Better Rewards Yield Better Summaries: Learning to Summarise Without References

Florian Böhm[1], **Yang Gao**[1*], **Christian M. Meyer**[1],
Ori Shapira[2], **Ido Dagan**[2], **and Iryna Gurevych**[1]
[1]Ubiquitous Knowledge Processing Lab, Technische Universität Darmstadt, Germany
[2]Computer Science Department, Bar-Ilan University, Ramat-Gan, Israel
https://www.ukp.tu-darmstadt.de/
yang.gao@rhul.ac.uk, obspp18@gmail.com, dagan@cs.biu.ac.il

Abstract

Reinforcement Learning (RL) based document summarisation systems yield state-of-the-art performance in terms of ROUGE scores, because they directly use ROUGE as the *rewards* during training. However, summaries with high ROUGE scores often receive low human judgement. To find a better reward function that can guide RL to generate human-appealing summaries, we learn a reward function from human ratings on 2,500 summaries. Our reward function only takes the document and system summary as input. Hence, once trained, it can be used to train RL-based summarisation systems without using any reference summaries. We show that our learned rewards have significantly higher correlation with human ratings than previous approaches. Human evaluation experiments show that, compared to the state-of-the-art supervised-learning systems and ROUGE-as-rewards RL summarisation systems, the RL systems using our learned rewards during training generate summaries with higher human ratings. The learned reward function and our source code are available at https://github.com/yg211/summary-reward-no-reference.

1 Introduction

Document summarisation aims at generating a summary for a long document or multiple documents on the same topic. *Reinforcement Learning (RL)* becomes an increasingly popular technique to build document summarisation systems in recent years (Chen and Bansal, 2018; Narayan et al., 2018b; Dong et al., 2018). Compared to the supervised learning paradigm, which "pushes" the summariser to reproduce the reference summaries at training time, RL directly optimises the summariser to maximise the *rewards*, which measure the quality of the generated summaries.

The *ROUGE* metrics (Lin, 2004b) are the most widely used rewards in training RL-based summarisation systems. ROUGE measures the quality of a generated summary by counting how many n-grams in the reference summaries appear in the generated summary. ROUGE correlates well with human judgements at *system level* (Lin, 2004a; Louis and Nenkova, 2013), i.e. by aggregating system summaries' ROUGE scores across multiple input documents, we can reliably rank summarisation systems by their quality. However, ROUGE performs poorly at *summary level*: given multiple summaries for the same input document, ROUGE can hardly distinguish the "good" summaries from the "mediocre" and "bad" ones (Novikova et al., 2017). Because existing RL-based summarisation systems rely on summary-level ROUGE scores to guide the optimisation direction, the poor performance of ROUGE at summary level severely misleads the RL agents. The reliability of ROUGE as RL reward is further challenged by the fact that most large-scale summarisation datasets only have one reference summary available for each input document (e.g. CNN/DailyMail (Hermann et al., 2015; See et al., 2017) and NewsRooms (Grusky et al., 2018)).

In order to find better rewards that can guide RL-based summarisers to generate more human-appealing summaries, we learn a reward function directly from human ratings. We use the dataset compiled by Chaganty et al. (2018), which includes human ratings on 2,500 summaries for 500 news articles from CNN/DailyMail. Unlike ROUGE that requires one or multiple reference summaries to compute the scores, our reward function only takes the document and the generated summary as input. Hence, once trained, our

* Since June 2019, Yang Gao is affiliated with Dept. of Computer Science, Royal Holloway, University of London.

reward can be used to train RL-based summarisation systems without any reference summaries.

The contributions of this work are threefold: **(i)** We study the summary-level correlation between ROUGE and human judgement on 2,500 summaries (§3), explicitly showing that ROUGE can hardly identify the human-appealing summaries. **(ii)** We formulate the reward learning problem as either a regression or a preference learning problem (§4), and explore multiple text encoders and neural architectures to build the reward learning model (§5). **(iii)** We show that our learned reward correlates significantly better with human judgements than ROUGE (§6), and that using the learned reward can guide both extractive and abstractive RL-based summarisers to generate summaries with significantly higher human ratings than the state-of-the-art systems (§7).

2 Related Work

RL-based summarisation. Most existing RL-based summarisers fall into two categories: *cross-input* systems and *input-specific* systems (Gao et al., 2019). Cross-input systems learn a summarisation policy at training time by letting the RL agent interact with a ROUGE-based reward function. At test time, the learned policy is used to generate a summary for each input document. Most RL-based summarisers fall into this category (Chen and Bansal, 2018; Narayan et al., 2018b; Dong et al., 2018; Kryscinski et al., 2018; Pasunuru and Bansal, 2018; Paulus et al., 2018). As an alternative, input-specific RL (Rioux et al., 2014; Ryang and Abekawa, 2012) does not require reference summaries: for each input document at test time, a summarisation policy is trained specifically for the input, by letting the RL summariser interact with a heuristic-based reward function, e.g. ROUGE between the generated summary and the input document (without using any reference summaries). However, the performance of input-specific RL falls far behind the cross-input counterparts.

In §7 we use our learned reward to train both cross-input and input-specific RL systems. A similar idea has been explored by Gao et al. (2019), but unlike their work that learns the reward from ROUGE scores, we learn our reward directly from human ratings. Human evaluation experiments suggest that our reward can guide both kinds of RL-based systems to generate human-appealing summaries without using reference summaries.

The reward learning idea is also related to *inverse RL* (IRL) (Ng and Russell, 2000). By observing some (near-)optimal sequences of actions, IRL algorithms learn a reward function that is consistent with the observed sequences. In the case of summarisation, human-written reference summaries are the (near-)optimal sequences, which are expensive to provide. Our method only needs human ratings on some generated summaries, also known as the *bandit feedback* (Kreutzer et al., 2017), to learn the reward function. Furthermore, when employing certain loss functions (see §4 and Eq. (2)), our method can even learn the reward function from preferences over generated summaries, an even cheaper feedback to elicit (Kreutzer et al., 2018; Gao et al., 2018).

Heuristic-based rewards. Prior work proposed heuristic-based reward functions to train cross-input RL summarisers, in order to strengthen certain properties of the generated summaries. Arumae and Liu (2019) propose four reward functions to train an RL-based extractive summariser, including the *question-answering competency* rewards, which encourage the RL agent to generate summaries that can answer cloze-style questions. Such questions are automatically created by removing some words in the reference summaries. Experiments suggest that human subjects can answer the questions with high accuracy by reading their generated summaries; but the human judgement scores of their summaries are not higher than the summaries generated by the state-of-the-art supervised system. Kryscinski et al. (2018) propose a simple heuristic that encourages the RL-based abstractive summariser to generate summaries with more *novel* tokens, i.e. tokens that do not appear in the input document. However, both ROUGE and human evaluation scores of their system are lower than the state-of-the-art summarisation systems (e.g. See et al., 2017). In addition, the above rewards require reference summaries, unlike our reward that only takes a document and a generated summary as input.

Rewards learned with extra data. Pasunuru and Bansal (2018) propose two novel rewards for training RL-based abstractive summarisers: *RougeSal*, which up-weights the salient phrases and words detected via a keyphrase classifier, and *Entail* reward, which gives high scores to

logically-entailed summaries using an entailment classifier. RougeSal is trained with the SQuAD reading comprehension dataset (Rajpurkar et al., 2016), and Entail is trained with the SNLI (Bowman et al., 2015) and Multi-NLI (Williams et al., 2018) datasets. Although their system achieves new state-of-the-art results in terms of ROUGE, it remains unclear whether their system generates more human-appealing summaries as they do not perform human evaluation experiments. Additionally, both rewards require reference summaries.

Louis and Nenkova (2013), Peyrard et al. (2017) and Peyrard and Gurevych (2018) build feature-rich regression models to learn a summary evaluation metric directly from the human judgement scores (Pyramid and Responsiveness) provided in the TAC'08 and '09 datasets[1]. Some features they use require reference summaries (e.g. ROUGE metrics); the others are heuristic-based and do not use reference summaries (e.g. the Jensen-Shannon divergence between the word distributions in the summary and the documents). Their experiments suggest that with only non-reference-summary-based features, the correlation of their learned metric with human judgements is lower than ROUGE; with reference-summary-based features, the learned metric marginally outperforms ROUGE. In §6, we show that our reward model does not use reference summaries but outperforms the feature-based baseline by Peyrard and Gurevych (2018) as well as ROUGE.

Unlike the above work that attempts to learn a summary evaluation metric, the target of our work is to learn a good *reward*, which is not necessarily a good *evaluation metric*. A good evaluation metric should be able to correctly rank summaries of different quality levels, while a good reward function focuses more on distinguishing the best summaries from the mediocre and bad summaries. Also, an evaluation metric should be able to evaluate summaries of different types (e.g. extractive and abstractive) and from different genres, while a reward function can be specifically designed for a single task. We leave the learning of a generic summarisation evaluation metric for future work.

3 Summary-Level Correlation Study

In this section, we study the summary-level correlation between multiple widely used summary evaluation metrics and human judgement scores, so as to further motivate the need for a stronger reward for RL. Metrics we consider include ROUGE (full length F-score), BLEU (Papineni et al., 2002) and METEOR (Lavie and Denkowski, 2009). Furthermore, in line with Chaganty et al. (2018), we also use the cosine similarity between the embeddings of the generated summary and the reference summary as metrics: we use InferSent (Conneau et al., 2017) and BERT-Large-Cased (Devlin et al., 2019) to generate the embeddings.

The human judgement scores we use are from Chaganty et al. (2018), collected as follows. First, 500 news articles were randomly sampled from the CNN/DailyMail dataset. For each news article, four summarisation systems were used to generate summaries: the *seq2seq* and *pointer* models proposed by See et al. (2017), and the *ml* and *ml+rl* models by Paulus et al. (2018). Hence, together with the human-written reference summaries provided in the CNN/DailyMail dataset, each news article has five summaries. Crowd-sourcing workers were recruited to rate the summaries in terms of their fluency, redundancy level and overall quality, on a 3-point Likert scale from -1 to 1. Higher scores mean better quality. Each summary was rated by five independent workers. We use the averaged overall quality score for each summary as the ground-truth human judgement.

RL-based summarisation systems assume that the summaries ranked highly by the reward function (e.g. ROUGE) are indeed "good" in terms of human judgement. We define *good* summaries as follows: a summary y for a news article x is good if (**i**) the average human judgement score for y is ≥ 0.5, and (**ii**) among the five summaries for x, y is ranked within the top two. To study to what extent the above assumption is true, we not only measure the summary-level correlation (Spearman's ρ and Pearson's r) between the reward function and human judgements, but also count how many good summaries identified by the reward function are indeed good (*G-Pre*), and how many indeed good summaries are identified by the reward function (*G-Rec*). We normalise the reward scores and the human judgements to the same range.

From Table 1, we find that all metrics we consider have low correlation with the human judgement. More importantly, their G-Pre and G-Rec scores are all below .50, which means that more than half of the good summaries identified by the metrics are actually not good, and more than 50%

[1] https://tac.nist.gov/data/

Metric	ρ	r	G-Pre	G-Rec
ROUGE-1	.290	.304	.392	.428
ROUGE-2	.259	.278	.408	.444
ROUGE-L	.274	.297	.390	.426
ROUGE-SU4	.282	.279	.404	.440
BLEU-1	.256	.281	.409	.448
BLEU-2	.301	.312	.411	.446
BLEU-3	.317	.312	.409	.444
BLEU-4	.311	.307	.409	.446
BLEU-5	.308	.303	.420	.459
METEOR	.305	.285	.409	.444
InferSent-Cosine	**.329**	**.339**	.417	.460
BERT-Cosine	.312	.335	**.440**	**.484**

Table 1: Quality of reward metrics. G-Pre and G-Rec are the precision and recall rate of the "good" summaries identified by the metrics, resp. All metrics here require reference summaries. We perform stemming and stop words removal as preprosessing, as they help increase the correlation. For InferSent, the embeddings of the reference/system summaries are obtained by averaging the embeddings of the sentences therein.

of the indeed good summaries cannot be identified by the considered metrics. Hence, hill-climbing on these metrics can hardly guide the RL agents to generate genuinely high-quality summaries. These observations clearly motivate the need for better rewards. Next, we formally formulate the reward learning task.

4 Problem Formulation

We focus on reward learning for single-document summarisation in this work, formulated as follows. Let $\mathcal{X}$ be the set of all input documents. For $x \in \mathcal{X}$, let Y_x be the set of all summaries for x that meet the length requirement. A *reward function* $R(x, y; \theta)$ measures the quality of summary y for document x, where θ stands for all parameters for R. Note that human judgements can be viewed as the ground-truth reward function, which we denote as R^* henceforth. At training time, suppose we have access to $\overline{\mathcal{X}} \subseteq \mathcal{X}$ documents and N summaries for each $x \in \overline{\mathcal{X}}$, denoted by $\overline{Y}_x = \{y_x^1, \cdots, y_x^N\} \subseteq Y_x$. Hence, we have $|\overline{\mathcal{X}}| \times N$ summaries at training time, and our training set includes the R^* scores for these summaries: $\bigcup_{x \in \overline{\mathcal{X}}} \{R^*(x, y_x^1), \cdots, R^*(x, y_x^N)\}$. Our target is to learn a reward function R that is as "close" to R^* as possible. Depending on the definition of "close", we explore two loss functions for reward learning, detailed below.

Regression loss. We first consider reward learning as a regression problem, by measuring the "closeness" between R and R^* by their mean squared errors:

$$\mathcal{L}^{MSE}(\theta) = \frac{1}{|\overline{\mathcal{X}}| \cdot N} \sum_{x \in \overline{\mathcal{X}}} \sum_{i=1}^{N} [R^*(x, y_x^i) - R(x, y_x^i; \theta)]^2. \quad (1)$$

Cross-entropy loss. An alternative definition of "closeness" is to measure the "agreement" between R and R^*, i.e. for a pair of summaries, whether R and R^* prefer the same summary. For two summaries $y_x^i, y_x^j \in \overline{Y}_x$, we estimate the likelihood that R prefers y_x^i over y_x^j as

$$P(y_x^i \succ y_x^j) = \frac{\exp(r^i)}{\exp(r^i) + \exp(r^j)}, \quad (2)$$

where $r^i = R(x, y_x^i; \theta)$, $r^j = R(x, y_x^j; \theta)$. Note that for each $x \in \overline{\mathcal{X}}$, we have N summaries available in $\overline{Y}_x$. Hence we can construct $N \cdot (N-1)/2$ pairs of summaries for each input x. Our target is to minimise the "disagreement" between R^* and R on the $|\overline{\mathcal{X}}| \cdot N \cdot (N-1)/2$ pairs of summaries, formally defined as the cross-entropy loss below:

$$\mathcal{L}^{CE}(\theta) = -\frac{1}{|\overline{\mathcal{X}}| N(N-1)/2} \sum_{x \in \overline{\mathcal{X}}} \sum_{i=1}^{N} \sum_{j>i}^{N} \{ \mathbb{1}[R^*(x, y_x^i) > R^*(x, y_x^j)] \log P(y_x^i \succ y_x^j) + \mathbb{1}[R^*(x, y_x^j) > R^*(x, y_x^i)] \log P(y_x^j \succ y_x^i) \}, \quad (3)$$

where $\mathbb{1}$ is the indicator function. Next, we will introduce our reward learning model that minimises the losses defined in Eq. (1) and (3).

5 Reward Learning Model

We explore two neural architectures for $R(x, y; \theta)$: *Multi-Layer Perceptron (MLP)* and *Similarity-Redundancy Matrix (SimRed)*, detailed below.

5.1 MLP

A straightforward method for learning $R(x, y_x; \theta)$ is to encode the input document x and summary y as two embeddings, and feed the concatenated embedding into a single-layer MLP to minimise the loss functions Eq. (1) and (3). We consider three text encoders to vectorise x and y_x. In supplementary material, we provide figures to further illustrate the architectures of these text encoders.

CNN-RNN. We use convolutional neural networks (CNNs) to encode the sentences in the input text, and feed the sentence embeddings into an LSTM to generate the embedding of the whole input text. In the CNN part, convolutions with different filter widths are applied independently as in (Kim, 2014). The most relevant features are selected afterwards with max-over-time pooling. In line with Narayan et al. (2018b), we reverse the order of sentence embeddings before feeding them into the LSTM. This encoder network yields strong performance on summarisation and sentence classification tasks (Narayan et al., 2018a,b).

PMeans-RNN. PMeans is a simple yet powerful sentence encoding method (Rücklé et al., 2018). PMeans encodes a sentence by computing the *power means* of the embeddings of the words in the sentence. PMeans uses a parameter p to control the weights for each word embedding: with $p = 1$, each word element is weighted equally, and 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. The output of PMeans is passed to an LSTM to produce the final document embedding. Note that only the LSTM is trainable; the p value is decided by the system designer.

BERT. We use the pre-trained BERT-Large-Cased model to encode news articles and summaries. The hidden state of the final layer that corresponds to the first token (i.e. "[CLS]") is taken as embedding. Note that the pre-trained BERT models can only encode texts with at most 512 tokens. In line with Alberti et al. (2019), we therefore use a sliding window approach with the offset size of 128 tokens to encode overlength summaries and news articles. We do not fine-tune the BERT model because our dataset is relatively small (only 2,500 summaries and 500 news articles), and the sliding-window of BERT requires much resources to fine-tune.

5.2 Similarity-Redundancy Matrix (SimRed)

Good summaries should be more *informative* (i.e. contain information of higher importance from the input documents) and less *redundant* than bad summaries. Based on this intuition, we propose the SimRed architecture, which explicitly measures the informativeness and redundancy of summary y_x for document x. SimRed maintains a *Similarity* matrix and a *Redundancy* matrix. In the Similarity matrix, cell (i, j) is the cosine similarity between the embeddings of the ith sentence in summary y_x and the jth sentence in document x. In the Redundancy matrix, each cell contains the square of the cosine similarity of a pair of sentences in summary y_x. We use the average over the Similarity matrix cells to measure the informativeness of y_x, the average over the Redundancy matrix cells to measure the redundancy, and compute the weighted sum of these two averaged values to yield the reward R:

$$R_{\text{SimRed}}(y_x, x) = \frac{\alpha}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} \cos(s_i, d_j) - \frac{1-\alpha}{N(N-1)/2} \sum_{k=1}^{N} \sum_{l>k}^{N} (\cos(s_k, s_l))^2, \quad (4)$$

where $s_i, i = 1, \cdots, N$ indicates the embedding of the ith sentence in summary y_x, and $d_j, j = 1, \cdots, M$ indicates the embedding of the jth sentence in document x. The sentence embeddings are generated using CNN, PMeans and BERT as described in §5.1. Because PMeans does not have trainable parameters and BERT is kept fixed, we put a trainable layer on top of them.

6 Reward Quality Evaluation

Experimental Setup. We perform 5-fold cross-validation on the 2,500 human summaries (described in §3) to measure the performance of our reward R. In each fold, the data is split with ratio 64:16:20 for training, validation and test.

The parameters of our model are detailed as follows, decided in a pilot study on one fold of the data split. The CNN-RNN encoder (see §5.1) uses filter widths 1-10 for the CNN part, and uses a unidirectional LSTM with a single layer whose dimension is 600 for the RNN part. For PMeans, we obtain sentence embeddings for each $p \in \{-\infty, +\infty, 1, 2\}$ and concatenate them per sentence. Both these two encoders use the pre-trained GloVe word embeddings (Pennington et al., 2014). On top of these encoders, we use an MLP with one hidden ReLU layer and a linear activation at the output layer. For the MLP that uses CNN-RNN and PMeans-RNN, the dimension of its hidden layer is 100, while for the MLP with BERT as encoder, the dimension of the hidden layer is 1024. As for SimRed, we set α (see Eq. (4)) to be 0.85. The trainable layer on top of BERT and PMeans – when used with SimRed – is a single

layer perceptron whose output dimension is equal to the input dimension.

Reward Quality. Table 2 shows the quality of different reward learning models. As a baseline, we also consider the feature-rich reward learning method proposed by Peyrard and Gurevych (2018) (see §2). MLP with BERT as encoder has the best overall performance. Specifically, BERT+MLP+Pref significantly outperforms ($p < 0.05$) all the other models that do not use BERT+MLP, as well as the metrics that rely on reference summaries (see Table 1). P-values between each pair of metrics/rewards can be found in the supplementary material. In general, preference loss (Eq. (2)) yields better performance than regression loss (Eq. (1)), because it "pushes" the reward function to provide correct preferences over summaries, which leads to more precise ranking.

Fig. 1 illustrates the distribution of some rewards/metrics for summaries with different human ratings. In the left-most sub-figure in Fig. 1, we find that, for summaries with average human rating 1.0 (the highest human rating; see §3), their average ROUGE-2 scores are similar to those with lower human ratings, which indicates that ROUGE-2 can hardly distinguish the highest-quality summaries from the rest. We make similar observations for InferSent-Cosine and BERT-Cosine. BERT+MLP+Pref provides higher scores to summaries with higher human ratings (the right-most sub-figure), although it does not use reference summaries. This explains the strong G-Pre and G-Rec scores of BERT+MLP+Pref. The distributions of the other metrics/rewards can be found in the supplementary material. Next, we use the reward learned by BERT+MLP+Pref to train some RL-based summarisation systems.

7 RL-based Summarisation with Learned Rewards

We consider two RL-based summarisation systems, an extractive system *NeuralTD* (Gao et al., 2019), and an abstractive system *ExtAbsRL* (Chen and Bansal, 2018). Note that ExtAbsRL is a cross-input RL while NeuralTD is an input-specific RL (see §2). Our study is performed on the test set of the non-anonymised CNN/DailyMail dataset (See et al., 2017), which includes 11,490 news articles and one reference summary for each article.

7.1 RL-based Summarisation Systems

NeuralTD. NeuralTD is an improved version of the RL-based extractive multi-document summarisation system proposed by Ryang and Abekawa (2012). We briefly recap the original system below. Suppose the RL agent has selected some sentences and has built a draft summary d using the selected sentences. The RL agent maintains a function $V: D \rightarrow \mathbb{R}$, where D denotes the set of all possible draft summaries. $V(d; w)$ estimates the quality and potential of d, where w are the learnable parameters in V. To select which sentence to add to d next, the agent samples sentences $s \in S$ with distribution

$$\pi(s; w) = \frac{\exp[V((d, s); w)]}{\sum_{s' \in S} \exp[V((d, s'); w)]},$$

where S is the set of all sentences in the input document that has not been added to d yet, and (d, s) is the resulting summary of concatenating d and s. The original system proposed by Ryang and Abekawa (2012) models V as a linear function (i.e. $V(d; w) = w \cdot \phi(d)$, where $\phi(d)$ is the vector for draft summary d). NeuralTD instead uses a neural network with multiple hidden layers to approximate V. Gao et al. (2019) show that NeuralTD significantly outperforms the original linear algorithm in multiple benchmark datasets.

In line with (Gao et al., 2019), we use the *delayed rewards* in NeuralTD: a non-zero reward is provided to the agent only when the agent finishes the sentence selection process (i.e. when agent performs the "end-of-construction" action). The assumption underlying this reward scheme is that the reward function can only precisely measure the quality of the summary when the summary is complete. Besides our learned reward, in order to encourage the agent to select the "lead" sentences (i.e. the first three sentences in each news article), we provide the agent with a small extra reward (0.5) for each "lead" sentence the agent chooses to extract. The value for the extra reward (0.5) is decided in a pilot study, in which we manually check the quality of the generated summaries with different extra rewards (0.1, 0.3, $\cdots$, 0.9).

ExtAbsRL has an *extractor* to extract salient sentences and an *abstractor* to rephrase the extracted sentences to generate abstractive summaries. The abstractor is a simple encoder-aligner-decoder model with copying mechanism, which

Model	Encoder	*Reg. loss (Eq. (1))* ρ	r	G-Pre	G-Rec	*Pref. loss (Eq. (3))* ρ	r	G-Pre	G-Rec
MLP	CNN-RNN	.311	.340	.486	.532	.318	.335	.481	.524
	PMeans-RNN	.313	.331	.489	.536	.354	.375	.502	.556
	BERT	**.487**	**.526**	**.544**	**.597**	**.505**	**.531**	**.556**	**.608**
SimRed	CNN	.340	.392	.470	.515	.396	.443	.499	.549
	PMeans	.354	.393	.493	.541	.370	.374	.507	.551
	BERT	.266	.296	.458	.495	.325	.338	.485	.533
(Peyrard and Gurevych, 2018)		.177	.189	.271	.306	.175	.186	.268	.174

Table 2: Summary-level correlation of learned reward functions. All results are averaged over 5-fold cross validations. Unlike the metrics in Table 1, all rewards in this table do not require reference summaries.

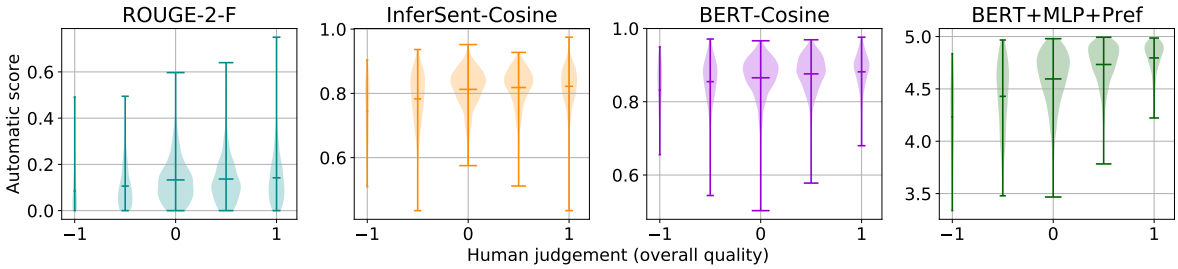

Figure 1: Distributions of some metrics/rewards for summaries with different human ratings. Among the four presented metrics/rewards, only BERT+MLP+Pref (the rightmost sub-figure) does not use reference summaries.

is pre-trained using standard supervised cross-entropy training. The extractor, on the other hand, applies an actor-critic RL algorithm on top of a pointer network. Unlike NeuralTD that uses delayed rewards, ExtAbsRL receives a non-zero reward after adding each new sentence, by computing the ROUGE-L score between the newly added sentence and the corresponding sentence in the reference summary. When the generated summary has more sentences than the reference summary, 0 is given as the reward for the extra sentences. At the final step, ROUGE-1 of the whole generated summary is granted as reward.

We follow the step-wise reward scheme in original ExtAbsRL but, instead of using ROUGE-L to compute the step-wise rewards, we apply our learned reward function to compute the score for the summary with and without the new sentence, and use their difference as reward. Similarly, for the final step reward, we also use our learned reward function. In addition, we force our summariser to stop adding new sentences when the number of tokens in the generated summary is 1.2 times as many as in the reference summary, because we find the abstractive summaries generated by the original ExtAbsRL algorithm are approximately of this length. Since the abstractor in ExtAbsRL is not RL-based, the different reward only influences the extractor.

System	Reward	R-1	R-2	R-L
(Kryscinski et al., 2018)	R-L	40.2	17.4	37.5
(Narayan et al., 2018b)	R-1,2,L	40.0	18.2	36.6
(Chen and Bansal, 2018)	R-L	41.5	18.7	37.8
(Dong et al., 2018)	R-1,2,L	41.5	18.7	37.6
(Zhang et al., 2018)	NA	41.1	18.8	37.5
(Zhou et al., 2018)	NA	41.6	19.0	38.0
(Kedzie et al., 2018)	NA	39.1	17.9	35.9
(ours) NeuralTD	Learned	39.6	18.1	36.5

Table 3: Full-length ROUGE F-scores of some recent RL-based (upper) and supervised (middle) extractive summarisation systems, as well as our system with learned rewards (bottom). R-1/2/L stands for ROUGE-1/2/L. Our system maximises the learned reward instead of ROUGE, hence receives lower ROUGE scores.

7.2 Extractive Summarisation

Table 3 presents the ROUGE scores of our system (NeuralTD+LearnedRewards) and multiple state-of-the-art systems. The summaries generated by our system receive decent ROUGE metrics, but are lower than most of the recent systems, because our learned reward is optimised towards high correlation with human judgement instead of ROUGE metrics.

To measure the human ratings on the gener-

	Ours	Refresh	ExtAbsRL
Avg. Human Rating	**2.52**	2.27	1.66
Best%	**70.0**	33.3	6.7

Table 4: Human evaluation on extractive summaries. Our system receives significantly higher human ratings on average. "Best%": in how many percentage of documents a system receives the highest human rating.

Reward	R-1	R-2	R-L	Human	Pref%
R-L (original)	40.9	17.8	38.5	1.75	15
Learned (ours)	39.2	17.4	37.5	**2.20**	**75**

Table 5: Performance of ExtAbsRL with different reward functions, measured in terms of ROUGE (center) and human judgements (right). Using our learned reward yields significantly ($p = 0.0057$) higher average human rating. "Pref%": in how many percentage of documents a system receives the higher human rating.

ated summaries, we invited five users to read and rate the summaries from three systems: NeuralTD+LearnedReward, the Refresh system (Narayan et al., 2018b) and the extractive version of the ExtAbsRL system, which only extracts salient sentences and does not apply sentence rephrasing. We selected Refresh and ExtAbsRL because they both have been reported to receive higher human ratings than the strong system proposed by See et al. (2017).

We randomly sampled 30 news articles from the test set in CNN/DailyMail, and asked the five participants to rate the three summaries for each article on a 3-point Likert scale from 1 to 3, where higher scores mean better overall quality. We asked them to consider the *informativeness* (whether the summary contains most important information in the article) and *conciseness* (whether the summary is concise and does not contain redundant information) in their ratings.

Table 4 presents the human evaluation results. summaries generated by NeuralTD receives significantly higher human evaluation scores than those by Refresh ($p = 0.0088$, double-tailed t-test) and ExtAbsRL ($p \ll 0.01$). Also, the average human rating for Refresh is significantly higher ($p \ll 0.01$) than ExtAbsRL, despite receiving significantly higher ROUGE scores than both Refresh and NeuralTD (see Table 3). We find that the summaries generated by ExtAbsRL include more tokens (94.5) than those generated by Refresh (83.4) and NeuralTD (85.6). Sun et al. (2019) recently show that, for summaries whose lengths are in the range of 50 to 110 tokens, longer summaries receive higher ROUGE-F1 scores. We believe this is the reason why ExtAbsRL has higher ROUGE scores. On the other hand, ExtAbsRL extracts more redundant sentences: four out of 30 summaries by ExtAbsRL include redundant sentences, while Refresh and NeuralTD do not generate summaries with two identical sentences therein. Users are sensitive to the redundant sentences in summaries: the average human rating for redundant summaries is 1.2, lower than the average rating for the other summaries generated by ExtAbsRL (1.66). To summarise, by using our learned reward function in training an extractive RL summariser (NeuralTD), the generated summaries receive significantly higher human ratings than the state-of-the-art systems.

7.3 Abstractive Summarisation

Table 5 compares the ROUGE scores of using different rewards to train the extractor in ExtAbsRL (the abstractor is pre-trained, and is applied to rephrase the extracted sentences). Again, when ROUGE is used as rewards, the generated summaries have higher ROUGE scores.

We randomly sampled 20 documents from the test set in CNN/DailyMail and asked three users to rate the quality of the two summaries generated with different rewards. We asked the users to consider not only the informativeness and conciseness of summaries, but also their grammaticality and faithfulness (whether the information in the summary is consistent with that in the original news). It is clear from Table 5 that using the learned reward helps the RL-based system generate summaries with significantly higher human ratings. Furthermore, we note that the overall human ratings for the abstractive summaries are lower than the extractive summaries (compared to Table 4). Qualitative analysis suggests that the poor overall rating may be caused by occasional information inconsistencies between a summary and its source text: for instance, a sentence in the source article reads *"after Mayweather was almost two hours late for his workout , Pacquiao has promised to be on time"*, but the generated summary outputs *"Mayweather has promised to be on time for the fight"*. High redundancy is another reason for the low human ratings: ExtAbsRL generates six summaries with redundant sentences when applying ROUGE-L as reward, while the number drops to

two when the learned reward is applied.

8 Conclusion & Discussion

In this work, we focus on Reinforcement Learning (RL) based summarisation, and propose a reward function directly learned from human ratings on summaries' overall quality. Our reward function only takes the source text and the generated summary as input (i.e. does not require reference summaries), and correlates significantly better with human judgements than existing metrics (e.g. ROUGE and METEOR, which require reference summaries). We use our learned reward to train both extractive and abstractive summarisation systems. Experiments show that the summaries generated from our learned reward outperform those by the state-of-the-art systems, in terms of human judgements. Considering that our reward is learned from only 2,500 human ratings on 500 summaries, while the state-of-the-art systems require two orders of magnitude (287k) more document-reference pairs for training, this work clearly shows that reward learning plus RL-based summarisation is able to leverage a relatively small set of human rating scores to produce high-quality summaries.

For future work, we plan to test our method in other summarisation tasks (e.g. multi-document summarisation) and downstream tasks of summarisation (e.g. investigating whether users can correctly answer questions by reading our summaries instead of the original documents). Also, we believe the "learning reward from human judgements" idea has potential to boost the performance of RL in other natural language generation applications, e.g. translation, sentence simplification and dialogue generation.

Acknowledgements

This work has been supported by the German Research Foundation (DFG), as part of the QA-EduInf project (GU 798/18-1 and RI 803/12-1) and through the German-Israeli Project Cooperation (DIP, DA 1600/1-1 and GU 798/17-1).

References

Chris Alberti, Kenton Lee, and Michael Collins. 2019. A BERT Baseline for the Natural Questions. *arXiv e-prints*.

Kristjan Arumae and Fei Liu. 2019. Guiding extractive summarization with question-answering rewards. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2566–2577, Minneapolis, USA.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal.

Arun Chaganty, Stephen Mussmann, and Percy Liang. 2018. The price of debiasing automatic metrics in natural language evalaution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Volume 1: Long Papers*, pages 643–653, Melbourne, Australia.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Volume 1: Long Papers*, pages 675–686, Melbourne, Australia.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota.

Yue Dong, Yikang Shen, Eric Crawford, Herke van Hoof, and Jackie Chi Kit Cheung. 2018. Banditsum: Extractive summarization as a contextual bandit. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3739–3748, Brussels, Belgium.

Yang Gao, Christian M. Meyer, and Iryna Gurevych. 2018. APRIL: interactively learning to summarise by combining active preference learning and reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4120–4130, Brussels, Belgium.

Yang Gao, Christian M. Meyer, and Iryna Gurevych. 2019. Preference-based Interactive Multi-Document Summarisation. *arXiv e-prints*.

Yang Gao, Christian M. Meyer, Mohsen Mesgar, and Iryna Gurevych. 2019. Reward learning for efficient reinforcement learning in extractive document summarisation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 2350–2356, Macao, China.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719, New Orleans, Louisiana, USA.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015*, pages 1693–1701, Montreal, Quebec, Canada.

Chris Kedzie, Kathleen R. McKeown, and Hal Daumé III. 2018. Content selection in deep learning models of summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1818–1828, Brussels, Belgium.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751, Doha, Qatar.

Julia Kreutzer, Artem Sokolov, and Stefan Riezler. 2017. Bandit structured prediction for neural sequence-to-sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Volume 1: Long Papers*, pages 1503–1513, Vancouver, Canada.

Julia Kreutzer, Joshua Uyheng, and Stefan Riezler. 2018. Reliability and learnability of human bandit feedback for sequence-to-sequence reinforcement learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Volume 1: Long Papers*, pages 1777–1788, Melbourne, Australia.

Wojciech Kryscinski, Romain Paulus, Caiming Xiong, and Richard Socher. 2018. Improving abstraction in text summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1808–1817, Brussels, Belgium.

Alon Lavie and Michael J. Denkowski. 2009. The meteor metric for automatic evaluation of machine translation. *Machine Translation*, 23(2-3):105–115.

Chin-Yew Lin. 2004a. Looking for a few good metrics: Rouge and its evaluation. In *NTCIR Workshop*.

Chin-Yew Lin. 2004b. ROUGE: A package for automatic evaluation of summaries. In *ACL Workshop "Text Summarization Branches Out"*.

Annie Louis and Ani Nenkova. 2013. Automatically assessing machine summary content without a gold standard. *Computational Linguistics*, 39(2):267–300.

Shashi Narayan, Ronald Cardenas, Nikos Papasarantopoulos, Shay B. Cohen, Mirella Lapata, Jiangsheng Yu, and Yi Chang. 2018a. Document modeling with external attention for sentence extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Volume 1: Long Papers*, pages 2020–2030, Melbourne, Australia.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018b. Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1747–1759, New Orleans, Louisiana, USA.

Andrew Y. Ng and Stuart J. Russell. 2000. Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 663–670, Stanford University, Stanford, CA, USA.

Jekaterina Novikova, Ondrej Dusek, Amanda Cercas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for NLG. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2241–2252, Copenhagen, Denmark.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, PA, USA.

Ramakanth Pasunuru and Mohit Bansal. 2018. Multi-reward reinforced summarization with saliency and entailment. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 646–653, New Orleans, Louisiana, USA.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *6th International Conference on Learning Representations, Conference Track Proceedings*, Vancouver, BC, Canada.

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

Processing (EMNLP), pages 1532–1543, Doha, Qatar.

Maxime Peyrard, Teresa Botschen, and Iryna Gurevych. 2017. Learning to score system summaries for better content selection evaluation. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 74–84, Copenhagen, Denmark.

Maxime Peyrard and Iryna Gurevych. 2018. Objective function learning to match human judgements for optimization-based summarization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 654–660, New Orleans, Louisiana, USA.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100, 000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, USA.

Cody Rioux, Sadid A. Hasan, and Yllias Chali. 2014. Fear the REAPER: A system for automatic multi-document summarization with reinforcement learning. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*, pages 681–690, Doha, Qatar.

Andreas Rücklé, Steffen Eger, Maxime Peyrard, and Iryna Gurevych. 2018. Concatenated Power Mean Word Embeddings as Universal Cross-Lingual Sentence Representations. *arXiv e-prints*.

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*, pages 256–265, Jeju Island, Korea.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Volume 1: Long Papers*, pages 1073–1083, Vancouver, Canada.

Simeng Sun, Ori Shapira, Ido Dagan, and Ani Nenkova. 2019. How to Compare Summarizers without Target Length? Pitfalls, Solutions and Re-Examination of the Neural Summarization Literature. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 21–29, Minneapolis, Minnesota.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana, USA.

Xingxing Zhang, Mirella Lapata, Furu Wei, and Ming Zhou. 2018. Neural latent extractive document summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 779–784, Brussels, Belgium.

Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. 2018. Neural document summarization by jointly learning to score and select sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Volume 1: Long Papers*, pages 654–663, Melbourne, Australia.