Quality Estimation for Automatically Generated Titles of eCommerce Browse Pages

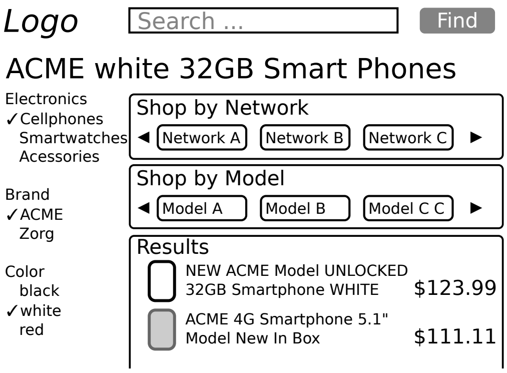
Nicola Ueffing and **José G. C. de Souza** and **Gregor Leusch**
Machine Translation Science Lab
eBay Inc.
Kasernenstraße 25
Aachen, Germany
{nueffing,jgcdesouza,gleusch}@ebay.com

Abstract

At eBay, we are automatically generating a large amount of natural language titles for eCommerce browse pages using machine translation (MT) technology. While automatic approaches can generate millions of titles very fast, they are prone to errors. We therefore develop quality estimation (QE) methods which can automatically detect titles with low quality in order to prevent them from going live. In this paper, we present different approaches: The first one is a Random Forest (RF) model that explores hand-crafted, robust features, which are a mix of established features commonly used in Machine Translation Quality Estimation (MTQE) and new features developed specifically for our task. The second model is based on Siamese Networks (SNs) which embed the metadata input sequence and the generated title in the same space and do not require hand-crafted features at all. We thoroughly evaluate and compare those approaches on in-house data. While the RF models are competitive for scenarios with smaller amounts of training data and somewhat more robust, they are clearly outperformed by the SN models when the amount of training data is larger.

1 Introduction

On eCommerce sites, multiple items can be grouped on a common page called *browse page (BP)*. Each browse page contains an overview of various items which share some, but not necessarily all characteristics. The characteristics can be expressed as slot/value pairs. Figure 1 shows an example of a browse page with a title, with navigation elements leading to related browse pages as well as the individual items listed on this page.



ACME white 32GB Smart Phones

Electronics
✓Cellphones
Smartwatches
Acessories

Brand
✓ACME
Zorg

Color
black
✓white
red

Shop by Network
◀ Network A | Network B | Network C ▶

Shop by Model
◀ Model A | Model B | Model C C ▶

Results

▢	NEW ACME Model UNLOCKED 32GB Smartphone WHITE	$123.99
▢	ACME 4G Smartphone 5.1" Model New In Box	$111.11

Figure 1: Example of a browse page.

The browse pages are linked among each other and can be organized in a hierarchy. This structure allows users to navigate laterally between different browse pages, or to dive deeper and refine their search. The example browse page in Figure 1 shows different white ACME smartphones with capacity 32GB. This page is linked from various browse pages, e.g. those for white ACME Smartphones, for ACME smartphones with 32GB, or for white smartphones with 32GB. It also links to browse pages with a higher number of slots, i.e. refining the set of listed items by additional features like network provider.

Different combinations of characteristics bijectively correspond to different browse pages, and consequently to different *browse page titles*. To show customers which items are grouped on a browse page, we need a human-readable description of the content of that particular page.

Large eCommerce sites can easily have tens of millions of such browse pages in many different languages. Each browse page has one to six slots to be realized. The number of unique slot-value pairs are in the order of hundreds of thousands. All these factors render the task of human creation of browse page titles infeasible. We have therefore developed several strategies to generate these human-readable titles automatically for

Proceedings of NAACL-HLT 2018, pages 52–59
New Orleans, Louisiana, June 1 - 6, 2018. ©2017 Association for Computational Linguistics

any possible browse page (Mathur, Ueffing, and Leusch, 2017). These strategies are based on MT technology and take the slot/value pairs mentioned in Section 1 as input. Examples of such slot/value pairs are the category to which the products belong, and characteristics like brand, color, size, storage capacity, which are dependant on the category. The slot/value pairs for the browse page from Figure 1 are shown in Table 1.

Slot Name	Value
Category	*Cell Phones & Smart Phones*
Brand	*ACME*
Color	*white*
Storage Capacity	*32GB*

Table 1: The underlying metadata for Figure 1.

These metadata are fed into an MT system and translated into natural language. We have developed three different MT-based systems, which are tailored towards different amounts of training data available across languages. These systems are shortly described in Section 4. In this paper, we compare our QE methods on output from different MT systems on English titles.

2 Approach

The automatically generated BP titles are regularly monitored, and quality is assessed by human experts, who label each title with one out of four error severity classes:

- Good: good quality, no issues,
- P3: minor issues, acceptable quality,
- P2: issues which impact the understandability of the title,
- P1: severe issues, like incorrect brand names.

We map these error classes to the two quality classes ‘OK’ and ‘Bad’: ‘Good’ and ‘P3’ represent acceptable title quality (‘OK’), while ‘P2’ and ‘P1’ constitute ‘Bad’ titles. For English browse page titles, we have a large amount of these manually assigned labels available (see Section 3). For automatically predicting the quality of a BP title, we train different machine learning models on these annotated data. In MT(QE) terms, the metadata for a browse page is considered the source language, and the target language is the natural language, English, in our experiments.

2.1 Random Forests

Random Forests are ensemble classifiers that induce several decision trees using some source of randomness to form a diverse set of estimators (Breiman, 2001). There are two sources of randomness: (i) each individual decision tree is trained over a sub-sample of the training data and (ii) when building the tree, the node splitting step is modified to use the best split among splits using random subsets of features. In our experiments, we used the Random Forest (RF) implementation from the Scikit-learn toolkit (Pedregosa et al., 2011).

2.1.1 Features

We trained various RF classifiers, using several different feature types. Some of those features are commonly used in MTQE (Blatz et al., 2004; Specia et al., 2015). Additionally, we developed specific features which are well-suited for browse page title generation. Our features can be grouped into several different classes:

- MTQE: These are common features from quality estimation for MT, such as title length, language model score, or number of unique words in the title;
- Browse-page-specific: These are new features we developed specifically for BP titles, based on the browse page’s metadata, such as the number of slots in the BP, binary indicators for the most frequent slot names, and indicators of incorrect brand names;
- Redundancy: These are features capturing redundancy, e.g. word repetitions, and within-title cosine distance based on word embeddings (Mikolov et al., 2013). We developed those because redundancy emerged as error pattern in the regular monitoring of the titles.

These features explore different sources of information. Some of them are based only on the title itself (e.g. title length and cosine distance between words) and capture the *fluency* of the title. Other features are based only on the browse page’s metadata (e.g. number of slots in the browse page) and capture the *complexity* of the input for title generation. Some features explore both metadata and generated title (e.g. checking for brand names that are not reproduced exactly in the title) and capture the *adequacy* of the generated title given the input data.

Note that all features are black-box features independent of the underlying system which generated the BP titles. This is important for our application because we have different algorithms in production which generate the BP titles. All of them are described in (Mathur, Ueffing, and Leusch, 2017). As we will see in Section 4, the QE model works well for all of them. Another important aspect is that the features can be easily applied to different languages without requiring complex resources.

Hyper-parameter optimization We performed hyper-parameter search of the RF models with random search for 100 iterations and 5-fold cross-validation in each iteration.

2.2 Siamese Networks

As in other areas in machine learning, neural networks have recently gained much attention in MTQE and have contributed to pushing the state of the art of the task (Kim et al., 2017; Martins et al., 2017). One type of neural network that can be used to predict similarity between paired inputs is called Siamese networks. These networks were originally defined by Bromley et al. (1994) as a neural network composed by two symmetric sub-networks that compare two input patterns and outputs a similarity between these inputs. The authors proposed this architecture in the context of signature verification, i.e., estimating how similar two signatures are to each other. SN models have also been applied to face verification (Chopra et al., 2005), metric learning in speech recognition, to extract speaker-specific information (Chen and Salman, 2011) and text similarity (Yih et al., 2011). Grégoire and Langlais (2017) proposes a siamese network architecture to extract parallel sentences out of parallel corpora. This is a pre-print publication found upon completion of the work described here and we plan to have a detailed comparison in future work.

In this work, we build a QE model inspired by work on sentence similarity (Mueller and Thyagarajan, 2016), which uses SN models to learn a similarity metric between paired inputs. The motivation to apply such architecture to QE is that the problem can be seen as a sentence similarity problem but across two "languages": given a sentence in English and its corresponding translation in French, we want to know if the translation is adequate and fluent with respect to the original sentence. In the problem described in this paper, we can reformulate the scenario as follows: given a segment of slot/value pairs representing the metadata and its corresponding title in English, we want to know if the title is adequate and fluent with respect to the metadata input.

2.2.1 Architecture

The SN architecture we are evaluating was built using a specific type of recurrent neural networks (RNNs) to model each segment input. RNNs are models well-suited to deal with variable-length input like natural language sentences. In RNNs, the standard feed-forward neural networks are adapted for sequence data $(x_1, \ldots, x_T)$, where at each time step $t \in 1, ..., T$, a hidden-state vector h_t is updated as $h_t = \sigma(Wx_t + Uh_{t-1})$, where x_t is the input at time t, W is the weight matrix from inputs to the hidden-state vector and U is the weight matrix on the hidden-state vector from the previous time step h_{t-1} and σ is the logistic function defined as $\sigma = (1 + e^{-x})^{-1}$.

Though RNNs can cope with variable-length sequences, the optimization of the weight matrices in RNNs is hard: when the gradients are back-propagated, they decrease to the point of becoming so small that the weights cannot be updated, specially over long input sequences. In order to alleviate this problem, Hochreiter and Schmidhuber (1997) proposed Long Short-Term Memory models (LSTMs), which are able to overcome the vanishing gradients problem by capturing long-range dependencies through its use of memory cell units that can store/access information across long input sequences. For more details on LSTMs we refer the interested reader to Greff et al. (2015).

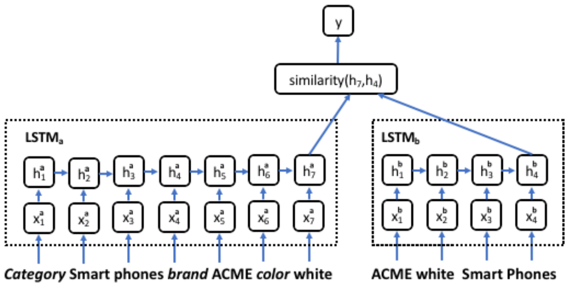


Figure 2: Example of an SN for title quality estimation. The left sequence represents the BP's metadata, the right sequence is the BP title.

The SN architecture we employ in this work is depicted in Figure 2[1]. The architecture consists of two networks, LSTM$_a$ and LSTM$_b$, one for each input sentence (the browse page's metadata on the left and the title on the right). Both LSTMs have tied weights, meaning that both networks have identical transformations in their paths in the experiments presented in this paper.

The architecture is defined in a supervised learning setting, in which each instance is a pair of sentences represented as a sequence of word vectors, $x_1^a, \ldots, x_{T_a}^a$ and $x_1^b, \ldots, x_{T_b}^b$, where $T_a \neq T_b$, and a binary label y that indicates whether the pair is similar or not. The two sequences of word embeddings are the input to their corresponding LSTM, which updates its hidden state at each sequence-index. The sentence is represented by the last hidden state h_T of the LSTM (h_7^a and h_4^b in Figure 2).

The similarity function is pre-defined and is used to compare the LSTM representations and infer their semantic similarity. In this paper, we use the cosine similarity between the final representations of each LSTM, $h_{T_a}^a$ and $h_{T_b}^b$:

$$\mathrm{s}(h_T^a, h_T^b) = \frac{h_T^a \cdot h_T^b}{||h_T^a|| \cdot ||h_T^b||} \tag{1}$$

The cumulative loss function for a training set $X = \{(x_i^a, x_i^b, y_i)\}_{i=1}^N$ is defined following (Neculoiu et al., 2016):

$$\mathcal{L}(X) = \sum_{i=1}^{N} L(x_i^a, x_i^b, y_i) \tag{2}$$

In Equation 2, N is the number of instances in the training set X and L is the instance loss composed of two terms: one for similar pairs (L_+), and one for dissimilar pairs (L_-):

$$\begin{aligned} L(x_i^a, x_i^b, y_i) &= y_i \cdot L_+(x_i^a, x_i^b) \\ &\quad + (1 - y_i) \cdot L_-(x_i^a, x_i^b), \end{aligned} \tag{3}$$

where the loss functions for the similar and dissimilar cases are given by:

$$\begin{aligned} L_+(x_i^a, x_i^b) &= (1 - s)^2 \\ L_-(x_i^a, x_i^b) &= \begin{cases} s^2 & \text{if } s < 0 \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{4}$$

where s stands for the cosine similarity, as defined in Equation 1.

3 Data

3.1 Training Data

For English, we have a large amount of training data, consisting of the browse page's metadata (slot/value pairs and category name), a title generated by the rule-based system (see section 3.2), and manually assigned error severity (see section 2) for this title. Table 2 shows some examples of training data instances.

Category	*MP3 Player Headphones & Earbuds*
Brand	*Sony*
Connector	*3.5mm (1/8in.)*
Features	*Volume Control*
Title	Sony 3.5mm (1/8in.) MP3 Player Headphones & Earbuds with Volume Control
Quality	OK
Category	*Nursery Bedding Sets*
To Fit	*Crib*
Brand	*My Baby Sam*
Title	My Nursery Bedding Sets Sam Baby Crib Shoes
Quality	Bad

Table 2: Examples of metadata, automatically generated title, manually assigned quality class.

Table 3 shows statistics on the training data. When we started working on quality estimation for these titles, we only had the first set of data, labeled *train1*. The other set, *train2*, is much larger and became available later on. We use these two training sets for evaluating the impact of adding more data to model training.

Data set	**# Browse Pages**	**Quality (%)**	
		OK	**Bad**
train1	81,251	65	35
train2	269,409	66	34
artificial P1	29,150	0	100

Table 3: Training data statistics.

The distribution of quality classes is similar across both training sets. The majority of titles is labeled as 'OK', and about one third are labeled as 'Bad'. Since the number of P1 samples in the training data is very low (approx. 1%), we generated 29k additional training samples with P1 issues semi-automatically, in order to increase their representation in the training data, and improve the models' prediction capabilities on this type of errors: We extracted BPs from the training data which contain "brand" slots, modified the curated reference title by misspelling the brand name, and added these modified titles to the training data with label 'Bad'.

[1] Inspired by the figure in (Mueller and Thyagarajan, 2016), and adapted to our use case.

Test Set	# BPs	Rule-based	Hybrid	APE
1	508	60 / 40	–	54 / 46
2	509	62 / 38	63 / 36	71 / 29
1+2	2,543	all combined : 63% OK 37% Bad		

Table 4: Evaluation data statistics. Numbers are % Good / % Bad in the three rightmost columns.

3.2 Evaluation Data

We constantly carry out human evaluation of title quality. From these evaluations, we have two test sets with approximately 500 browse pages each, called *test1* and *test2*. For those browse pages, we have automatically generated titles from three different systems along with manual assessment of title quality. The three different title generation systems are described in detail in (Mathur, Ueffing, and Leusch, 2017). In short, they are:

- a strictly rule-based approach with a manually created grammar. This is especially useful when the amount of human-curated training data is limited.
- a hybrid generation approach which combines rule-based language generation and statistical MT techniques for situations in which monolingual data for the language is available, but human-curated titles are not.
- an Automatic Post-Editing (APE) system which first generates titles with the rule-based approach, and then uses statistical MT techniques for automatically correcting the errors made by the rule-based approach.

See Table 4 for the amount of data and title quality across these different test sets and system outputs. Apart from the APE system, the class distribution is similar for all sets, and also similar to the distribution on the training data. The APE system was significantly improved between these two evaluation rounds, leading to a much higher percentage of 'OK' labels on *test2*. The hybrid system was manually evaluated only on *test2*.

4 Results

We evaluated our QE models in the following scenario: given a browse page's metadata and an automatically generated title, we want to decide whether the title meets the quality standards and should be presented on our website. Evaluation metrics are F1-score per class and total (weighted) F1-score, and Matthew's correlation.

4.1 Model comparison

We first compared QE models obtained using different learning algorithms and trained only on *train1* because model training is faster and we expect the observed trends to be independent of the amount of training data. Table 5 shows the results. The majority baseline (accepting all titles as 'OK') yields fairly low F1-score, because all bad titles are labeled incorrectly. For the RF classifiers, we can see how adding information improves the model. The model based only on MTQE features achieves the worst performance (60 points F1 and correlation 0.24). Our newly developed browse-page-specific features in isolation perform 5 points better both in F1 and in correlation. Combining those two feature groups yields a significant improvement in correlation, though not in total F1. It significantly increases the F1-score for 'Bad' titles, but hurts a bit on the 'OK' titles, which are more frequent in the test data. The redundancy features additionally increase correlation by 1 point absolute.

Model	F1(OK)	F1(Bad)	F1	MC
Majority ('OK')	0.77	0.00	0.48	0.00
Random Forest				
MTQE features	0.61	0.58	0.60	0.24
BP features	0.68	0.59	0.65	0.29
MTQE + BP	0.66	**0.64**	0.65	0.36
MTQE + BP + redun.	0.66	**0.64**	0.65	**0.37**
Siamese Network				
fastText, dim50	**0.80**	0.54	**0.70**	**0.37**
word2vec, dim50	0.79	0.55	**0.70**	**0.37**

Table 5: F1-scores and Matthew's correlation (MC) for different QE models. Training on *train1*, evaluation on *test1+2*. Best results in bold.

Training data	F1(OK)	F1(Bad)	F1	MC
Random Forest				
train1	0.66	**0.64**	0.65	0.37
train1+2	0.76	**0.64**	0.72	0.41
train1+2 + artif.	0.78	**0.64**	0.73	0.43
Siamese Network				
train1	0.79	0.55	0.70	0.37
train1+2 + artif.	**0.82**	**0.64**	**0.75**	**0.48**

Table 6: QE performance for different amounts of training data. Evaluation on *test1+2*. RF with all features. SN with word2vec embeddings. Best results in bold.

We compared SN models with two different pre-trained word embeddings, using either word2vec (Mikolov et al., 2013) or fastText (Bojanowski et al., 2016). As we see in Table 5, their QE performance is almost identical, and we

Model	**F1-score / Matthew's correlation**				
	test1 RB	test2 RB	test1 APE	test2 APE	test2 hybrid
RF trained on *train1*	0.68 / 0.44	0.64 / 0.35	0.62 / 0.29	0.65 / 0.30	0.68 / 0.40
RF trained on all data	0.75 / 0.47	0.74 / 0.46	**0.70 / 0.39**	0.71 / 0.31	0.75 / 0.46
SN trained on *train1*	0.74 / 0.46	0.74 / 0.47	0.60 / 0.24	0.71 / 0.28	0.73 / 0.41
SN trained on all data	**0.79 / 0.57**	**0.81 / 0.59**	0.66 / 0.36	**0.72 / 0.32**	**0.79 / 0.54**

Table 7: QE performance per title generation system. RF with all features. SN with word2vec embeddings. Best results marked in bold. RB is rule-based.

will use the word2vec embeddings going forward. Both SN models significantly outperform the RF models in total F1-score, which increases by 5 points. This stems from much better classification of 'OK' titles, while 'Bad' titles are better recognized by the RF models. Matthew's correlation is at 0.37 both for the best RF and the SN models.

4.2 Impact of training data

After the original experiments described in section 4.1, we obtained a much larger amount of training data. We then trained RF models on the combined sets *train1* and *train2*, with 349k titles. As Table 6 shows, this yields a gain of 7 points in F1-score and 4 points in correlation, caused by improved classification on 'OK' titles. Manual analysis of QE performance showed that it was particularly low on titles with P1 issues. As described in section 3.1, we therefore generated artificial training data for better representing P1 errors in training. Adding these in training further improves the RF model, yielding total F1 of 73 points and correlation of 0.43. The effect of an increased amount of training data is even stronger for the SN models. QE performance increases by 5 points in F1 and 11 points in correlation. This SN trained on all 376k titles is the best QE model according to all metrics.

4.3 System-specific evaluation

We are constantly improving the system for BP title generation and have implemented different approaches. It is therefore important that the QE models work equally well for output from different title generation systems, i.e. they should not be heavily tailored to one specific system.

We evaluated the QE models per evaluation set (*test1* and *test2*) and per title generation system. The QE performance per system output is shown in Table 7, with notable difference in F1-score and Matthew's correlation across the five different sets. The SN models perform best on the titles from the rule-based generation system, i.e. when training and test titles are similar – with F1-scores around 0.8 and Matthew's correlation in the high 50s. The worst classification performance is achieved for the APE titles on *test1*, which is the set with the lowest title quality (see Table 4). This is also the only set on which the RF models outperform the SN models. The RF models were trained with class weights adjusted inversely proportional to class frequencies in the training data, making them more robust w.r.t. the differences between training and test data. The neural network model does not have the same class imbalance treatment, which makes the model biased towards most frequent classes in training data sets in which the imbalance is high (e.g. the rule-based system). In future work, we plan to apply the same balancing to SN training. This setting could potentially improve the SN performance.

5 Conclusion

We developed different methods for automatically assessing the quality of browse page titles. One is a Random Forest classifier which combines well-studied QE features with new features which are specific to the task and explore information from the browse page's metadata. The second approach is a neural network model using a Siamese architecture. The classification performance of the methods was evaluated on in-house data, showing that: (i) Random Forest models are significantly improved by using new task-specific features; (ii) Siamese networks significantly outperform Random Forest models in most settings; (iii) Random Forest models show more robust quality estimation performance on titles where error distribution diverges from what was observed in training; (iv) unsurprisingly, a drastic increase in the amount of training data significantly improves QE performance for both model types; (v) adding artificial training data, which alleviates the imbalanced distribution of error types, improves both types of models. The Siamese architecture presented in this paper could also be employed in the context of machine translation or other language generation tasks in which one needs to estimate the output quality.

As future work, we plan to bring those research and pilot systems into production and gather experience on their use; as well as extending them to multi-class prediction for finer-grained QE, directly predicting the error severity classes (Good, P3, P2, P1). Furthermore, we plan to develop QE methods for languages other than English, where the amount of training data is much smaller.

6 Acknowledgements

We would like to thank our colleague Kashif Shah for the fruitful discussions.

Appendix: Examples

Table 8 shows examples of quality predictions from different QE models. The first block contains titles where both model types correctly predict quality, such as bad titles which have issues with fluency or repetition, or well-formed titles which contain all relevant aspects.

The second block shows examples where none of the models correctly predicts title quality. In the first two examples, the bad quality is caused by omissions of words (“Water” and “Row”), and none of the QE models detects this. This is probably due to the structure of the metadata input, with aspect slot/value pairs like {“Water Type”: “Pond”}, which needs to be realized as “Pond Water” and not just “Pond” in the title – this type of omission is hard to capture for the QE models. Similar observations hold for the slot/value pair {“Row”: “5”} in the next example. In the third and fourth example in the second block, there is a mismatch between the category name in the metadata input and the realization in the title, which might be the cause for the “Bad” QE predictions. Category names are “Sculptures & Carvings Direct from the Artist” and “Barware Glasses & Cups”, respectively, and significant portions of the category names are dropped in both cases.

The third block of the table shows examples where the RF models perform better. The first example is an incorrect brand name (“aden anais”), for which we explicitly designed features in the RF models.

The last block of Table 8 contains titles which the SN models classified correctly, but the RF model did not. The first one is again a case of missing information and resulting disfluency in the title, which seems to be harder to capture for the RF models.

References

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Leo Breiman. 2001. Random forests. *Machine Learning*, 45(1):5–32.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1994. Signature verification using a" siamese" time delay neural network. In *Advances in Neural Information Processing Systems*, pages 737–744.

Ke Chen and Ahmad Salman. 2011. Extracting speaker-specific information with a regularized siamese deep network. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 298–306. Curran Associates, Inc.

Sumit Chopra, Raia Hadsell, and Yann LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 539–546. IEEE.

Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R. Steunebrink, and Jürgen Schmidhuber. 2015. LSTM: A search space odyssey. *CoRR*, abs/1503.04069.

Francis Grégoire and Philippe Langlais. 2017. A deep neural network approach to parallel sentence extraction. *CoRR*, abs/1709.09783.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Lstm can solve hard long time lag problems. In *Advances in neural information processing systems*, pages 473–479.

Hyun Kim, Jong-Hyeok Lee, and Seung-Hoon Na. 2017. Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation. In *Proceedings of the Second Conference on Machine Translation*, pages 562–568, Copenhagen, Denmark. Association for Computational Linguistics.

Title		Quality		
Automatic	**Reference**	**RF**	**SN**	**Ref.**
both correct				
Telefunken Portable AM/FM Radios with Alarm	Telefunken Portable AM/FM Radios with Alarm	OK	OK	OK
Dog Standard Shampoos	Dog Standard Shampoos	OK	OK	OK
Samsung Cell Display: Lens Screens Parts for Verizon	Samsung Cellphone & Smartphone Lens Screens for Verizon	Bad	Bad	Bad
Science Fiction Fiction & Literature Books	Science Fiction Books in German	Bad	Bad	Bad
both incorrect				
Pond Aquarium Filter Media & Accessories	Pond Water Aquarium Filter Media and Accessories	OK	OK	Bad
3 5 Concert Tickets	Row 5 3 Concert Tickets	OK	OK	Bad
Less than 12in. Figurines Direct from the Artist	Less than 12" Direct from the Artist Figurines	Bad	Bad	OK
Whisky Glasses Barware	Barware Whisky Glasses	Bad	Bad	OK
RF better				
aden anais Cribskirts & Dust Ruffles	aden+anais Cribskirts & Dust Ruffles	Bad	OK	Bad
Full Sun Dry H2 (1 to 5°C) Cactus & Succulent Plants	Full Sun Dry H2 (1 to 5°C) Cactus & Succulent Plants	OK	Bad	OK
SN better				
Topshop Mid L32 Jeans for Women	Topshop Mid Rise L32 Jeans for Women	OK	Bad	Bad
Simulation Sony PlayStation 2 Manual Included PAL Video Games	Sony PlayStation 2 Simulation PAL Video Games with Manual	Bad	OK	OK

Table 8: Examples of quality predictions from the RF and SN models, and true label (Ref.)

André F. T. Martins, Fabio Kepler, and Jose Monteiro. 2017. Unbabel's participation in the WMT17 translation quality estimation shared task. In *Proceedings of the Second Conference on Machine Translation*, pages 569–574, Copenhagen, Denmark. Association for Computational Linguistics.

Prashant Mathur, Nicola Ueffing, and Gregor Leusch. 2017. Generating titles for millions of browse pages on an e-commerce site. In *Proceedings of the International Conference on Natural Language Generation*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *AAAI*, pages 2786–2792.

Paul Neculoiu, Maarten Versteegh, and Mihai Rotaru. 2016. Learning text similarity with siamese recurrent networks. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 148–157.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Lucia Specia, Gustavo Henrique Paetzold, and Carolina Scarton. 2015. Multi-level translation quality prediction with QuEst++. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 115–120.

Wen-tau Yih, Kristina Toutanova, John C Platt, and Christopher Meek. 2011. Learning discriminative projections for text similarity measures. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 247–256. Association for Computational Linguistics.