Feature-Enriched Character-Level Convolutions for Text Regression

Gustavo Henrique Paetzold and **Lucia Specia**
Department of Computer Science
University of Sheffield, UK
{g.h.paetzold,l.specia}@sheffield.ac.uk

Abstract

We present a new model for text regression that seamlessly combine engineered features and character-level information through deep parallel convolution stacks, multi-layer perceptrons and multi-task learning. We use these models to create the SHEF/CNN systems for the sentence-level Quality Estimation task of WMT 2017 and Emotion Intensity Analysis task of WASSA 2017. Our experiments reveal that combining character-level clues and engineered features offers noticeable performance improvements over using only one of these sources of information in isolation.

1 Introduction

Text regression consists in estimating a numeric label based on information available from the text. The label can represent any abstract property of said text: its appropriateness, sentiment, fluency, simplicity, quality, etc. Due to their wide applicability in both research and industry, some of these tasks have been gaining a lot of attention. These include Quality Estimation and Emotion Intensity Analysis, which are the subjects of shared tasks held at the WMT 2017 conference[1] and WASSA 2017 workshop[2] (Mohammad and Bravo-Marquez, 2017), respectively.

In Quality Estimation (QE), one attempts to estimate the quality of a machine translated text based on the information that can be extracted from the original sentence and its translation. The task has many variants, given that the quality of a translation can be estimated at word, phrase, sentence or even document level. Quality estimates can be incorporated in Machine Translation (MT) decoding or used for re-ranking of top candidates, for example, allowing for a more intelligently guided translation process (Avramidis, 2012), or they can be used to help human translators decide which automatic translations are worth post-editing, and which should be re-translated from scratch (Turchi et al., 2015). Sentence-level QE is the most popular variant, mostly due the fact that most modern statistical and neural MT systems translate one sentence at a time. In this task, the input is the original-translated sentence pair and the output is some numeric label that represents quality. The most commonly used label is HTER, which measures the human post-editing effort required to fix the translation in question (Snover et al., 2006).

As shown in (Bojar et al., 2016), the performance of QE approaches submitted to the WMT shared tasks have steadily improved in recent years. However, the nature of these approaches have not changed much: most of the top ranking systems employ well-known regression methods and extensive feature engineering. Some of the most notable examples are the RTM systems of WMT 2014 and 15, which managed to reach the top of the ranks by employing Referential Translation Machines trained with SVMs for regression (Bicici, 2016). The LORIA (Langlois, 2015) and YSDA (Kozlova et al., 2016) systems of WMT 2015 and 2016, respectively, achieved similar performance by also pairing SVMs with many resource-heavy features.

Neural Networks for sentence-level QE were introduced in WMT 2016 with the SimpleNets (Paetzold and Specia, 2016) and POSTECH (Kim and Lee, 2016) systems. While the SimpleNets system uses sequence-to-label LSTMs to predict the quality of a translation's n-grams and then

[1] http://www.statmt.org/wmt17

[2] http://optima.jrc.it/wassa2017

Proceedings of the Conference on Machine Translation (WMT), Volume 2: Shared Task Papers, pages 575–581
Copenhagen, Denmark, September 711, 2017. ©2017 Association for Computational Linguistics

combines them, the POSTECH system learns quality labels at word-level using a sequence-to-sequence model, and then combines them with a sequence-to-label model to predict quality at sentence-level. Though very interesting and distinct strategies, neither of them managed to outperform the best scoring SVM-based approach of WMT 2016.

In the task of Emotion Intensity Analysis (EIA), Neural Networks have not yet been successfully employed. Unlike typical Sentiment Analysis tasks, which are set up as either binary or multi-class classification problems that require one to determine the opinion or sentiment in a given text, EIA aims at quantifying a certain emotion in a text, such as fear, anger, joy, sadness, etc. In the Emotion Intensity shared task of SemEval 2016 (Kiritchenko et al., 2016), which is the first of its kind, none of the five systems submitted employ neural regressors. We were also unable to find any other contributions outside the SemEval 2016 task that explore neural approaches to EIA.

Given the volume of opportunities available when it comes to neural solutions for text regression, we introduce a new neural approach for the task. We innovate by using deep convolutional networks and multi-task learning to combine character-level information from the texts at hand with engineered features. Using this approach, we create the SHEF/CNN systems for the sentence-level QE task of WMT 2017 and the Emotion Intensity Analysis task of WASSA 2017. In what follows, we describe our approach in detail.

2 Overview of Tasks

As previously mentioned, we address two text regression tasks in this paper: the sentence-level Quality Estimation task of WMT 2017 and Emotion Intensity Analysis task of WASSA 2017. The next Sections describe each of those tasks.

2.1 Quality Estimation at WMT 2017

In the sentence-level QE task of WMT 2017 participants were asked to create systems that predict the human post-editing effort required to correct an automatically translated sentence. Training, development and test sets were provided for two language pairs: English-German and German-English. The training and development sets for both language pairs are composed of 23,000/25,000 and 1,000/1,000 instances, respectively. Each instance is composed of a source (original) and target (translated) sentence pair, as well as the target's manually post-edited version and an HTER label between 0 and 1 calculated based on the post-edit. The test set is composed of 2,000 instances without post-edits nor HTER labels. For training, development and test sets the organizers made available a set of 17 baseline features.

The task is divided in two sub-tasks: scoring and ranking. In the scoring task, systems had to estimate HTER scores and were evaluated through Pearson correlation. In the ranking task, systems had to rank the translations in the test set from highest to lowest quality, and were evaluated through Spearman correlation. The main difference between the data provided for the WMT 2017 QE tasks and the data of previous editions is that, for the first time, the tasks of all QE levels (sentence, word and phrase) contain annotations for the same set of translations. Because of that, one can very intuitively employ any variety of multi-task learning approaches.

2.2 Emotion Intensity at WASSA 2017

Systems submitted to the Emotion Intensity Analysis task of WASSA 2017 were asked to estimate the intensity of various emotions felt by authors while writing tweets. Training, development and test sets were made available containing four emotions: anger, fear, joy and sadness. The size of the datasets is illustrated in Table 1.

Emotion	Train	Dev	Test
Anger	857	84	760
Fear	1,147	110	995
Joy	823	79	714
Sadness	786	74	673

Table 1: Dataset sizes for the Emotion Intensity Analysis task of WASSA 2017

Each instance is composed of a tweet and an intensity label between 0 and 1 of the emotion in question. Labels were collected through crowdsourcing. Systems were evaluated through Pearson correlation.

3 Model Architecture

Figure 1 illustrates the neural model architecture of the SHEF/CNN systems for the QE task of

WMT 2017. As it can be noticed, the model takes as input a one-hot character-level representation of the source and target, as well as a set of engineered features. As output, our model produces the numeric labels desired.

The model is divided in three main sections: a pair of deep convolution layer stacks for the source (original) and target (translated) sentences, a multi-layer perceptron for the engineered features, and a final multi-layer perceptron to combine all this information. The model used for the EIA task of WASSA 2017 is identical, except that it only has one set of convolution stacks for the tweet being analysed.

3.1 Extracting Character-Level Clues

In order to exploit the information at character-level from the text, we use a convolution architecture similar to the one introduced by (Kim et al., 2016), who successfully employ character-level information for language modelling. First we transform the one-hot character-level representation of the sentence into a sequence of character embeddings. We then feed them to a series of parallel one-dimensional convolutions of different window sizes. Each of these convolutions captures the information of character n-grams of a given length: a convolution of window size one addresses unigrams, one with size two addresses bigrams, and so on. Finally, the resulting values produced by the convolution filters are passed on to a one-dimensional max-pooling layer.

In order to capture information at different abstraction levels, we stack various convolution and max-pooling layers for each window size, thus creating a deep architecture. This deep architecture differs from the one used by (Kim et al., 2016) in the sense that they apply only one stack of convolution/max-pooling layers for each window size. The values produced by the last max-pooling layer of each window size are then flattened so that they can be easily concatenated.

The intuition behind using such an architecture lies in the assumption that sequences of characters hold important clues with respect to the text's properties, such as quality and emotion. In QE, these clues could be sequences containing morphological errors in words from the source or target sentences, or sequences in-between tokens of the target that suggest an ungrammatical segment, for example. In EIA, these clues can be emotionally charged emojis, curse words, exclamation marks, etc.

3.2 Incorporating Engineered Features

We complement character-level information with engineered features, given that the most effective QE and EIA methods in previous work heavily exploit them (Kim and Lee, 2016; Kozlova et al., 2016; Refaee and Rieser, 2016; Wang et al., 2016). To do so, we apply a simple multi-layer perceptron (MLP) over a set of input engineered features. This allows to capture abstract relations between the features provided. The output of the outermost layer is then concatenated with the flattened character-level information provided by the remainder of the network.

Finally, we pass the concatenated features and character-level information to another MLP in order for our model to be able to capture any relations between them. At the very edge of our model, we include output nodes for as many tasks as we wish to train our model over.

4 SHEF/CNN Model for QE

As illustrated in Figure 1, the sentence-level QE model employs one convolution stack for each of the source and target sides of the translation pair. We configure the model as follows:

- **Embedding size:** We train character embeddings with 50 dimensions.
- **Window range:** We use 4 parallel stacks of convolutions with window sizes from 1 to 4.
- **Convolution depth:** Each stack contains 4 pairs of convolution/max-pooling layers with 50 convolution filters each and a pool length of 4.
- **Feature MLP depth:** We stack 2 dense layers with 50 hidden units over engineered features.
- **Final MLP depth:** The MLP that combines convolutions and features is composed of 2 stacked dense layers with 50 hidden units each.
- **Engineered feature set:** We use the 17 baseline features provided by the task organizers.

This architecture was selected through experimentation. The output nodes of our multi-task QE setup predict three values:

- HTER from the sentence-level dataset;

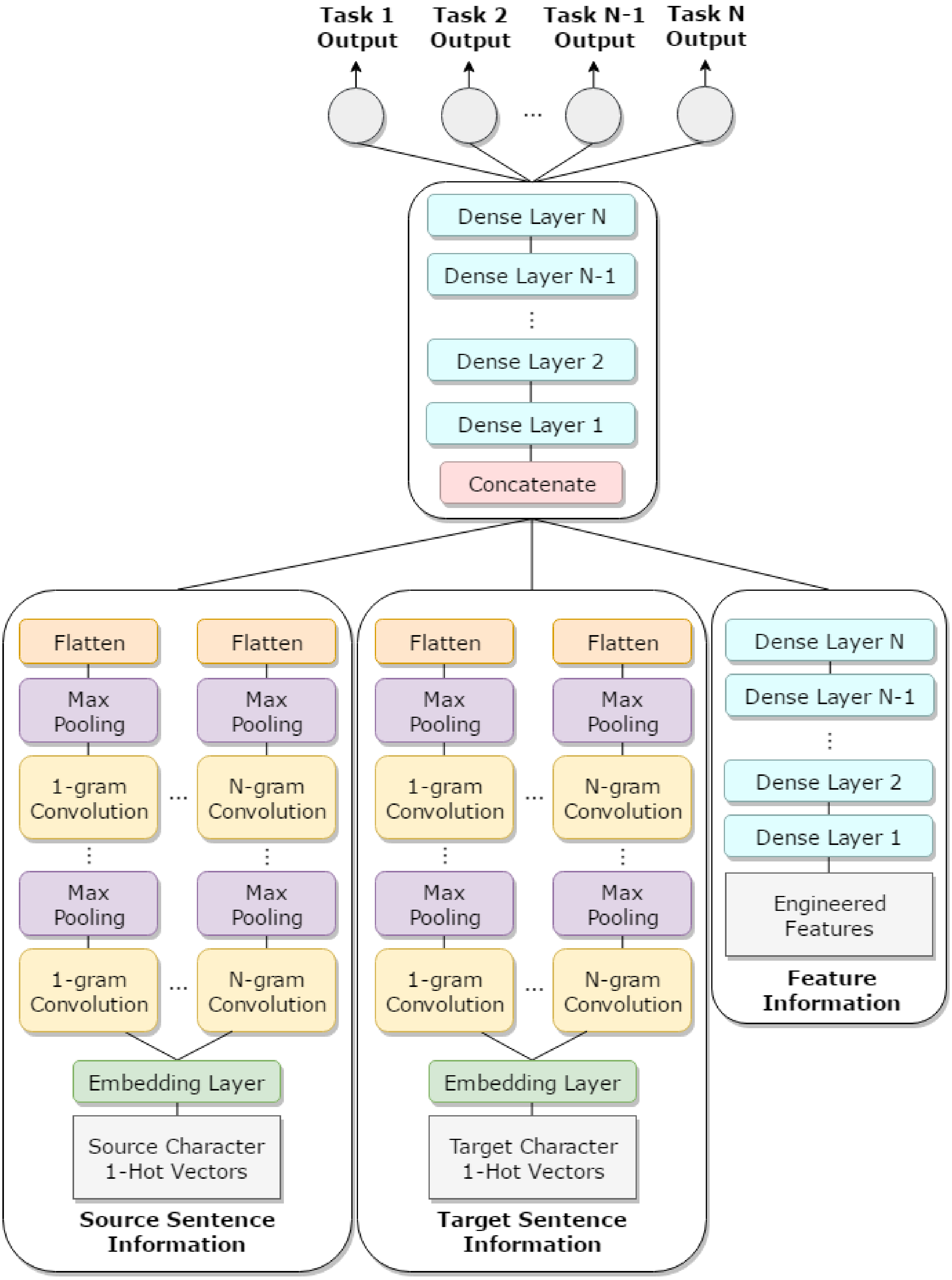

Figure 1: Architecture of the SHEF/CNN+BASE systems

- The number of BAD labels from the word-level dataset; and
- The number of BAD labels from the phrase-level dataset.

Note that the data from the word and phrase-level datasets are used as a mere complement to HTER prediction. It is important to mention that we also tried predicting the full label sequences for word and phrase-level, but the results obtained were not as promising. We train our model until convergence with Stochastic Gradient Descent and Mean Squared Error over all outputs jointly.

5 SHEF/CNN Model for EIA

The model used for the EIA task of WASSA 2017 applies only one convolution stack over the tweet being analysed, given that the task is not characterized by a sentence pair. The window range, convolution depth, as well as feature and final MLP depths are identical to the model used for the WMT 2017 task. We train one model for each emotion targeted in the shared task: anger, fear, joy and sadness.

Since the organizers did not provide a set of baseline features, we produced our own features using the Stanford Sentiment Treebank (Socher et al., 2013), which is composed of 239,232 text segments annotated with respect to their positivity probability i.e. how likely they are to convey a positive emotion. The positivity values range from 0.0 (absolutely negative) to 1.0 (absolutely positive). Using this data, we extract nine features from each tweet:

- Minimum, maximum and average positivity of single words in the tweet;
- Minimum, maximum and average positivity of bigrams in the tweet; and
- Minimum, maximum and average positivity of trigrams in the tweet.

Our multi-task learning setup is composed of two output layers that predict:

- The tweets' emotion intensity; and
- The tweets' positivity value.

We first train our models over the sentiment positivity values from the Stanford Sentiment Treebank until convergence, then train them over the emotion intensity training sets of WASSA 2017 until convergence. The training algorithm and metric used are Stochastic Gradient Descent and Mean Squared Error, respectively.

6 WMT 2017 Results

We evaluate the performance of four variants of the SHEF/CNN model:

- **SHEF/CNN-F:** Uses only the MLP over the engineered features trained over HTER.
- **SHEF/CNN-C:** Uses only the character-level convolution stacks trained over HTER.
- **SHEF/CNN-C+F:** Uses both engineered features and character-level information trained over HTER.
- **SHEF/CNN-C+F+M:** Uses the same architecture of SHEF/CNN-C+F, but the model is trained through multi-task learning over the values listed in Section 4.

Table 2 illustrates the Pearson, Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) scores for the scoring task, and Spearman correlation scores for the ranking task of each language pair. Boldface values represent the best scores obtained across SHEF/CNN models. We also include the results from the official baseline and from the top performing team (POSTECH).

The results reveal that, although we outperform the task baseline for English-German, the SHEF/CNN models do not offer competitive performance to state-of-the-art QE systems that rely on resource-heavy strategies. Nonetheless, some valuable observations can be drawn from the results. Combining engineered features with character-level clues yields a more reliable model than simply using either of them alone, which suggests that character-level clues can be a valuable source of complementary information to engineered features. Our multi-task learning setup did not improve on the results of our model. We hypothesize that the secondary output labels could not offer a significant volume of complementary information to the model.

7 WASSA 2017 Results

Table 3 illustrates the Pearson and Spearman correlation scores for each emotion. We compare the

	English-German				German-English			
	p	MAE	RMSE	r	p	MAE	RMSE	r
POSTECH/MultiLevel	0.714	0.096	0.134	0.710	0.728	0.091	0.133	0.470
POSTECH/SingleLevel	0.686	0.101	0.139	0.690	0.715	0.094	0.136	0.440
Baseline	0.397	0.136	0.175	0.425	0.441	0.128	0.175	0.450
SHEF/CNN-F	0.384	0.176	0.137	0.412	0.092	0.208	0.145	0.034
SHEF/CNN-C	0.374	0.181	0.146	0.393	0.379	0.184	0.148	**0.408**
SHEF/CNN-C+F	**0.416**	**0.174**	**0.135**	0.441	**0.390**	**0.179**	**0.136**	0.382
SHEF/CNN-C+F+M	0.402	0.178	**0.135**	**0.448**	0.350	0.202	0.162	0.380

Table 2: Results for the sentence-level QE task of WMT 2017

	Fear		Joy		Anger		Sadness	
	p	r	p	r	p	r	p	r
Prayas	0.732	0.729	0.732	0.710	0.762	0.743	0.765	0.761
Emkay	0.690	0.690	0.705	0.692	0.726	0.703	0.767	0.764
venkatesh-1729	0.728	0.728	0.678	0.654	0.705	0.684	0.749	0.744
Baseline	0.652	0.635	0.654	0.662	0.639	0.615	0.648	0.651
SHEF/CNN-F	0.166	0.153	0.271	0.313	0.222	0.212	0.241	0.240
SHEF/CNN-C	0.217	0.221	0.328	0.302	0.120	0.142	0.259	0.253
SHEF/CNN-C+F	**0.293**	**0.284**	**0.517**	**0.510**	0.279	0.260	**0.323**	**0.326**
SHEF/CNN-C+F+M	0.109	0.096	0.407	0.392	**0.311**	**0.276**	0.233	0.228

Table 3: Results for the EIA task of WASSA 2017

performance of all SHEF/CNN variants described in the previous sections and also include the official task baseline and the three top performing approaches in the EIA task: the Prayas, Emkay and venkatesh-1729 systems.

The SHEF/CNN models are outperformed by a noticeable margin by strategies that heavily employ engineered features and external resources, such as large databases of emotion intensity labels. Nonetheless, our results reveal the same phenomenon highlighted in our experiments with QE: for all emotions, combining engineered features with character-level information yields better performance scores than using only one of these information sources. This serves as further evidence that character-level convolutions can be effectively used as a complement to engineered features.

Our multi-task learning approach only managed to obtain performance improvements for anger. We believe this is due to fact that the positivity values present in the Stanford Sentiment Treebank, which is used in our multi-task setup, accurately quantify only the degree with which the reviewer is pleased, and hence happy, or displeased, and hence angry. Because the other emotions in the WASSA 2017 task do not commonly permeate the act of writing a product review, the multi-task setup was not able to help the model trained for them.

8 Conclusions

We introduced a text regression model that uses deep convolution neural networks and multi-layer perceptrons to combine the character-level information present in texts with the information from engineered features.

We tested several variants of our model in two text regression shared tasks: the sentence-level Quality Estimation task of WMT 2017 and the Emotion Intensity Analysis task of WASSA 2017. We found that, although our model is not able to outperform classic resource-heavy strategies, combining character-level data with engineered features results in noticeable performance gains for both tasks. We also found that, although multi-task learning can in principle help our model, the setup must be carefully crafted, otherwise it compromises its performance.

We plan to further test with other tasks the hypothesis that character-level convolutions constitute an intuitive way of complementing the performance of typical feature-based text regression models. We will also test more elaborate convolution architectures, such as using stacked LSTMs.

Acknowledgments

This work was supported by the QT21 project (H2020 No. 645452).

References

Eleftherios Avramidis. 2012. Comparative quality estimation: Automatic sentence-level ranking of multiple machine translation outputs. In *Proceedings of 24th COLING*. The COLING 2012 Organizing Committee, pages 115–132.

Ergun Bicici. 2016. Referential translation machines for predicting translation performance. In *Proceedings of the First Conference on Machine Translation*. Association for Computational Linguistics, Berlin, Germany, pages 777–781.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the 1st WMT*. Association for Computational Linguistics, Berlin, Germany, pages 131–198.

Hyun Kim and Jong-Hyeok Lee. 2016. Recurrent neural network based translation quality estimation. In *Proceedings of the First Conference on Machine Translation*. Association for Computational Linguistics, Berlin, Germany, pages 787–792.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2016. Character-aware neural language models. In *Proceedings of the 13th AAAI*. AAAI Press, AAAI'16, pages 2741–2749.

Svetlana Kiritchenko, Saif Mohammad, and Mohammad Salameh. 2016. Semeval-2016 task 7: Determining sentiment intensity of english and arabic phrases. In *Proceedings of the 10th SemEval*. Association for Computational Linguistics, San Diego, California, pages 42–51.

Anna Kozlova, Mariya Shmatova, and Anton Frolov. 2016. Ysda participation in the wmt'16 quality estimation shared task. In *Proceedings of the First Conference on Machine Translation*. Association for Computational Linguistics, Berlin, Germany, pages 793–799.

David Langlois. 2015. Loria system for the wmt15 quality estimation shared task. In *Proceedings of the 10th WMT*. Association for Computational Linguistics, Lisbon, Portugal, pages 323–329.

Saif M. Mohammad and Felipe Bravo-Marquez. 2017. Emotion intensities in tweets. In *Proceedings of the 6th *Sem*. Vancouver, Canada.

Gustavo Paetzold and Lucia Specia. 2016. Simplenets: Quality estimation with resource-light neural networks. In *Proceedings of the First Conference on Machine Translation*. Association for Computational Linguistics, Berlin, Germany, pages 812–818.

Eshrag Refaee and Verena Rieser. 2016. ilab-edinburgh at semeval-2016 task 7: A hybrid approach for determining sentiment intensity of arabic twitter phrases. In *Proceedings of the 10th SemEval*. Association for Computational Linguistics, San Diego, California, pages 474–480.

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

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 EMNLP*. Association for Computational Linguistics, Seattle, Washington, USA, pages 1631–1642.

Marco Turchi, Matteo Negri, and Marcello Federico. 2015. MT quality estimation for computer-assisted translation: Does it really help? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Association for Computational Linguistics, Beijing, China, pages 530–535.

Feixiang Wang, Zhihua Zhang, and Man Lan. 2016. Ecnu at semeval-2016 task 7: An enhanced supervised learning method for lexicon sentiment intensity ranking. In *Proceedings of the 10th SemEval*. Association for Computational Linguistics, San Diego, California, pages 491–496.