAD and F1_MULTI scores, is much higher for the EN–DE system (+0.0978 Recall_BAD and +0.0322 F1_MULTI) than for the EN–NL system (+0.0170 Recall_BAD and +0.0079 F1_MULTI). Given that the relative

[l]In the SCATE corpus of SMT errors, the context of the missing word errors are marked by annotating the word after which an insertion must be made.
[m]This analysis was made for the best system BiGRU-MS+W2V_WMT/AD (L), as listed in Table 5.
[n]This analysis excludes the context words which were already annotated as BAD due to other post-edit operations.

https://doi.org/10.1017/S1351324919000111 Published online by Cambridge University Press

Table 7. The impact of retraining systems with additional labels that mark the context of insertions made during post-editing. The retrained systems are indicated with "(ins.)". Evaluation is performed on the full test sets

System	Pre_BAD	Rec_BAD	F1_BAD	F1_OK	F1_MULTI
PE-effort NL	0.4252	0.5285	0.4711	**0.8736**	0.4116
PE-effort NL (ins.)	**0.4684**	**0.5455**	**0.5040**	0.8322	**0.4195**
PE-effort DE	0.4178	0.5315	0.4679	**0.8503**	0.3979
PE-effort DE (ins.)	**0.4775**	**0.6293**	**0.5430**	0.7921	**0.4301**

Table 8. The impact of retraining systems with additional labels that mark the context of insertions made during post-editing. The retrained systems are indicated with "(ins.)". Evaluation is performed only on the labels for the sentences that contain *insertions*

System	Pre_BAD	Rec_BAD	F1_BAD	F1_OK	F1_MULTI
PE-effort NL	0.3651	**0.6331**	0.4631	**0.8097**	0.3750
PE-effort NL (ins.)	**0.5032**	0.5772	**0.5377**	0.7916	**0.4256**
PE-effort DE	0.2899	0.4842	0.3627	**0.7683**	0.2787
PE-effort DE (ins.)	**0.4482**	**0.5077**	**0.4761**	0.7308	**0.3479**

amount of newly introduced BAD labels in both training data is roughly similar (5.9% and 6.7% in EN–NL and EN–DE, respectively, the difference in the performance improvements can be attributed, at least partially, to the different data sets used for both language pairs. It is also possible that the QE systems benefit more from using this additional information with increasing sizes of the training data.

To get a better understanding about how modifying error labels affects the QE performance, in a second experiment, we evaluate only the labels that come from the sentences, in which new words have been inserted during post-editing. As a result, in this experiment, 35,449 labels (from 1435 sentences) from the EN–NL data set and 19,557 labels (from 1066 sentences) from the EN–DE set have been evaluated, which correspond to 61% and 57% of all labels in the test sets for EN–NL and EN–DE, respectively.[o] Table 8 provides the QE performances obtained on this reduced test set.

Similar to Table 7, Table 8 shows that introducing the additional BAD labels to the data sets leads to overall improvements in QE performance for both language pairs. Moreover, for both language pairs, Table 8 shows that the sentences, in which *insertions* are made, benefit more from the modification of error labels, as the improvements in QE performance become more outspoken for the labels that come from such sentences (+0.0506 and +0.0692 F1_MULTI for EN–NL and EN–DE data sets, respectively).

7. Conclusions and future work

The aim of this study was to investigate the effectiveness of using monolingual information contained in the machine-translated text to estimate its quality. To this end, we presented an RNN architecture which takes morpho-syntactic and distributed word representations obtained from monolingual data alone as input to estimate word-level quality of SMT. Our approach combines PoS, morphology, and dependency information extracted from words with the semantic

[o]Given that EN–NL experiments were evaluated using 10-fold cross validation, without using an explicit test set, for this evaluation these labels have been extracted from the whole data set.

https://doi.org/10.1017/S1351324919000111 Published online by Cambridge University Press

relationships modeled by word embeddings as separate input vectors to the NN. We tested the proposed method first on word-level detection of fluency errors in SMT output for EN–NL. Our results show that the proposed method is able to outperform alternative NN architectures on detecting all fluency errors at once. We observed that splitting morpho-syntactic information and semantics modeled by word embeddings leads to better neural models and estimation performance. Moreover, we also showed that careful selection of word representations is key to building accurate error detection systems on specific error types.

Secondly, we utilized this method to estimate the overall post-editing effort by detecting words that have been modified during post-editing for the EN–NL and EN–DE language pairs. We demonstrated that by only using monolingual information, successful QE systems can be built to predict overall post-editing effort involved in correcting the SMT output. The proposed method achieved better estimation performance than the majority of the competing systems in the WMT'16 shared task on QE. The high predictive power of monolingual information additionally revealed the importance of fluency errors on defining overall SMT quality.

Other key findings in this study unveiled the impact on QE performance of the type of information and the resources from which word embeddings are learned. On both EN–NL and EN–DE language pairs and for both tasks, learning word embeddings from lemmas had a positive impact on estimation performance, when they were combined with morpho-syntactic features. Furthermore, learning word embeddings from relatively small but in-domain data sets improved the results in all cases, compared to training them from large but general-domain data.

An obvious direction for future work is to add bilingual information to the proposed architecture. This can be done in different ways for both word- and sentence-level QE tasks. On the one hand, the same NN architecture can be applied to the task of predicting post-editing effort by extending it with additional bilingual information. To this end, we would like to analyze the impact of adding different types of source-related information, such as morpho-syntactic word representations and word embeddings, as input to the proposed NN architecture.

On the other hand, the proposed method can be used as a main component in two-step sentence-level QE-systems which consist of detecting specific error types in the first step and use them to estimate a global score on sentence-level quality in a second step. The proposed method can be combined with a second error detection system that focuses on adequacy errors to build such a two-step QE system. Two-step QE systems would not only inform the users on the nature of errors that are made by SMT systems on word level but also would provide reasons why estimated sentence-level quality is better in certain cases than others. This would lead to more informative QE systems compared to existing systems that try to estimate overall sentence-level quality with a single score. Moreover, the informative QE approach is not limited to SMT applications and it can be applied to different MT paradigms, provided that similar data sets with manual error annotations are available in the future.

Finally, the tasks of detecting fluency errors in MT and detecting errors made by non-native speakers can be considered similar, given that they both focus on the monolingual aspects of a single language. From this perspective, the results of this study might also be useful to develop supervised error detection systems which can locate the mistakes in texts that are written by non-natives.

Acknowledgements. This research has been carried out in the framework of the SCATE project funded by the Flemish government agency IWT (IWT-SBO 130041).

References

Abadi M., *et al.* (2016). Tensorflow: Large-Scale machine learning on heterogeneous distributed systems. In *CoRR, abs/1603.04467*.

Abdelsalam A., Bojar O. and El-Beltagy S. (2016). Bilingual embeddings and word alignments for translation quality estimation. In *Proceedings of the First Conference on Machine Translation.* Berlin, Germany: Association for Computational Linguistics, pp. 764–771.

https://doi.org/10.1017/S1351324919000111 Published online by Cambridge University Press

Anastasakos T., Kim Y.-B. and Deoras A. (2014). Task specific continuous word representations for mono and multilingual spoken language understanding. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3246–3250

Avraham O. and Goldberg Y. (2017). The interplay of semantics and morphology in word embeddings. In *CoRR, abs/1704.01938*. Retrieved from http://arxiv.org/abs/1704.01938

Avramidis E. (2017). Comparative quality estimation for machine translation observations on machine learning and features. *The Prague Bulletin of Mathematical Linguistics* **108**(1), 307–318.

Axelrod A., He X. and Gao J. (2011). Domain adaptation via pseudo in-domain data selection. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 355–362). Stroudsburg, PA, USA: Association for Computational Linguistics. Retrieved from http://dl.acm.org/citation.cfm?id=2145432.2145474

Bahdanau D., Cho K. and Bengio Y. (2014). Neural machine translation by jointly learning to align and translate. In *CoRR, abs/1409.0473*. Retrieved from http://arxiv.org/abs/1409.0473

Bentivogli L., Bisazza A., Cettolo M. and Federico M. (2016). Neural versus phrase-based machine translation quality: A case study. In *CoRR, abs/1608.04631*.

Bertoldi N. and Federico M. (2009). Domain adaptation for statistical machine translation with monolingual resources. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 182–189. Retrieved from http://dl.acm.org/citation.cfm?id=1626431.1626468

Blain F., Scarton C. and Specia L. (2017). Bilexical embeddings for quality estimation. In *Proceedings of the Second Conference on Machine Translation*, pp. 545–550.

Blatz J., et al. (2004). Confidence estimation for machine translation. In *Proceedings of the 20th International Conference on Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics. Retrieved from https://doi.org/10.3115/1220355.1220401

Bohnet B. and Nivre J. (2012). A transition-based system for joint part-of speech tagging and labeled non-projective dependency parsing. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, pp. 1455–1465.

Bojar O., et al. (2014). Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pp. 12–58.

Bojar O., et al. (2015). Findings of the 2015 workshop on statistical machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Lisbon, Portugal: Association for Computational Linguistics, pp. 1–46. Retrieved from http://aclweb.org/anthology/W15-3001

Bojar O., et al. (2016). Findings of the 2016 conference on machine translation. In *Proceedings of the Frst Conference on Machine Translation, WMT 2016, Colocated with ACL 2016, Berlin, Germany*, pp. 131–198.

Bojar O., et al. (2017). Findings of the 2017 conference on machine translation (WMT17). In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 169–214.

Castilho S., Moorkens J., Gaspari F., Calixto I., Tinsley J. and Way A. (2017). Is neural machine translation the new state of the art? *The Prague Bulletin of Mathematical Linguistics* **108**(1), 109–120.

Cho K., van Merriënboer B., Gülçehre Ç., Bahdanau D., Bougares F., Schwenk H. and Bengio Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, pp. 1724–1734.

Chung J., Gülçehre Ç., Cho K. and Bengio Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. In *CoRR, abs/1412.3555*. Retrieved from http://arxiv.org/abs/1412.3555

Costa Â., Ling W., Luıs T., Correia R. and Coheur L. (2015). A linguistically motivated taxonomy for machine translation error analysis. *Machine Translation* **29**(2), 127–161.

Daems J., Macken L. and Vandepitte S. (2014). On the origin of errors: A finegrained analysis of mt and pe errors and their relationship. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*. European Language Resources Association (ELRA), pp. 62–66.

Daems J., Vandepitte S., Hartsuiker R.J. and Macken L. (2017). Identifying the machine translation error types with the greatest impact on post-editing effort. *Frontiers in Psychology* **8**, 1282. http://journal.frontiersin.org/article/10.3389/fpsyg.2017.01282

de Almeida G. (2013). *Translating the post-editor: An investigation of post-editing changes and correlations with professional experience across two romance languages* (Unpublished doctoral dissertation). Dublin City University.

Gandrabur S. and Foster G. (2003). Confidence estimation for translation prediction. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*. Association for Computational Linguistics, pp. 95–102.

Glorot X. and Bengio Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10). Society for Artificial Intelligence and Statistics*.

Graham Y., Baldwin T., Moffat A. and Zobel J. (2014). Is machine translation getting better over time? In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 443–451.

https://doi.org/10.1017/S1351324919000111 Published online by Cambridge University Press

Hokamp C. (2017). Ensembling factored neural machine translation models for automatic post-editing and quality estimation. In *CoRR, abs/1706.05083*.

Hokamp C., Calixto I., Wagner J. and Zhang J. (2014). Target-centric features for translation quality estimation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pp. 329–334.

Jones K.S. and Galliers J.R. (1995). *Evaluating Natural Language Processing Systems: An Analysis and Review,* vol. 1083. Germany: Springer Science & Business Media.

Junczys-Dowmunt M. and Grundkiewicz R. (2016). Log-linear combinations of monolingual and bilingual neural machine translation models for automatic post-editing. In *CoRR, abs/1605.04800*.

Kim H. and Lee J.-H. (2016). Recurrent neural network based translation quality estimation. In *Proceedings of the first conference on machine translation: Volume 2, shared task papers*, pp. 787–792.

Kim H., Lee J.-H. and Na S.-H. (2017). Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation. In *Proceedings of the Second Conference on Machine Translation*, pp. 562–568.

Klubička F., Toral A. and Sánchez-Cartagena V.M. (2017). Fine-grained human evaluation of neural versus phrase-based machine translation. *The Prague Bulletin of Mathematical Linguistics* **108**(1), 121–132.

Koponen M., Aziz W., Ramos L. and Specia L. (2012). Post-editing time as a measure of cognitive effort. In *AMTA 2012 Workshop on Post-Editing Technology and Practice (WPTP 2012)*. San Diego, USA, pp. 11–20.

Kreutzer J., Schamoni S. and Riezler S. (2015). QUality Estimation from ScraTCH (QUETCH): Deep learning for word-level translation quality estimation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation, WMT@EMNLP 2015, Lisbon, Portugal*, pp. 316–322.

Kusner M., Sun Y., Kolkin N. and Weinberger, K.Q. (2015). From word embeddings to document distances. In Blei, D. and & Bach F. (eds), *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*. JMLR Workshop and Conference Proceedings, pp. 957–966.

Li J., Li J., Fu X., Masud M. and Huang J.Z. (2016). Learning distributed word representation with multi-contextual mixed embedding. *Knowledge-Based Systems* **106**, 220–230. http://www.sciencedirect.com/science/article/pii/S0950705116301435; doi: http://dx.doi.org/10.1016/j.knosys.2016.05.045

Logacheva V., Hokamp C. and Specia L. (2016a). Marmot: A toolkit for translation quality estimation at the word level. In *Proceedings of the 10th Edition of the Language Resources and Evaluation Conference (LREC)*.

Logacheva V., Lukasik M. and Specia L. (2016b). Metrics for evaluation of word-level machine translation quality estimation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (volume 2: Short papers)*, pp. 585–590.

Lommel A.R., Uszkoreit H. and Burchardt A. (2014). Multidimensional Quality Metrics (MQM). *Tradumàtica **12***, 455–463.

Ma W. and McKeown K. (2012). Detecting and correcting syntactic errors in machine translation using feature-based lexicalized tree adjoining grammars. *IJCLCLP* **17**(4), pp. 1–14.

Macken L., De Clercq O. and Paulussen H. (2011). Dutch parallel corpus: A balanced copyright-cleared parallel corpus. *Meta: Journal des traducteursMeta:/ Translators' Journal* **56**(2), 374–390.

Martins A.F., Astudillo R.F., Hokamp C. and Kepler F. (2016). Unbabel's participation in the WMT16 word-level translation quality estimation shared task. In *Proceedings of the First Conference on Machine Translation*. Berlin, Germany: Association for Computational Linguistics, pp. 806–811.

Martins A.F., Kepler F. and Monteiro J. (2017). Unbabel's participation in the WMT17 translation quality estimation shared task. In *Proceedings of the Second Conference on Machine Translation,* pp. 569–574.

Mikolov T., Chen K., Corrado G. and Dean J. (2013). Efficient estimation of word representations in vector space. In *CoRR, abs/1301.3781*.

Oostdijk N., Reynaert M., Monachesi P., Noord G.V., Ordelman R. and Schuurman I. (2008). From DCoi to SoNaR: A reference corpus for dutch. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation*.

Owczarzak K., van Genabith J. and Way A. (2007). Labelled dependencies in machine translation evaluation. In *Proceedings of the Second Workshop on Statistical Machine Translation*. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 104–111. Retrieved from http://dl.acm.org/citation.cfm?id=1626355.1626369

Patel R.N. and Sasikumar M. (2016). Translation quality estimation using recurrent neural network. In *CoRR, abs/1610.04841*.

Řehůřek R. and Sojka P. (2010). Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, pp. 45–50. Retrieved from http://is.muni.cz/publication/884893/en

Scarton C., Beck D., Shah K., Smith K.S. and Specia L. (2016). Word embeddings and discourse information for machine translation quality estimation. In *Proceedings of the First Conference on Machine Translation*. Berlin, Germany: Association for Computational Linguistics, pp. 831–837.

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

Socher R., Lin C.C., Ng A.Y. and Manning C.D. (2011a). Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*.

https://doi.org/10.1017/S1351324919000111 Published online by Cambridge University Press

Socher R., Pennington J., Huang E.H., Ng A.Y. and Manning C.D. (2011b). Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 151–161. Retrieved from http://dl.acm.org/citation.cfm?id=2145432.2145450

Specia L., Turchi M., Cancedda N., Dymetman M. and Cristianini, N. (2009). Estimating the sentence-level quality of machine translation systems. In *13th Annual Conference of the European Association for Machine Translation*. Barcelona, Spain, pp. 28–37. Retrieved from http://www.mt-archive.info/EAMT-2009-Specia.pdf

Specia L., Shah K., De Souza J.G.C., Cohn T. and Kessler F.B. (2013). QuEst - A translation quality estimation framework. In *Proceedings of the 51th Conference of the Association for Computational Linguistics (ACL), Demo Session*.

Specia L., Logacheva V. and Scarton C. (2016). *WMT16 quality estimation shared task training and development data.* Retrieved from http://hdl.handle.net/11372/LRT-1646 (LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University)

Srivastava N., Hinton G., Krizhevsky A., Sutskever I. and Salakhutdinov R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* **15**(1), 1929–1958.

Stymne, S. and Ahrenberg, L. (2010). Using a grammar checker for evaluation and postprocessing of statistical machine translation. In *Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10)*. European Language Resources Association (ELRA).

Tezcan A., Hoste V. and Macken L. (2016). Detecting grammatical errors in machine translation output using dependency parsing and treebank querying. *Baltic Journal of Modern Computing* **4**(2), 203–217.

Tezcan A., Hoste V. and Macken, L. (2017a). A neural network architecture for detecting grammatical errors in statistical machine translation. *The Prague Bulletin of Mathematical Linguistics* **108**, 133–145.

Tezcan A., Hoste V. and Macken L. (2017b). Scate taxonomy and corpus of machine translation errors. In Pastor G.C. and Durán-Muñoz I. (eds), *Trends in e-Tools and Resources for Translators and Interpreters,* vol. 45. Leiden, The Netherlands: Brill | Rodopi, pp. 219–244.

Tieleman T. and Hinton G. (2012). *Lecture 6.5–RmsProp: Divide the gradient by a running average of its recent magnitude*. COURSERA: Neural Networks for Machine Learning.

Toury G. (2000). The nature and role of norms in translation. *The Translation Studies Reader* **2**, 198–212.

Turian J., Ratinov L. and Bengio Y. (2010). Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 384–394.

Ueffing N. and Ney H. (2005). Application of word-level confidence measures in interactive statistical machine translation. In *Proceedings of EAMT 2005 10th Annual Conference of the European Association for Machine Translation*, pp. 262–270.

Van Noord G. (2006). At last parsing is now operational. In *TALN06. Verbum ex machina. Actes de la 13e conference sur le traitement automatique des langues naturelles*, pp. 20–42.

Vilar D., Xu J., D'haro L.F. and Ney H. (2006). Error analysis of statistical machine translation output. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC-2006)*. Genoa, Italy: European Language Resources Association (ELRA). (ACL Anthology Identifier: L06–1244)

White J.S. (1995). Approaches to black box MT evaluation. In *Proceedings of Machine Translation Summit V*, vol. 10.

Wolk K. and Marasek K. (2015). Building subject-aligned comparable corpora and mining it for truly parallel sentence pairs. In *CoRR, abs/1509.08881*. Retrieved from http://arxiv.org/abs/1509.08881

Xu J., Deng Y., Gao Y. and Ney H. (2007). Domain dependent statistical machine translation. In *Proceedings of the MT Summit XI,* pp. 515–520.

Cite this article: Tezcan A, Hoste V and Macken L (2020). Estimating word-level quality of statistical machine translation output using monolingual information alone. *Natural Language Engineering* **26**, 73–94. https://doi.org/10.1017/S1351324919000111

https://doi.org/10.1017/S1351324919000111 Published online by Cambridge University Press