QUality Estimation from ScraTCH (QUETCH): Deep Learning for Word-level Translation Quality Estimation

Julia Kreutzer and **Shigehiko Schamoni**
Computational Linguistics
Heidelberg University
69120 Heidelberg, Germany
{kreutzer,schamoni}@cl.uni-heidelberg.de

Stefan Riezler
Computational Linguistics & IWR
Heidelberg University
69120 Heidelberg, Germany
riezler@cl.uni-heidelberg.de

Abstract

This paper describes the system submitted by the University of Heidelberg to the Shared Task on Word-level Quality Estimation at the 2015 Workshop on Statistical Machine Translation. The submitted system combines a continuous space deep neural network, that learns a bilingual feature representation from scratch, with a linear combination of the manually defined baseline features provided by the task organizers. A combination of these orthogonal information sources shows significant improvements over the combined systems, and produces very competitive F_1-scores for predicting word-level translation quality.

1 Introduction

This paper describes the University of Heidelberg submission to the Shared Task on Word-level Quality Estimation (QE Task 2) at the 2015 Workshop on Statistical Machine Translation (WMT15). The task consists of predicting the word-level quality level ("OK"/"BAD") of English-to-Spanish machine translations, without the use of human references, and without insight into the translation derivations, that is, by treating the Machine Translation (MT) system that produced the translations as a black box.

The task organizers provided training and development data comprising tokenized MT outputs that were automatically annotated for errors as edit operations (replacements, insertions, or deletions) with respect to human post-edits (Snover et al., 2006). Furthermore, a set of 25 baseline features that operate on source and target translation, but do not use features of the SMT pipeline that produced the translations, was provided. Even though the distribution of binary labels is skewed towards "OK" labels, even more so than in the previous QE task at WMT14[1], the most common approach is to treat the problem as a supervised classification task. Furthermore, most approaches rely on manually designed features, including source and target contexts, alignments, and generalizations by linguistic categories (POS, syntactic dependency links, WordNet senses) as reported by Bojar et al. (2014), similar to the 25 feature templates provided by the organizers.

We apply the framework of Collobert et al. (2011) to learn bilingual correspondences "from scratch", i.e. from raw input words. To this aim, a continuous space deep neural network is pre-trained by initializing the lookup-table with distributed word representations (Mikolov et al., 2013b), and fine-tuned for the QE classification task by back-propagating word-level prediction errors using stochastic gradient descent (Rumelhart et al., 1986). Moreover, we train a linear combination of the manually defined baseline features provided by the task organizers. A combination of the orthogonal information based on the continuous space features and the manually chosen baseline features shows significant improvements over the combined systems, and produces very competitive F_1 scores for predicting word-level translation quality.

2 Deep Learning for Quality Estimation

Continuous space neural network models are credited with the advantage of superior modeling power by replacing discrete units such as words or n-grams by vectors in continuous space, allowing similar words to have similar representations, and avoiding data sparsity issues. These advantages have been demonstrated experimentally by showcasing meaningful structure in vector space

[1] A factor of 4.22 on WMT15 train, and 4.21 on WMT15 dev, as opposed to 1.84 for WMT14 train and 1.81 for WMT14 test for the same language pair.

Proceedings of the Tenth Workshop on Statistical Machine Translation, pages 316–322,
Lisboa, Portugal, 17-18 September 2015.

representations (Mikolov et al. (2013c), Pennington et al. (2014) *inter alia*), or by producing state-of-the-art performance in applications such as language modeling (Bengio et al. (2003), Mikolov et al. (2010), *inter alia*) or statistical machine translation (Kalchbrenner and Blunsom (2013), Bahdanau et al. (2015), *inter alia*). The property that makes these models most attractive for various applications is the ability to learn continuous space representations "from scratch"(Collobert et al., 2011), and to infuse the representation with non-linearity. The deep layers of the neural network capture these representations – even a single hidden layer is sufficient (Hornik et al., 1989).

We present an approach to address the challenges of word-level translation quality estimation by learning these continuous space bilingual representations instead of relying on manual feature engineering. While the neural network architecture presented by Collobert et al. (2011) is limited to monolingual word-labeling tasks, we extend it to the bilingual context of QE. The multilayer feedforward neural network is pre-trained in an unsupervised fashion by initializing the lookup-table with `word2vec` representations (Mikolov et al., 2013b). This is not only an effective way of guiding the learning towards minima that still allow good generalization in non-convex optimization (Bengio, 2009; Erhan et al., 2010), but it also proves to yield considerably better results in our application. In addition, we train a linear combination of the manually defined baseline features provided by the task organizers. We combine these orthogonal information sources and find significant improvements over each individual system.

3 QUETCH

Our *QUality Estimation from scraTCH* (QUETCH) system is based on a neural network architecture built with `Theano` (Bergstra et al., 2010). We design a multilayer perceptron (MLP) architecture with one hidden layer, non-linear `tanh` activation functions and a lookup-table layer as proposed by Collobert et al. (2011). The lookup-table has the function of mapping word to continuous vectors and is updated during training. Figure 1 illustrates the connections between the input, hidden lookup-table and linear layer, and the output.

Training is done by optimizing the log-likelihood of the model given the training data via back-propagation and stochastic gradient descent (Rumelhart et al., 1986). Trainable parameters are the bias vectors ($\mathbf{b_1}$,$\mathbf{b_2}$) and weight matrices (W_1,W_2) of the linear layers and the matrix $M \in \mathbb{R}^{d_{wrd} \times |V|}$ that represents the lookup-table. Tunable hyper-parameters are the number of units of the hidden linear layer, the lookup-table dimensionality d_{wrd} and the learning rate. The number of output units is set to two, since the QE task 2 requires binary classification. The softmax over the activation of these output units is interpreted as score for the two classes.

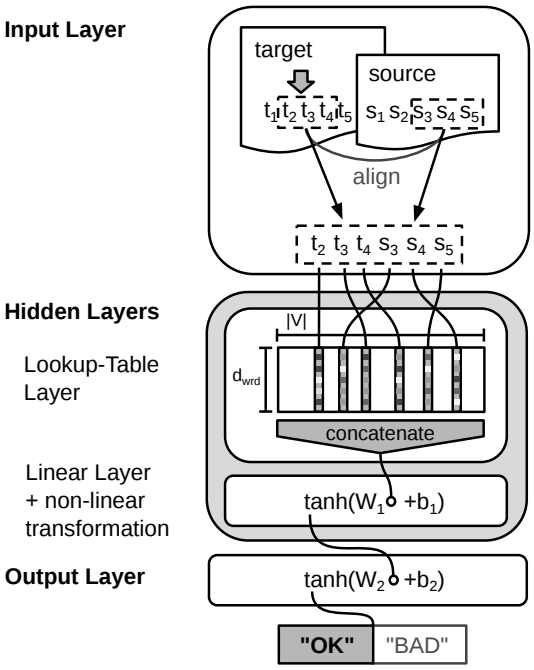

Figure 1: Neural network architecture for predicting word-level translation quality given aligned source and target sentences. The lookup-table matrix M contains d_{wrd}-dimensional vectors for each word in the vocabulary *V*. In this example, the context window sizes $|win_{src}|$ and $|win_{tgt}|$ are set to three and the target word t_3 is classified "OK".

3.1 Bilingual Representation Learning

Given a target word, we consider bilingual context information: From the target sentence we extract a fixed-size word window win_{tgt} centered at the target word. From the aligned source sentence we extract a fixed-size word window win_{src} centered at a position that is either estimated heuristically or via word alignments. Concatenating target and source windows, we obtain a bilingual context vector for a given target word. This context vector is the input for the lookup-table layer, which maps

each context word to a d_{wrd}-dimensional vector[2]. All lookup-table output vectors are concatenated to form the input to the MLP hidden layer. Since the lookup-table representations of words are updated during training, QUETCH learns representations of words in bilingual contexts that are optimized for QE.

3.2 Unsupervised Pre-training

Usually, the parameters of a neural network are initialized with zeros or random numbers, i.e. no a-priori knowledge is captured in the network. However, the learning process can benefit from knowledge that is encoded into the architecture prior to training (Saxe et al., 2011). In case of QE, we want the model to know what well-written source and target sentences look like – before actually seeing translations. `word2vec` (Mikolov et al., 2013b; Mikolov et al., 2013a) offers efficient methods to pre-train word representations in an unsupervised fashion such that they reflect word similarities and relations. Initializing the lookup-table with pre-trained `word2vec` vectors allows us to incorporate prior linguistic knowledge about source and target language into QUETCH. During the learning process, these representations are further optimized for QE and the vocabulary encountered during training.

4 Baseline Features and System Combination

In contrast to word-based quality estimation tasks from previous years, this year's data additionally provides a number of baseline features. A straightforward approach would be to integrate the baseline features in the deep learning system on the same level as word-features and train lookup-tables for each feature class (Collobert et al., 2011). While this certainly works for word-similar features like POS-tags, this is not suitable for continuous numerical features. Preliminary tests of extending QUETCH with a lookup-table for POS-tags did not result in better F1 scores. Also, training took considerably longer, because of (1) the additional lookup-table to train and (2) the larger dimensionality of the vector representing a target word with its context. If we added all 25 features for each target word in the context window, the input to the first linear layer would grow by $25 * |win_{tgt}| * d_{wrd}$ dimensions.

For these reasons, we decided to design a system combination that treats the QUETCH system and the baseline features individually and independently. For many complex applications, system combination has proven to be effective strategy to boost performance. In machine translation tasks, Heafield and Lavie (2011) and Karakos et al. (2008), *inter alia*, increased overall performance by cleverly combining the outputs of several MT systems. In cross-lingual information retrieval, Schamoni and Riezler (2015) empirically showed that it is more beneficial to combine systems that are most dissimilar than those that have highest single scores.

Our approach is to train separate systems, one based on the deep learning approach described in Section 3, and one based solely on the baseline features provided for the shared task. In a final step, we combine both systems together with binarized versions of selected baseline features. From this modular combination of both systems, we can furthermore gain knowledge about their individual contribution to the combined system which will help to understand their usefulness for the QE task.

4.1 Baseline Features System

To obtain a system for baseline features that is most complementary to QUETCH, we used the Vowpal Wabbit (VW) toolkit (Goel et al., 2008) to train a linear classifier, i.e. a single-layer perceptron. We built new features by "pairing" baseline features, thus we quadratically expand the feature space and learn a weight for each possible pair.

Assuming two feature vectors $\mathbf{p} \in \{0,1\}^P$ and $\mathbf{q} \in \{0,1\}^Q$ of sizes P and Q where the n^{th} dimension indicates the occurrence of the n^{th} feature, we define our linear model as

$$f(\mathbf{p},\mathbf{q}) = \mathbf{p}^\top W \mathbf{q} = \sum_{i=1}^{P}\sum_{j=1}^{Q} p_i W_{ij} q_j,$$

where $W \in \mathbb{R}^{P\times Q}$ encodes a feature matrix (Bai et al., 2010; Schamoni et al., 2014). The value of $f(\cdot,\cdot)$ is the prediction of the classifier given a target vector $\mathbf{p}$ and a vector of related features $\mathbf{q}$.

To address the problem of data sparsity, we reduced the number of possible feature pairs by restricting the feature expansion to two groups: (1) *target words* are combined with *target context*

[2] All words are indexed within a vocabulary V. The vocabulary contains the entire training, development and test data of the QE task and is realized as a `gensim` dictionary (Řehůřek and Sojka, 2010).

words and *source aligned words*, and (2) *target POS tags* are combined with *source aligned POS tags*. In total, we observed 3.5M different features during training of the VW model.

4.2 System Combination

For the final system combination, we reused the VW toolkit. The combined systems comprises 82 features: the QUETCH-score, the VW-score, and the remaining 80 features are binary features derived from the baseline feature set. The QUETCH-score is the system's prediction combined with its likelihood, for VW we directly utilize the raw predictions with clipping at ± 1. Binarized features were inserted to enrich the classifier with additional non-linearity. They consist of (1) the binary features from the baseline feature set, and (2) binned versions of the numerical features from the same set. For small groups of discrete values we assigned a binary feature to each possible value, for larger groups and real-valued features we heuristically defined intervals ("bins") containing roughly the same number of instances. The integration of the single components for the system combination is illustrated in Figure 2.

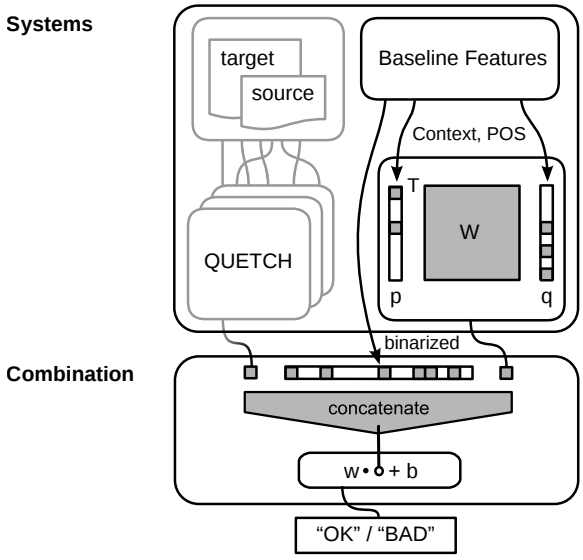

Figure 2: Architecture of the QUETCHPLUS system combination.

5 Experiments

5.1 WMT14

We first ran experiments on the WMT14 task 2 data to compare QUETCH's performance with the WMT14 submissions. With outlook to this year's task we considered only the binary classification task where words are labeled either "BAD" or "OK".

In contrast to the WMT15 data, the WMT14's data covers not only English to Spanish translations (en-es) but also German to English (de-en) and vice versa. Since the plain QUETCH system does not rely on language-specific features, we simply use the same deep learning architecture for all of these language pairs.

QUETCH is trained on the WMT14 training set, with a source and target window size of 3, a lookup-table dimensionality of 10, 300 hidden units, and a constant learning rate of 0.001. Test and training data were lowercased. The alignments used for positioning the target window as described in Section 3.1 were created with `fast_align` from the `cdec` toolkit (Dyer et al., 2010). The collection of corpora provided with WMT13's translation task[3] is utilized as source for unsupervised pre-training: Europarl v7 (Koehn, 2005), Common Crawl corpus, and News Commentary. Note that we did not use these corpora because of their parallel structure, but because they are large, multilingual, and are commonly used in WMT submissions.

Following the WMT14 evaluation (Bojar et al., 2014), we report on accuracy and BAD F_1-score, the latter being the task's primary evaluation metric. The WMT14 baselines trivially predict either only BAD or only OK labels. Table 1 presents the best F_1-scores during training and the according accuracies for QUETCH under different configurations.

The plain QUETCH system yields an acceptable accuracy, but the BAD F_1-scores are not competitive. Adding alignment information further improves the accuracy for all language pairs but de-en. It improves the F_1-score only for es-en and en-de, which indicates that the model is still prone to local optima. It is in fact pre-training that boosts the BAD F_1-score – this initial positioning in the parameter space appears to have a larger impact on the training outcome than the introduction

[3] `http://www.statmt.org/wmt13/translation-task.html`

	configuration	**BAD F_1**	**Accuracy**
en-es	(v)	0.4378	0.5087
	(a)	0.4164	0.5107
	(p)	0.5206	0.4026
	(a), (p)	**0.5228**	0.4196
es-en	(v)	0.2197	0.7604
	(a)	0.2470	0.7749
	(p)	0.3203	0.8051
	(a), (p)	**0.3396**	0.8076
en-de	(v)	0.3743	0.6090
	(a)	0.4197	0.6381
	(p)	0.4684	0.6060
	(a), (p)	**0.4863**	0.6271
de-en	(v)	0.2482	0.7001
	(a)	0.2426	0.6837
	(p)	0.3734	0.6657
	(a), (p)	**0.3791**	0.6792

Table 1: QUETCH results on WMT14 task 2 test data under different configurations: (v)anilla system, (p)retraining of word embeddings, (a)lignments from an SMT system.

of translation knowledge via alignments. However, we can achieve further improvement when combining both pre-training and alignments. As a result, QUETCH outperforms the official winning systems of the WMT14 QE task (see Table 2) and the trivial baselines for all language pairs. The fact that the overall tendencies are consistent across languages proves that QUETCH is capable of language-independent quality estimation.

	submission	**BAD F_1**	**Acc.**
en-es	FBK-UPV-UEDIN/RNN	0.4873	0.6162
es-en	RTM-DCU/RTU-GLMd	0.2914	0.8298
en-de	RTM-DCU/RTU-GLM	0.4530	0.7297
de-en	RTM-DCU/RTU-GLM	0.2613	0.7614

Table 2: Winning submissions of the WMT14 Quality Estimation Task 2 (Bojar et al., 2014).

5.2 WMT15

With the insights from the experiments on the WMT14 data we proceed to the experiments on the WMT15 en-es data. We introduce a weight *w* for BAD training samples, such that QUETCH is trained on each BAD sample *w* times. In this way, we easily counterbalance the skewed distribution of labels, without modifying the classifier's loss function. Also, we utilize the larger and non-parallel Wikicorpus (Reese et al., 2010) in English and Spanish for pre-training. As described in Section 1, 25 baseline features are supplied with training, development and test data. This allows us to evaluate the approach for system combination introduced in Section 4.

During training of the VW-system, we experimented with various loss functions (hinge, squared, logistic) and found the model trained on squared loss to return the highest accuracy. Unwanted collisions in VW's hashed weight vector were reduced by increasing the size of the hash to 28 bits. To prevent the model from degenerating towards OK-labels, we utilized VW's option to set the weight for each training instance individually and increased the weights of the BAD-labeled instances to 4.0.

The VW-system and the system combination were trained in a 10-fold manner, i.e. the VW-system was trained on 9 folds and the weights for system combination were tuned on the 10^{th} fold of the training data. The final weights of the model for evaluation were averaged among all 10 folds.

Table 3 presents the results on the WMT15 data for both QUETCH, the baseline feature VW model, and the system combination referred to as QUETCH+. The QUETCH results were produced under the same parameter conditions as in the WMT14 experiments, and the newly introduced w is set to 2 for the submitted and the combined model, and 5 for another model that was explicitely designed for a high BAD F_1-score.

	configuration	**BAD F_1**	**Accuracy**
QUETCH	(v)	0.2535	0.7104
	(a)	0.2628	0.7099
	(p)	0.2535	0.7668
	(a), (p)	0.2793	0.7716
	†(a), (p), (w)	0.3527	0.7508
	(a), (p), (w)	0.3876	0.6031
	‡(a), (p), (w)	0.2985	0.7888
	‡VW	0.4084	0.7335
	†QUETCH+	**0.4305**	0.6977

Table 3: QUETCH results on en-es WMT15 task 2 test data under different configuration setting: (v)anilla model vs. models using (p)re-training, (a)lignments from an SMT-System, and (w)eighting of the BAD-instances. Submitted systems are preceded by †, components of the final QUETCH+ system are marked with ‡.

Although proceeding in the same manner as in the WMT14 experiments, we see slightly different tendencies here: Adding alignments has a positive

effect on the BAD F_1-score, whereas pre-training improves mainly the accuracy. Still, the combination of both yields both a high BAD F_1-score and a high accuracy, which indicates that QUETCH succeeds in integrating both contributions in a complementary way. Adding BAD weights furthermore improves the BAD F_1-score, yet losing some accuracy. Further increasing the weight up to 5 strengthens this effect, such that we obtain a model with very high BAD F_1-score, but rather low accuracy.

The stand-alone VW model yields generally higher BAD F_1-score, but does not reach QUETCH's accuracy. To enhance the orthogonality of the two models for combination, we select a QUETCH model with extremely high accuracy for the system combination[4]. Interestingly, the system combination appears to profit from both models, resulting in the overall best BAD F_1-score. The resulting VW weights of 1.188 for QUETCH and 0.951 for VW underline each system's contribution. The next most important features for the combination were *pseudo_reference* and *is_proper_noun* with weights of 0.2208 and 0.1557, respectively.

system	**BAD F_1**	**OK F_1**	**All F_1**
baseline	0.1678	0.8893	0.7531
QUETCH	0.3527	0.8456	0.7526
QUETCH+	0.4305	0.7942	0.7256
UAlacant/OnLine-SBI-Baseline	0.4312	0.7807	0.7147

Table 4: Official test results on WMT15 task 2 for word level translation quality. The All F_1-score is the weighted average of BAD F_1 and OK F_1, where the weights are determined by the frequency of the classes in the test data. The UAlacant/OnLine-SBI-Baseline and the QUETCH+ predictions show no significant difference at p=0.05 and are both announced official winners.

Table 4 shows the final test results on the WMT15 task 2 for the main evaluation metric of F_1 for predicting BAD word level translation quality, the F_1 for predicting OK translations and their weighted average. Both submitted systems, QUETCH and QUETCH+, yield considerable improvements over the baseline. The QUETCH+ system that combines the neural network with the linearly weighted baseline features is nominally outperformed by one other system by 0.07% BAD F_1 points, but their difference is not significant at p=0.05.

[4]We observe that the training process first produces high BAD F_1-score models, then further improves the accuracy whilst slowly decreasing the BAD F_1-score. This is due to the fact that we do not optimize on the BAD F_1-score directly, but the log-likelihood of the data, which is skewed towards the OK label. This behavior allows us to select models with individual trade-offs between BAD F_1-score and accuracy at different stages of training.

6 Conclusion

We successfully applied a continuous space deep neural network to the task of quality estimation. With QUETCH we built a language-independent neural network architecture that learns representations for words in bilingual contexts from scratch. Furthermore we showed how this architecture benefits from unsupervised pre-training on large corpora. Winning the WMT15 QE task we found evidence that the combination of such a continuous space deep model with a discrete shallow model benefits from their orthogonality and produces very competitive F_1-scores for quality estimation. Further work will address the transfer to sentence-based predictions and the introduction of convolution and recurrence into the neural network architecture.

Acknowledgments.

This research was supported in part by DFG grant RI-2221/1-2 "Weakly Supervised Learning of Cross-Lingual Systems".

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, CA.

Bing Bai, Jason Weston, David Grangier, Ronan Collobert, Kunihiko Sadamasa, Yanjun Qi, Olivier Chapelle, and Kilian Weinberger. 2010. Learning to rank with (a lot of) word features. *Information Retrieval Journal*, 13(3):291–314.

Yoshua Bengio, Rejean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Joural of Machine Learning Research*, 3:1137–1155.

Yoshua Bengio. 2009. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127.

James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. 2010. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, Austin, TX.

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, et al. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation (WMT)*, Baltimore, MD.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Chris Dyer, Adam Lopez, Juri Ganitkevitch, Jonathan Weese, Ferhan Ture, Phil Blunsom, Hendra Setiawan, Vladimir Eidelman, and Philip Resnik. 2010. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Proceedings of the ACL 2010 System Demonstrations*, Uppsala, Sweden.

Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. 2010. Why does unsupervised pre-training help deep learning? *The Journal of Machine Learning Research*, 11:625–660.

Sharad Goel, John Langford, and Alexander L. Strehl. 2008. Predictive indexing for fast search. In *Advances in Neural Information Processing Systems (NIPS)*, Vancouver, Canada.

Kenneth Heafield and Alon Lavie. 2011. CMU system combination in WMT 2011. In *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation*, Edinburgh, Scotland, United Kingdom.

Kurt Hornik, Maxwell Stinchcombe, and Halber White. 1989. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Seattle, WA.

Damianos Karakos, Jason Eisner, Sanjeev Khudanpur, and Markus Dreyer. 2008. Machine translation system combination using itg-based alignments. In *Proceedings of ACL-08: HLT, Short Papers*, Columbus, Ohio.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X*, Phuket, Thailand.

Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proceedings of Interspeech*, Makuhari, Chiba, Japan.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at ICLR*, Scottsdale, AZ.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems (NIPS)*, Lake Tahoe, CA.

Tomas Mikolov, Wen tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Atlanta, Georgia.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar.

Samuel Reese, Gemma Boleda, Montse Cuadros, Llus Padr, and German Rigau. 2010. Wikicorpus: A word-sense disambiguated multilingual wikipedia corpus. In *Proceedings of 7th Language Resources and Evaluation Conference (LREC)*, La Valleta, Malta.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, Valletta, Malta.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. Learning representations by back-propagating errors. *Nature*, 323.

Andrew Saxe, Pang W Koh, Zhenghao Chen, Maneesh Bhand, Bipin Suresh, and Andrew Y Ng. 2011. On random weights and unsupervised feature learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, Bellevue, WA.

Shigehiko Schamoni and Stefan Riezler. 2015. Combining Orthogonal Information in Large-Scale Cross-Language Information Retrieval. In *Proceedings of the 38th Annual ACM SIGIR Conference (SIGIR)*, Santiago, Chile.

Shigehiko Schamoni, Felix Hieber, Artem Sokolov, and Stefan Riezler. 2014. Learning translational and knowledge-based similarities from relevance rankings for cross-language retrieval. In *Proceedings of the Association of Computational Linguistics (ACL)*, Baltimore, MD.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas (AMTA)*, Cambridge, MA.