Machine Translation Evaluation using Recurrent Neural Networks

Rohit Gupta[1], **Constantin Orăsan**[1], **Josef van Genabith**[2]
[1]Research Group in Computational Linguistics, University of Wolverhampton, UK
[2]Saarland University and German Research Center for Artificial Intelligence (DFKI), Germany
{r.gupta, c.orasan}@wlv.ac.uk
josef.van_genabith@dfki.de

Abstract

This paper presents our metric (UoW-LSTM) submitted in the WMT-15 metrics task. Many state-of-the-art Machine Translation (MT) evaluation metrics are complex, involve extensive external resources (e.g. for paraphrasing) and require tuning to achieve the best results. We use a metric based on dense vector spaces and Long Short Term Memory (LSTM) networks, which are types of Recurrent Neural Networks (RNNs). For WMT-15 our new metric is the best performing metric overall according to Spearman and Pearson (Pre-TrueSkill) and second best according to Pearson (TrueSkill) system level correlation.

1 Introduction

Deep learning approaches have turned out to be successful in many NLP applications such as paraphrasing (Mikolov et al., 2013b; Socher et al., 2011), sentiment analysis (Socher et al., 2013b), parsing (Socher et al., 2013a) and machine translation (Mikolov et al., 2013a). While dense vector space representations such as those obtained through Deep Neural Networks (DNNs) or Recurrent Neural Networks (RNNs) are able to capture semantic similarity for words (Mikolov et al., 2013b), segments (Socher et al., 2011) and documents (Le and Mikolov, 2014) naturally, traditional measures can only achieve this using resources like WordNet and paraphrase databases.

This paper presents a novel, efficient and compact MT evaluation measure based on RNNs. Our metric (Gupta et al., 2015) is simple in the sense that it does not require much machinery and resources apart from the dense word vectors. This cannot be said of most of the state-of-the-art MT evaluation metrics, which tend to be complex and require extensive feature engineering. Our metric is based on RNNs and particularly on Tree Long Short Term Memory (Tree-LSTM) networks (Tai et al., 2015). LSTM is a sequence learning technique which uses a memory cell to preserve a state over a long period of time. This enables distributed representations of sentences using distributed representations of words. Tree-LSTM (Tai et al., 2015) is a recent approach, which is an extension of the simple LSTM framework (Hochreiter and Schmidhuber, 1997; Zaremba and Sutskever, 2014).

2 Related Work

Many metrics have been proposed for MT evaluation. Earlier popular metrics are based on n-gram counts (e.g. BLEU (Papineni et al., 2002) and NIST (Doddington, 2002)) or word error rate. Other popular metrics like METEOR (Denkowski and Lavie, 2014) and TERp (Snover et al., 2008) also use external resources like WordNet and paraphrase databases. However, system-level correlation with human judgements for these metrics remains below 0.90 Pearson correlation coefficient (as per WMT-14 results, BLEU-0.888, NIST-0.867, METEOR-0.829, TER-0.826, WER-0.821).

Recent best performing metrics in the WMT-14 metric shared task (Machácek and Bojar, 2014) used a combination of different metrics. The top performing system DiskoTK-Party-Tuned (Joty et al., 2014) in the WMT-14 task uses five different discourse metrics and twelve different metrics from the ASIYA MT evaluation toolkit (Giménez and Màrquez, 2010). The metric computes the number of common sub-trees between a reference and a translation using a convolution tree kernel (Collins and Duffy, 2001). The basic version of the metric does not perform well but in combination with the other 12 metrics from the ASIYA toolkit obtained the best results for the WMT-14

Proceedings of the Tenth Workshop on Statistical Machine Translation, pages 380–384,
Lisboa, Portugal, 17-18 September 2015.

metric shared task. Another top performing metric LAYERED (Gautam and Bhattacharyya, 2014), uses linear interpolation of different metrics. LAYERED uses BLEU and TER to capture lexical similarity, Hamming score and Kendall Tau Distance (Birch and Osborne, 2011) to identify syntactic similarity, and dependency parsing (De Marneffe et al., 2006) and the Universal Networking Language[1] for semantic similarity.

For our participation in the WMT-15 task, we used our metric ReVal (Gupta et al., 2015). ReVal metric is based on dense vector spaces and Tree Long Short Term Memory networks. This metric achieved state of the art results for the WMT-14 dataset. The metric including training data is available at https://github.com/rohitguptacs/ReVal.

3 LSTMs and Tree-LSTMs

Recurrent Neural Networks allow processing of arbitrary length sequences, but early RNNs had the problem of vanishing and exploding gradients (Bengio et al., 1994). RNNs with LSTM (Hochreiter and Schmidhuber, 1997) tackle this problem by introducing a memory cell composed of a unit called constant error carousel (CEC) with multiplicative input and output gate units. Input gates protect against irrelevant inputs and output gates against current irrelevant memory contents. This architecture is capable of capturing important pieces of information seen in a bigger context. Tree-LSTM is an extension of simple LSTM. A typical LSTM processes the information sequentially whereas Tree-LSTM architectures enable sentence representation through a syntactic structure. Equation (1) represents the composition of a hidden state vector for an LSTM architecture. For a simple LSTM, c_t represents the memory cell and o_t the output gate at time step t in a sequence. For Tree-LSTM, c_t represents the memory cell and o_t represents the output gate corresponding to node t in a tree. The structural processing of Tree-LSTM makes it more favourable for representing sentences. For example, dependency tree structure captures syntactic features and model parameters capture the importance of words (content vs. function words).

$$h_t = o_t \odot \tanh c_t \tag{1}$$

[1] http://www.undl.org/unlsys/unl/unl2005/UW.htm

Figure 1 shows simple LSTM and Tree-LSTM architectures.

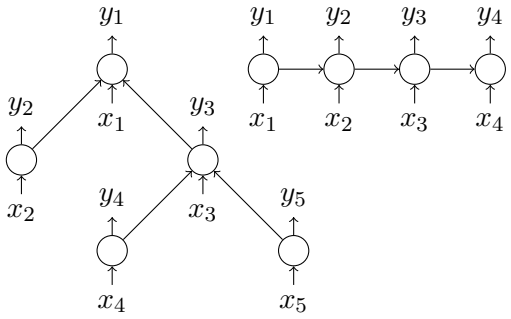

Figure 1: Tree-LSTM (left) and simple LSTM (right)

4 Evaluation Metric

We used the ReVal (Gupta et al., 2015) metric for this task. This metric represents both the reference (h_{ref}) and the translation (h_{tra}) using a dependency Tree-LSTM (Tai et al., 2015) and predicts the similarity score $\hat{y}$ based on a neural network which considers both distance and angle between h_{ref} and h_{tra}:

$$\begin{aligned} h_\times &= h_{ref} \odot h_{tra} \\ h_+ &= |h_{ref} - h_{tra}| \\ h_s &= \sigma\left(W^{(\times)} h_\times + W^{(+)} h_+ + b^{(h)}\right) \\ \hat{p}_\theta &= \text{softmax}\left(W^{(p)} h_s + b^{(p)}\right) \\ \hat{y} &= r^T \hat{p}_\theta \end{aligned} \tag{2}$$

where, σ is a sigmoid function, $\hat{p}_\theta$ is the estimated probability distribution vector and $r^T = [1\ 2 ... K]$. The cost function $J(\theta)$ is defined over probability distributions p and $\hat{p}_\theta$ using regularised Kullback-Leibler (KL) divergence.

$$J(\theta) = \frac{1}{n} \sum_{i=1}^{n} \text{KL}\left(p^{(i)} || \hat{p}_\theta^{(i)}\right) + \frac{\lambda}{2} ||\theta||_2^2 \tag{3}$$

In Equation 3, i represents the index of each training pair, n is the number of training pairs and p is the sparse target distribution such that $y = r^T p$ is defined as follows:

$$p_j = \begin{cases} y - \lfloor y \rfloor, & j = \lfloor y \rfloor + 1 \\ \lfloor y \rfloor - y + 1, & j = \lfloor y \rfloor \\ 0 & \text{otherwise} \end{cases}$$

Metric	fr-en	fi-en	de-en	cs-en	ru-en	PAvg	Pre-TrueSkills Avg	SAvg
UoW-LSTM	**.997** ± .003	**.976** ± .008	.960 ± .010	.983 ± .003	.963 ± .009	.976 ± .007	**.976** ± .011	**.916** ± .038
DPMFCOMB	.995 ± .004	.958 ± .011	.973 ± .009	.991 ± .002	.974 ± .008	**.978** ± .007	.970 ± .012	.882 ± .041
BEER_TREEPEL	.981 ± .008	.971 ± .010	.952 ± .012	.992 ± .002	**.981** ± .008	.975 ± .008	.962 ± .014	.861 ± .051
DPMF	**.997** ± .003	.951 ± .011	.960 ± .010	.984 ± .003	.973 ± .008	.973 ± .007	.965 ± .012	.893 ± .035
UPF-COBALT	.987 ± .006	.962 ± .010	**.981** ± .007	**.993** ± .002	.929 ± .014	.971 ± .008	.970 ± .012	.888 ± .040
BLEU	.975 ± .009	.929 ± .014	.865 ± .020	.957 ± .006	.851 ± .022	.915 ± .014	.889 ± .021	.796 ± .052
TER	.979 ± .008	.872 ± .019	.890 ± .018	.907 ± .008	.907 ± .017	.911 ± .014	.884 ± .022	.768 ± .054
WER	.977 ± .009	.853 ± .020	.884 ± .018	.888 ± .018	.895 ± .018	.899 ± .015	.871 ± .023	.747 ± .057

Table 1: Results WMT-15 Evaluation: System-Level Correlations

Test	fr-en	fi-en	de-en	cs-en	ru-en	Average
UoW-LSTM	.332 ± .011	.376 ± .012	.375 ± .011	.385 ± .008	.356 ± .010	.365 ± .011
DPMFCOMB	.395 ± .012	**.445** ± .012	**.482** ± .009	**.495** ± .007	**.418** ± .013	**.447** ± .011
BEER_TREEPEL	.389 ± .014	.438 ± .010	.447 ± .008	.471 ± .007	.403 ± .014	.429 ± .011
RATATOUILLE	**.398** ± .010	.421 ± .011	.441 ± .010	.472 ± .007	.393 ± .013	.425 ± .010
UPF-COBALT	.386 ± .012	.437 ± .013	.427 ± .011	.457 ± .007	.402 ± .013	.422 ± .011
SENTBLEU	.358 ± .013	.308 ± .012	.360 ± .011	.391 ± .006	.329 ± .011	.349 ± .011

Table 2: Results WMT-15 Evaluation: Segment-Level Correlations

for $1 \leq j \leq K$. Where, $y \in [1, K]$ is the similarity score of a training pair. For example, for $y = 2.7$, $p^T = [0\ 0.3\ 0.7\ 0\ 0]$. In our case, the similarity score y is a value between 1 and 5.

To compute our training data we automatically convert the human rankings of the WMT-13 evaluation data into similarity scores between the reference and the translation. These translation-reference pairs labelled with similarity scores are used for training. We also augment the WMT-13 data with 4500 pairs from the SICK training set (Marelli et al., 2014), resulting in a training dataset of 14059 pairs in total.

The metric uses *Glove* word vectors (Pennington et al., 2014) and the simple LSTM, the dependency Tree-LSTM and neural network implementations by Tai et al. (2015). Training is performed using a mini batch size of 25 with learning rate 0.05 and regularization strength 0.0001. The memory dimension is 300, hidden dimension is 100 and compositional parameters are 541,800. Training is performed for 10 epochs. System level scores are computed by aggregating and normalising the segment level scores. Full details can be found in (Gupta et al., 2015).[2]

5 Results

The results for WMT-15 are presented in Table 1 and Table 2.

Table 1 shows system-level Pearson correlation (TrueSkill) (see (Bojar et al., 2013) for difference between TrueSkill and Pre-TrueSkill system-ranking approaches) obtained on different language pairs as well as average (PAvg) over all language pairs. The second last column shows average Pearson correlation (Pre-TrueSkill). The last column shows average Spearman correlation (SAvg). The 95% confidence level scores are obtained using bootstrap resampling as used in the WMT-2015 metric task evaluation. Table 2 shows results on segment-wise Kendall tau correlation.

The first section of Table 1 and Table 2 shows the results of our ReVal metric as UoW-LSTM, the second section shows the other four top performing metrics and the third section shows baseline metrics (BLEU, TER and WER for system-level and SENTBLEU for segment level).

Table 1 shows that our metric obtains the best results overall for both Pearson (Pre-TrueSkill) and Spearman system-level correlation and second best overall using Pearson (TrueSkill) correlation. Table 2 shows that while improving over SENTBLEU our metric does not obtain high segment level scores.

[2]Please refer to L+Sick(100, 300) in (Gupta et al., 2015) for more details and results on the WMT-14 settings.

6 Conclusion and Future Work

Our dense-vector-space-based ReVal metric is simple, elegant and fully competitive with the best of the current complex alternative approaches that involve system combination, extensive external resources, feature engineering and tuning. In future work we will investigate the difference between system and segment level evaluation scores.

Acknowledgement

The research leading to these results has received funding from the People Programme (Marie Curie Actions) of the European Unions Seventh Framework Programme FP7/2007-2013/ under REA grant agreement no. 317471 and the EC- funded project QT21 under Horizon 2020, ICT 17, grant agreement no. 645452.

References

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166.

Alexandra Birch and Miles Osborne. 2011. Reordering metrics for MT. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1027–1035. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Barry Haddow, Matthias Huck, Philipp Koehn, Matteo Negri, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria, August. Association for Computational Linguistics.

Michael Collins and Nigel Duffy. 2001. Convolution kernels for natural language. In *Advances in Neural Information Processing Systems*, pages 625–632.

Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation

for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pages 138–145. Morgan Kaufmann Publishers Inc.

Shubham Gautam and Pushpak Bhattacharyya. 2014. Layered: Metric for machine translation evaluation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*.

Jesús Giménez and Lluís Màrquez. 2010. Linguistic measures for automatic machine translation evaluation. *Machine Translation*, 24(3-4):209–240.

Rohit Gupta, Constantin Orăsan, and Josef van Genabith. 2015. Reval: A simple and effective machine translation evaluation metric based on recurrent neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Lisbon, Portugal.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Shafiq Joty, Francisco Guzmán, Lluís Màrquez, and Preslav Nakov. 2014. DiscoTK: Using Discourse Structure for Machine Translation Evaluation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of The 31st International Conference on Machine Learning*, pages 1188–1196.

Matouš Machácek and Ondrej Bojar. 2014. Results of the WMT-14 metrics shared task. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A sick cure for the evaluation of compositional distributional semantic models. In *Proceedings of LREC 2014*.

Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013a. Exploiting Similarities among Languages for Machine Translation. *CoRR*, pages 1–10.

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*, pages 3111–3119.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the ACL*, pages 311–318.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, pages 1532–1543.

Matthew Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. 2008. TERp system description. In *MetricsMATR workshop at AMTA*. Citeseer.

Richard Socher, Eh Huang, and Jeffrey Pennington. 2011. Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection. In *Advances in Neural Information Processing Systems*, pages 801–809.

Richard Socher, John Bauer, Christopher D. Manning, and Andrew Y. Ng. 2013a. Parsing With Compositional Vector Grammars. *In Proceedings of the ACL*, pages 455–465.

Richard Socher, Alex Perelygin, and Jy Wu. 2013b. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*, pages 1631–1642.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing, China. Association for Computational Linguistics.

Wojciech Zaremba and Ilya Sutskever. 2014. Learning to execute. *arXiv preprint arXiv:1410.4615*.