Better Rewards Yield Better Summaries: Learning to Summarise Without References Supplementary Material

1 Reward Learning Architectures

CNN. The architecture of the CNN encoder is visualised in Fig. 1 (taken from (Kim, 2014)) for two different filter widths (red=2, orange=3). The feature maps are reduced with max-over-time pooling to a fixed-size feature vector.

PMeans. For a sentence s with L words taken from x or y, PMeans vectorise s as

$$\phi(s;p) = \left(\frac{\mathbf{z}_1^p + \cdots + \mathbf{z}_L^p}{L}\right)^{1/p} \in \mathbb{R}^e, \quad (1)$$

where $\mathbf{z}_i$ is the word embedding of the ith word in s and e is the dimension of the word embeddings. The addition, multiplication and exponentiation operations are element-wise. $p \in \mathbb{R} \cup \{\pm\infty\}$ controls "weights" assigned to each element: with $p = 1$, for instance, each word element is weighted equally and $\phi(s;1)$ is the standard average-pooling; with the increase of p, it assigns higher weights to the elements with higher values. With $p = +\infty$, PMeans is equivalent to element-wise max-pooling.

BERT. The BERT encoder with sliding window is shown in Fig. 2. The summary and document are encoded independently. All tokens of the same sliding window will be passed to BERT that produces an embedding for each sliding window. Average-over-time pooling (element-wise) reduces these embeddings to one embedding per summary and document. The final embedding is the concatenation of both.

MLP. The MLP at the bottom part of Fig. 2 has one fully-connected layer with ReLU activation and a single output with linear activation. During training, dropout (Srivastava et al., 2014) with 50% chance is applied at the input and hidden layer for regularisation. For CNN-RNN and PMeans-RNN encoders, the hidden layer is of size 100. For BERT+MLP we use a hidden layer of size 1024.

Additional Layer If PMeans and BERT sentence embeddings are used for SimRed, an additional layer is put on top of the embeddings because PMeans has no trainable parameters and the BERT model weights are kept fixed. For this purpose we use a fully-connected layer with ReLU activation, where the output dimension is equal to the input dimension, shown in Fig. 3.

SimRed. Fig. 4 illustrates the SimRed architecture and shows the Similarity and Redundancy matrices.

2 Significance Tests

To proof that BERT+MLP+Pref outperforms other learned metrics significantly, we use the double-tailed t-test. Table 1 shows the p-values in comparison of BERT+MLP+Pref against our other approaches for Spearman's ρ, Pearson's r, G-Pre and G-Rec on summary level. Table 2 shows the p-values for BERT+MLP+Pref in comparison to reward metrics that require reference summaries (e.g. ROUGE and BLEU).

3 Reward/Metric Distribution

Fig. 5 and 6 illustrate the distributions of multiple metrics and learned rewards for summaries with different human ratings, respectively. An interesting pattern we observe is that, for summaries with the lowest human rating (-1), their metrics/rewards are mostly uniformly distributed, regardless of the type of metrics or rewards. This is because the summaries in our training set are generated by two state-of-the-art summarisation systems, hence are mostly of relatively high quality. As such, our reward learning models witness only few low-

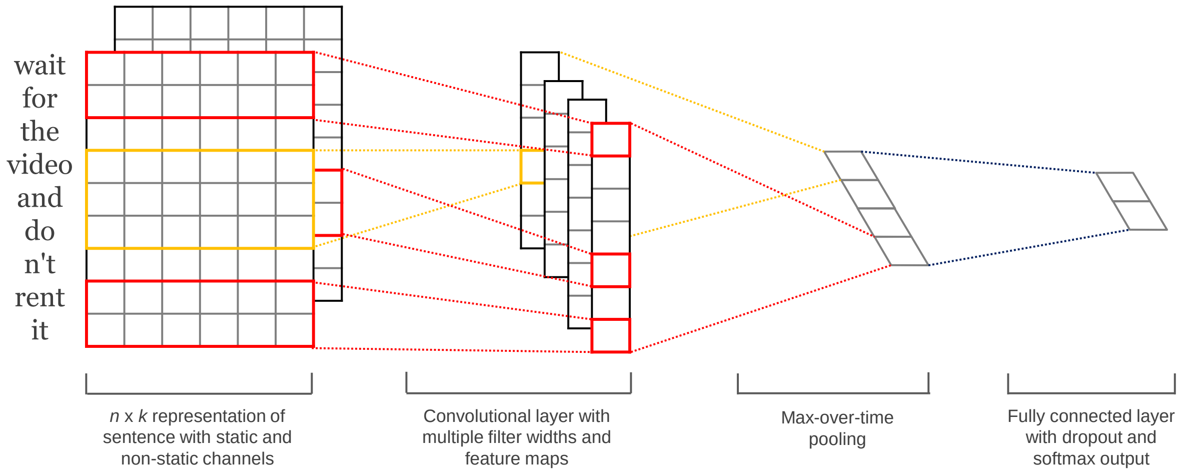
wait
for
the
video
and
do
n't
rent
it
n x k representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

Figure 1: CNN encoder architecture. Figure taken from (Kim, 2014).

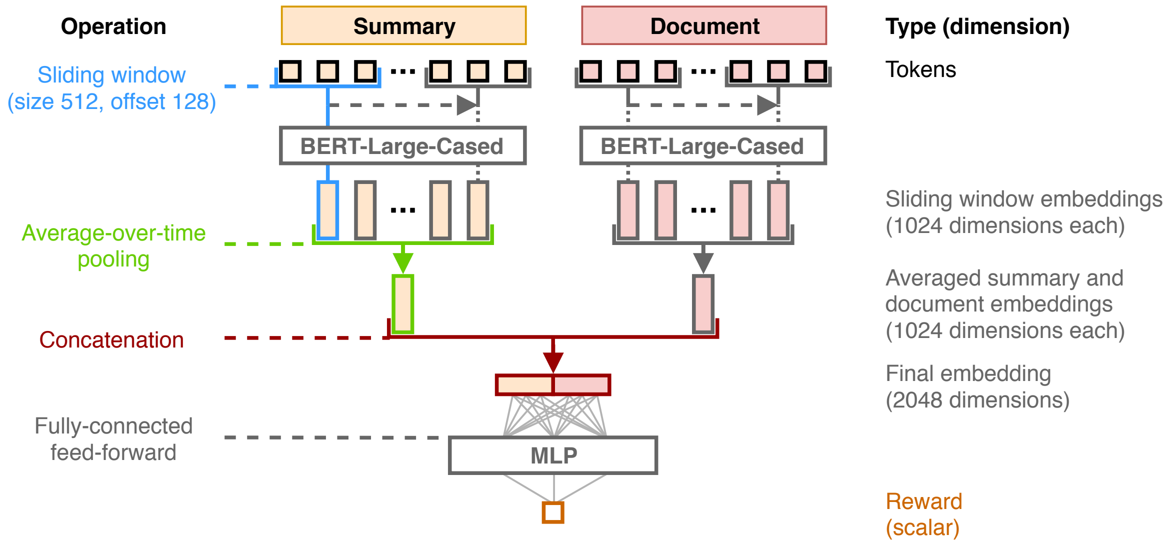
Operation
Summary
Document
Type (dimension)
Sliding window (size 512, offset 128)
Tokens
BERT-Large-Cased
BERT-Large-Cased
Sliding window embeddings (1024 dimensions each)
Average-over-time pooling
Averaged summary and document embeddings (1024 dimensions each)
Concatenation
Final embedding (2048 dimensions)
Fully-connected feed-forward
MLP
Reward (scalar)

Figure 2: BERT with sliding window as independent encoder for a summary and its corresponding document.

Model	Encoder	Loss	ρ	r	G-Pre	G-Rec
MLP	CNN-RNN	Regr.	5.61e-12	2.56e-09	3.97e-04	3.85e-04
	CNN-RNN	Pref.	3.13e-11	2.05e-07	1.09e-04	8.11e-05
	PMeans-RNN	Regr.	3.02e-11	3.93e-07	9.89e-04	1.03e-03
	PMeans-RNN	Pref.	5.33e-08	5.65e-06	6.67e-03	1.55e-02
	BERT	Regr.	.468	.106	.551	.615
SimRed	CNN	Regr.	4.07e-09	3.03e-07	2.30e-05	2.41e-05
	CNN	Pref.	4.83e-05	2.10e-05	3.41e-03	5.88e-03
	PMean	Regr.	7.41e-08	2.83e-05	1.32e-03	1.58e-03
	PMean	Pref.	1.33e-06	1.70e-04	.013	8.12e-03
	BERT	Regr.	5.33e-16	4.32e-09	1.30e-06	2.46e-07
	BERT	Pref.	2.61e-10	2.54e-06	4.24e-04	5.83e-04

Table 1: P-values of double-tailed t-test between BERT+MLP+Pref and our other approaches.

Metric	ρ	r	G-Pre	G-Rec
ROUGE-1	6.85e-17	<u>.201</u>	1.17e-14	3.81e-15
ROUGE-2	2.98e-20	.015	1.04e-13	8.35e-15
ROUGE-L	4.09e-19	.020	4.78e-14	6.53e-15
ROUGE-SU4	1.57e-18	.013	2.67e-13	4.95e-14
BLEU-1	1.64e-15	<u>.160</u>	2.49e-09	3.05e-16
BLEU-2	4.33e-17	.029	2.98e-09	4.23e-16
BLEU-3	5.75e-19	.009	3.53e-09	5.13e-16
BLEU-4	6.99e-19	.006	1.11e-08	2.16e-15
BLEU-5	1.47e-18	.004	5.53e-08	1.65e-14
METEOR	1.03e-14	.039	2.2e-14	7.22e-15
InferSent-Cosine	1.45e-11	<u>.798</u>	1.33e-12	6.76e-12
BERT-Cosine	2.16e-12	<u>.769</u>	8.61e-10	1.84e-09

Table 2: P-values of double-tailed t-test between BERT+MLP+Pref and metrics that use reference summaries. The underlined values in the column for Pearson's r are $p > \alpha = 0.05$, hence a significant difference can not be assumed in these cases.

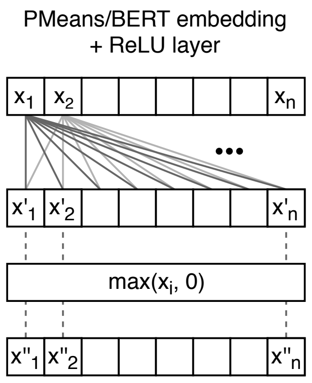

Figure 3: One fully-connected ReLU layer is put on top of PMeans and BERT when used with the SimRed model.

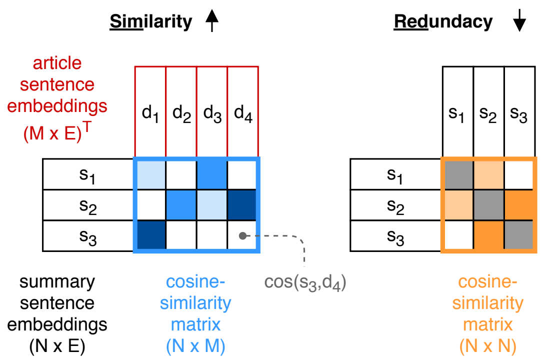
Similarity
Redundacy
article sentence embeddings (M x E)^T
d_1 d_2 d_3 d_4
s_1 s_2 s_3
summary sentence embeddings (N x E)
cosine-similarity matrix (N x M)
cos(s_3,d_4)
cosine-similarity matrix (N x N)

Figure 4: SimRed cosine similarity matrices. E is the dimension of the sentence embeddings.

quality summaries during training, and thus can hardly allocate low scores to the summaries with low human ratings. This observation also shows that it is important to have a wide diversity of summaries in the training set.

4 NeuralTD

The original reinforcement learning algorithm used in (Ryang and Abekawa, 2012) is the linear temporal-difference algorithm (LinearTD). For a draft summary s, $V(s; w) = w \cdot \phi(s)$, where $\phi(s)$ is vector representation for s. In NeuralTD, we approximate the V-values with a neural network illustrated in Fig. 7. It reads $\phi(s)$ as input, and use two hidden ReLU layers after the input, and finally output a single real value. The dimension of the hidden layer is the half of the size of the input vector $\phi(s)$.

We test the performance of NeuralTD on the same DUC datasets where LinearTD is tested in (Ryang and Abekawa, 2012). We use the sum of ROUGE-1 and ROUGE-2 as rewards to train both LinearTD and NeuralTD. Because both algorithms are stochastic, we repeat each algorithm by 10 times. The averaged results are reported in Table 3. NeuralTD outperforms LinearTD in terms of all ROUGE metrics we considered, and some of the improvements are significant.

References

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751.

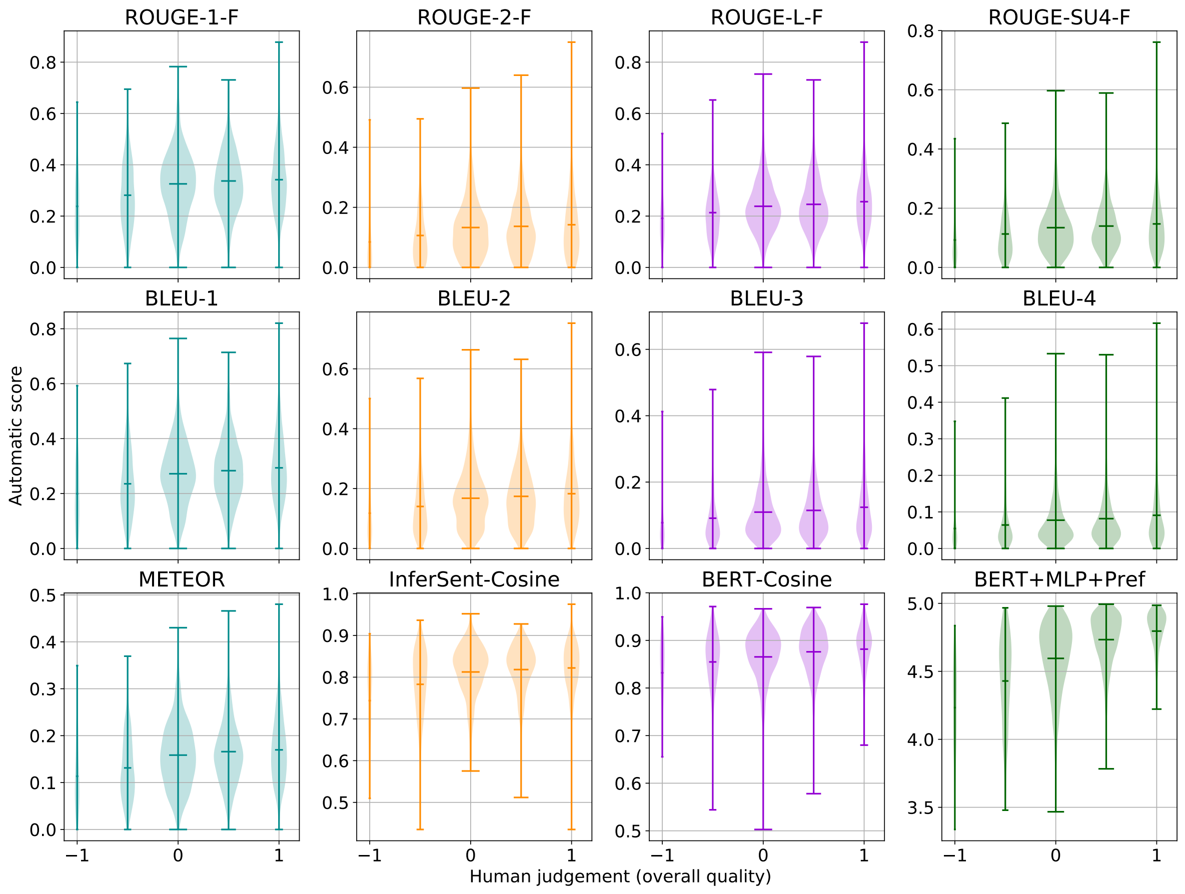

Figure 5: Distributions of rewards/metrics for summaries with different human ratings. Among all presented, only BERT+MLP+Pref (third row, rightmost sub-figure) does not use reference summaries.

Dataset	RL	R-1	R-2	R-L	R-SU4
DUC'01	LinearTD	.442	.161	.349	.172
	NeuralTD	**.452***	**.169**	**.359***	**.177**
DUC'02	LinearTD	.475	.179	.374	.189
	NeuralTD	**.483***	**.181**	**.379**	**.193**
DUC'04	LinearTD	.473	.174	.378	.192
	NeuralTD	**.492***	**.189**	**.391***	**.203***

Table 3: ROUGE Recall scores of NeuralTD and LinearTD on three DUC datasets. All summaries generated by both systems meet the 100-word length restriction. Asterisk indicates significant improvement ($p < 0.05$, double-tailed t-test).

Seonggi Ryang and Takeshi Abekawa. 2012. Framework of automatic text summarization using reinforcement learning. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, pages 256–265.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

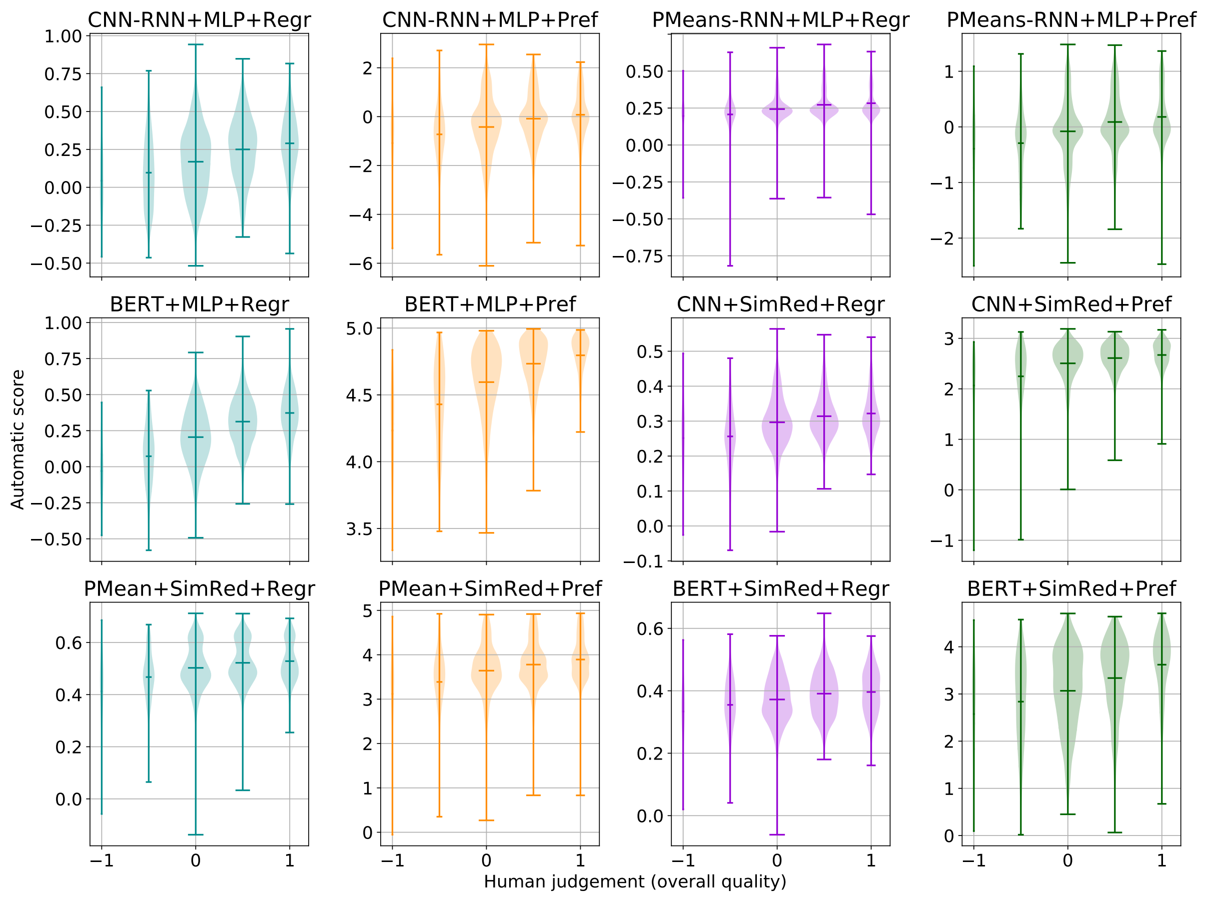

Figure 6: Distributions of learned rewards for summaries with different human ratings.

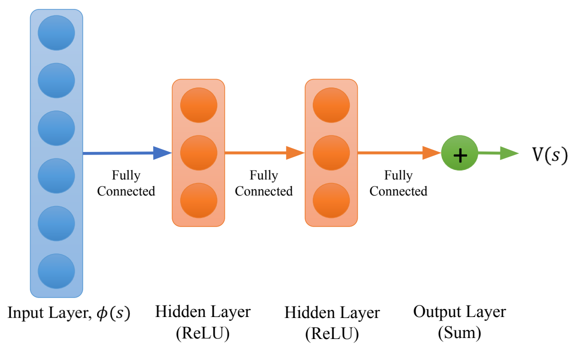

Figure 7: NeuralTD arthitecture.