MaxSD: A Neural Machine Translation Evaluation Metric Optimized by Maximizing Similarity Distance

Qingsong Ma[1,2(✉)], Fandong Meng[1,2], Daqi Zheng[1,2], Mingxuan Wang[1,2], Yvette Graham[3], Wenbin Jiang[1,2], and Qun Liu[1,3]

[1] Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
{maqingsong,mengfandong,zhengdaqi,wangmingxuan,jiangwenbin}@ict.ac.cn, qun.liu@dcu.ie
[2] University of Chinese Academy of Sciences, Beijing, China
[3] ADAPT Centre, School of Computing, Dublin City University, Dublin, Ireland
graham.yvette@gmail.com

Abstract. We propose a novel metric for machine translation evaluation based on neural networks. In the training phrase, we maximize the distance between the similarity scores of high and low-quality hypotheses. Then, the trained neural network is used to evaluate the new hypotheses in the testing phase. The proposed metric can efficiently incorporate lexical and syntactic metrics as features in the network and thus is able to capture different levels of linguistic information. Experiments on WMT-14 show state-of-the-art performance is achieved in two out of five language pairs on the system-level and one on the segment-level. Comparative results are also achieved in the remaining language pairs.

Keywords: Machine translation evaluation · Neural networks · Similarity distance · Maximization

1 Introduction

With the development of machine translation (MT), MT evaluation (MTE) has received increasing attention. Traditional lexical-based metrics such as BLEU [8], Meteor [3], and TERp [11] take n-grams, synonyms, stems, word order, and phrases into account. However, metrics based on lexical and syntactic information are insufficient to evaluate the quality of the hypotheses, due to mismatch errors caused by limited synonyms and references.

Recently, semantic-based metrics have become more feasible with the help of deep learning. This paper presents an effective metric based on neural networks, i.e. Bidirectional Long Short Term Memory (Bi-LSTM) network [7,10] for MTE. To capture the inner connection between hypotheses and references, we also explore the effect of an enhanced Bidirectional Combined LSTM (BiC-LSTM)

C.-Y. Lin et al. (Eds.): NLPCC-ICCPOL 2016, LNAI 10102, pp. 153–161, 2016.
DOI: 10.1007/978-3-319-50496-4_13

network, which takes the concatenation of the hypothesis and the reference as the input, rather than feeding them separately into the network as Bi-LSTM does.

Generally, the goal of the framework is to predict quality scores of hypotheses, which requires references and hypotheses together with quality scores as training examples. However, the difficulty of obtaining hypotheses with quality scores leads to the insufficiency of training examples. For instance, ReVal [6] devotes extra effort to compute quality scores of hypotheses, producing less than 15 thousand training examples from the human judgement file of WMT-13 [1], and subsequently requires extra resources to enlarge the training set. As the amount of training examples is crucial to network performance, we design a new objective during the training process, which maximizes the distance between two similarity scores: one between the reference ref and a high-quality hypothesis $posh$, and the other one between ref and a low-quality one $negh$. Thus, two hypotheses, as well as the reference comprise a training example, which allows us to extract adequate training examples from WMT human judgements. Furthermore, for testing, the network takes only one hypothesis and one reference as an input, then outputs an evaluation score of the hypothesis. Compared with Guzmán et al. (2015), our metric significantly reduces complexity in this respect, as we can evaluate with a single hypothesis, while they require a pairwise setting. Experiments on WMT-14 show that state-of-the-art performance is achieved in two out of five language pairs on the system-level and one on the segment-level, comparative results are obtained for remaining language pairs.

2 Learning Task

The goal of the training process in our neural network is to maximize the distance of the similarity score between ref and $posh$, and the other one between ref and $negh$. In the testing process, we evaluate the quality of hyp given ref by computing the similarity score between them.

Thus, the input of our neural network is a tuple, marked as $(ref, posh, negh)$. The loss function of the neural network is formulated as follows:

$$J_\theta = -\sum_n max(0, simP - simN) \tag{1}$$

where $simP$ is the similarity score between ref and $posh$, and $simN$ is that between ref and $negh$. A more detailed computation is illustrated below.

3 MaxSD Model: Maximizing Similarity Distance Model

3.1 MaxSD Model

In order to learn the similarity scores, $simP$ and $simN$, we build a maxSD model. We explore two versions of MaxSD model, the performance of two LSMT networks, namely Bi-LSTM and BiC-LSTM. As showed in Fig. 1, we first obtain

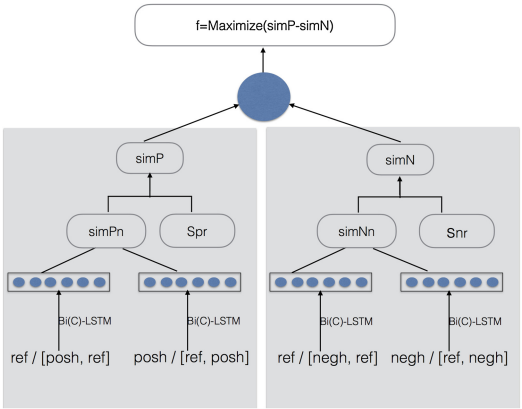

Fig. 1. The overall architecture of the maxSD model. Bi(C)-LSTM means either Bi-LSTM or BiC-LSTM network. Bi-LSTM network takes the left side of '/' as input, while BiC-LSTM the right one. The Bi-LSTM or BiC-LSTM network produces the representation of each input, which then are used to compute $simPn$ and $simNn$. $simP$ and $simN$ are computed by incorporating 5 metric scores, namely s_{pr} and s_{nr} respectively. The objective of the architecture is to maximize the distance between $simP$ and $simN$ are.

the continuous space representations of ref, $posh$, and $negh$ through the Bi-LSTM and BiC-LSTM networks, respectively. Then, the representations are fed into a feed-forward neural network as inputs to obtain neural network(NN)-based similarity scores, which are computed as below:

$$simPn = \sigma(\mathbf{V} \cdot \sigma(\mathbf{W}[ref_r, posh_r] + \mathbf{b})) \tag{2}$$

$$simNn = \sigma(\mathbf{V} \cdot \sigma(\mathbf{W}[ref_r, negh_r] + \mathbf{b})) \tag{3}$$

where $posh_r$ denotes the representation of $posh$, and $negh_r$ of $negh$. $simPn$ refers to the NN-based similarity score of $posh$, while $simNn$ of $negh$ given ref_r. $simPn$ and $simNn$ share the same parameter weights $\mathbf{W}$, $\mathbf{V}$ and the bias term $\mathbf{b}$.

Incorporating Other Metrics. Next, we further optimize our model by incorporating lexical and syntactic metrics as features (in terms of metric scores), namely BLEU, NIST, METEOR, TERp and DPMF [13]. The concateanation of these 5 metric scores and the NN-based similarity scores are fed into a feed-forward layer, whose output shows the final similarity scores, $simP$ and $simN$ (mentioned in formula (1)).

$$simP = \sigma(\mathbf{W_s}[simPn, s_{pr}] + \mathbf{b_s}) \tag{4}$$

$$simN = \sigma(\mathbf{W_s}[simNn, s_{nr}] + \mathbf{b_s}) \tag{5}$$

where $\mathbf{W_s}$ is the parameter weight and $\mathbf{b_s}$ is a bias term. s_{nr} refers to the concatenated 5 metric scores of neg, while s_{pr} that of pos.

The Testing Phase. During the testing phase, given a hypothesis hyp and a corresponding reference ref, the similarity score between them is computed as follows.

Firstly, the NN-based similarity score:

$$sim(ref, hyp) = \sigma(\mathbf{V} \cdot \sigma(\mathbf{W}[ref_r, hyp_r] + \mathbf{b})) \tag{6}$$

where ref_r denotes the representation of ref, and hyp_r of hyp. $\mathbf{W}$ and $\mathbf{V}$ are parameter weights, and $\mathbf{b}$ is the bias term. All $\mathbf{W}$, $\mathbf{V}$ and $\mathbf{b}$ are the same with that in the training phase. Then, the final similarity score

$$sim = \sigma(\mathbf{W_s}[sim(ref, hyp), s_r] + \mathbf{b_s}) \tag{7}$$

where $\mathbf{W_s}$, $\mathbf{b_s}$ are the same with that in the training phase. s_r refers to the concatenated 5 metric scores of hyp given ref.

3.2 Bi-LSTM and BiC-LSTM Networks

We use Bi-LSTM and BiC-LSTM networks separately to produce the continuous space representations of ref, $posh$ and $negh$, which are denoted as ref_r, $posh_r$ and $negh_r$.

Bi-LSTM Network. Bi-LSTM networks have been employed to substantially improve performance in several NLP tasks. As illustrated in Fig. 1, Bi-LSTM network consists of two parallel layers, a forward and a backward layer, propagating in two directions. These two layers enable the network to capture both past and future features for a given timestep. The two representation sequences produced by each layer are concatenated at each timestep, followed by mean pooling which outputs the representation of the sentence.

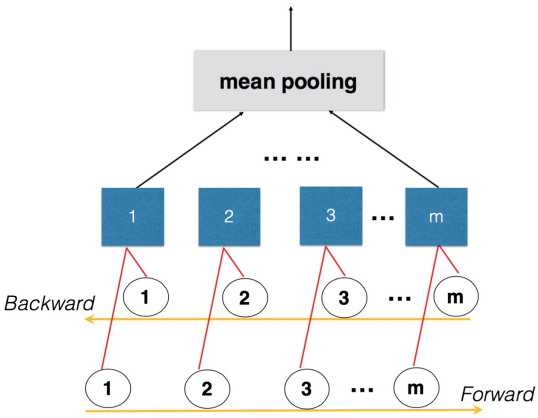

Fig. 2. The Bi-LSTM network. The circles marked from 1 to m consist of a sentence, whose representation is trained by the Bi-LSTM network.

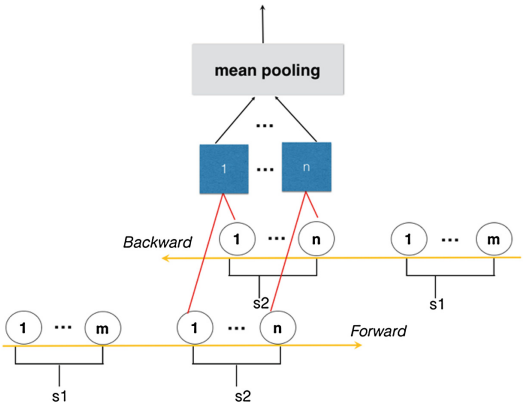

Fig. 3. The BiC-LSTM network. $s1$ denotes a sentence with length of m, while $s2$ the other with that of n. $s1$ and $s2$ are concatenated to go through the BiC-LSTM network, producing the representation of the second sentence $s2$, which contains the inner connection between $s1$ and $s2$.

BiC-LSTM Network. In order to capture inner connection between two sentences, we further propose an enhanced BiC-LSTM network (as illustrated in Fig. 2), which takes the concatenation of the two sentences as input. The output is the representation of the second sentence. For instance, if the input of the forward layer is the concatenation of hyp and ref, denoted by $[hyp, ref]$, and that of the backward layer is the concatenation of reversals of both hyp and ref, then the network produces the representation of ref (Fig. 3).

4 Experiments and Results

4.1 Datasets

Experiments are conducted on the WMT metric shared task. Each training example is a tuple $(ref, posh, negh)$, extracted from the human judgement file of WMT-13, of which each line contains 5 human ranks of 5 randomly chosen hypotheses of a specific segment.

For duplicated tuples, we only retain one of them. There are also two tuples with opposite positions of $posh$ and $negh$ due to the inconsistent ranks between two annotators [2], in which case we remove the tuple appearing less often. Hence, we clean the training with respect to inconsistency and redundance. In all, we obtain 285908 tuples for training. Evaluation is conducted on WMT-14 for other languages into English.

4.2 Setups

Sentences with lengths exceeding 100 words are filtered out. The 300-dimensional *glove* word vectors [9] are used as the word embedding. The parameter weights

are initialized by sampling from a normal distribution. We train for 10 epoches using adadelta. Our mini-batch size is 16, and dropout is used as suggested by [14]. The average of segment-level scores is the system-level score.

4.3 Results

We present two versions of our metric, namely maxSD-1 and maxSD-2 based on Bi-LSTM and BiC-LSTM networks respectively. We compare our metric with the best two in WMT-14, DISCOTK-PARTY-TUNED and BEER [12] on segment-level, and DISCOTK-PARTY-TUNED and LAYERED [4] on system-level respectively. Additionally, the other incorporated metrics are also listed in Tables 1 and 2 for comparison. Scores in bold indicate best scores overall and those in bold italic show best scores achieved by our metric. Results in Tables 1 and 2 show that two versions of our metric outperform all other metrics, except DISCOTK-PARTY-TUNED, in all five directions both at the segment- and system-level. And our metrics are slightly behind the top-performing metric DISCOTK-PARTY-TUNED, which combines 17 different metrics requiring external resources and tuning efforts. However, for 'hi-en', we yield better results than DISCOTK-PARTY-TUNED, achieving the state-of-the-art results, with Kendall tau of 0.444 on the segment level and Pearson correlation of 0.979 on the system level. It is also worthy noting that maxSD-2 achieves the best performance in two ('hi-en' and 'fr-en') out of five directions at the system-level, and maxSD-1 the best in one direction at the segment-level. One interesting finding is that the enhanced maxSD-2 does not outperform maxSD-1. We suspect that the long length of the concatenated sentence affects the performance of BiC-LSTM network. As recommended by [5], significance tests for differences in dependent correlation with human assessment were carried out for all competing metrics. Results of significance tests are shown in Fig. 4.

Table 1. Segment-level Kendall's tau correlations on WMT-14.

Metrics	cs-en	de-en	fr-en	ru-en	hi-en	PAvg
BLEU	.218	.266	.376	.263	.299	.285
NIST	.231	.295	.392	.285	.342	.309
TERp-A	.293	.335	.389	.307	.407	.346
METEOR	.282	.334	.406	.333	.407	.355
DPMF	.283	.332	.404	.324	.426	.354
maxSD-1	***.312***	***.353***	.429	***.342***	***.444***	***.376***
maxSD-2	.310	***.353***	***.431***	***.342***	.440	.375
DISCOTK-PARTY-TUNED	**.328**	**.380**	**.433**	**.355**	.434	**.386**
BEER	.284	.337	.417	.333	.438	.362

Table 2. System-level correlations on WMT-14.

Metrics	cs-en	de-en	fr-en	ru-en	hi-en	Average
BLEU	.963	.830	.961	.784	.928	.893
NIST	.949	.803	.964	.796	.667	.836
TERp-A	.863	.909	.976	.815	.438	.800
METEOR	.980	.927	.975	.807	.457	.829
DPMF	**.999**	.920	.967	.832	.882	.920
maxSD-1	.945	***.920***	**.977**	***.827***	.978	***.930***
maxSD-2	***.948***	.919	**.977**	.825	**.979**	***.930***
DISCOTK-PARTY-TUNED	.975	**.943**	**.977**	**.870**	.956	**.944**
LAYERED	.941	.893	.973	.854	.976	.927

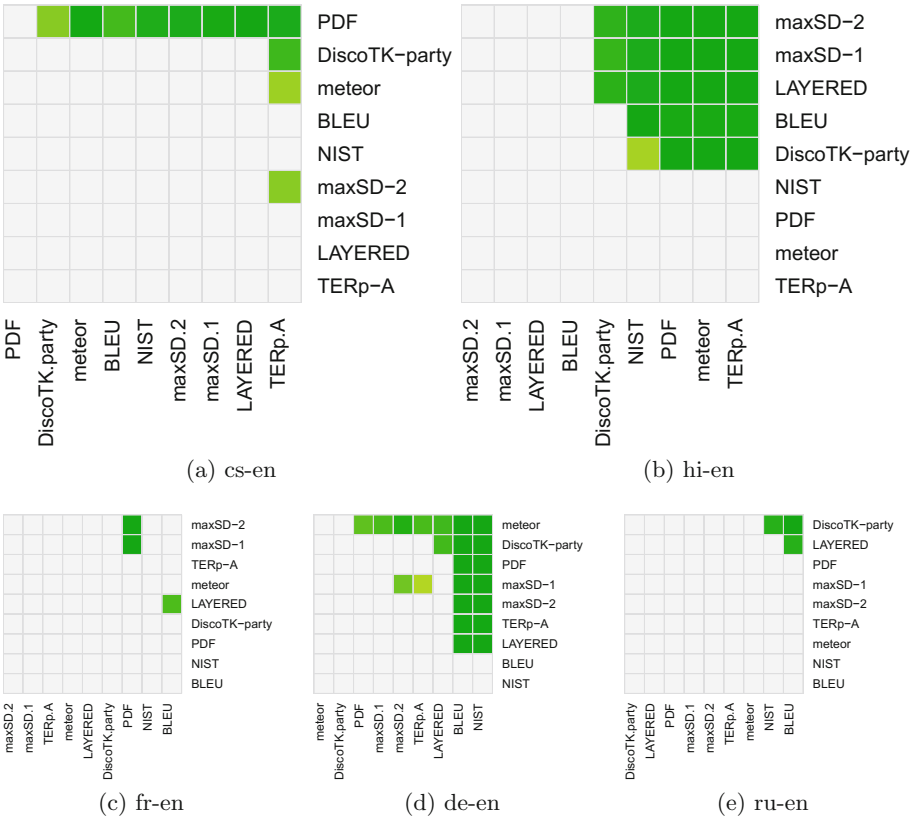

Fig. 4. Significance test results for differences in dependent correlation with human judgement (Williams test) for all competing pairs of metrics. A green cell denotes a significant win for the metric in a given row over the metric in a given column at $p < 0.05$."PDF" in the figure corresponds to "DPMF" mentioned above. (Color figure online)

5 Conclusion

Our proposed metric based on neural networks effectively achieves the state-of-the-art performance in two out of five language pairs on system-level and one on segment-level, and achieve comparative results for the remaining language pairs.

Acknowledgements. This work is supported by National Natural Science Foundation of P. R. China under Grant No. 61379086, European Union Horizon 2020 research and innovation programme under grant agreement 645452 (QT21), and the ADAPT Centre for Digital Content Technology (www.adaptcentre.ie) at Dublin City University funded under the SFI Research Centres Programme (Grant 13/RC/2106) co-funded under the European Regional Development Fund.

References

1. Bojar, O., Buck, C., Callison-Burch, C., Federmann, C., Haddow, B., Koehn, P., Monz, C., Post, M., Soricut, R., Specia, L.: Findings of the 2013 workshop on statistical machine translation. In: Proceedings of the Eighth Workshop on Statistical Machine Translation, Sofia, Bulgaria, pp. 1–44. Association for Computational Linguistics, August 2013
2. Callison-Burch, C., Fordyce, C., Koehn, P., Monz, C., Schroeder, J.: (Meta-) evaluation of machine translation. In: Proceedings of the Second Workshop on Statistical Machine Translation, pp. 136–158. Association for Computational Linguistics (2007)
3. Denkowski, M., Lavie, A.: Meteor universal: language specific translation evaluation for any target language. In: Proceedings of the Ninth Workshop on Statistical Machine Translation. Citeseer (2014)
4. Gautam, S., Bhattacharyya, P.: LAYERED: metric for machine translation evaluation. In: ACL 2014, p. 387 (2014)
5. Graham, Y., Baldwin, T.: Testing for significance of increased correlation with human judgment. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, pp. 172–176. Association for Computational Linguistics, October 2014. http://www.aclweb.org/anthology/D14-1020
6. Gupta, R., Orasan, C., van Genabith, J.: 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, Lisbon, Portugal, pp. 1066–1072. Association for Computational Linguistics, September 2015
7. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
8. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 311–318. Association for Computational Linguistics (2002)
9. Pennington, J., Socher, R., Manning, C.D.: GloVe: global vectors for word representation. EMNLP **14**, 1532–1543 (2014)
10. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Trans. Signal Process. **45**(11), 2673–2681 (1997)

11. Snover, M., Madnani, N., Dorr, B., Schwartz, R.: Fluency, adequacy, or HTER? Exploring different human judgments with a tunable MT metric. In: Proceedings of the Fourth Workshop on Statistical Machine Translation, Athens, Greece, pp. 259–268. Association for Computational Linguistics, March 2009
12. Stanojevic, M., Sima'an, K.: BEER: better evaluation as ranking. In: Proceedings of the Ninth Workshop on Statistical Machine Translation, Baltimore, Maryland, USA, pp. 414–419. Association for Computational Linguistics, June 2014
13. Yu, H., Wu, X., Jiang, W., Liu, Q., Lin, S.: An automatic machine translation evaluation metric based on dependency parsing model. arXiv preprint arXiv:1508.01996 (2015)
14. Zaremba, W., Sutskever, I., Vinyals, O.: Recurrent neural network regularization. arXiv preprint arXiv:1409.2329 (2014)