Sentence Mover's Similarity: Automatic Evaluation for Multi-Sentence Texts

Elizabeth Clark[1*] **Asli Celikyilmaz**[2] **Noah A. Smith**[1,3]
[1]Paul G. Allen School of Computer Science & Engineering, University of Washington
[2]Microsoft Research
[3]Allen Institute for Artificial Intelligence
{eaclark7,nasmith}@cs.washington.edu aslicel@microsoft.com

Abstract

For evaluating machine-generated texts, automatic methods hold the promise of avoiding collection of human judgments, which can be expensive and time-consuming. The most common automatic metrics, like BLEU and ROUGE, depend on exact word matching, an inflexible approach for measuring semantic similarity. We introduce methods based on *sentence mover's similarity*; our automatic metrics evaluate text in a continuous space using word and sentence embeddings. We find that sentence-based metrics correlate with human judgments significantly better than ROUGE, both on machine-generated summaries (average length of 3.4 sentences) and human-authored essays (average length of 7.5). We also show that sentence mover's similarity can be used as a reward when learning a generation model via reinforcement learning; we present both automatic and human evaluations of summaries learned in this way, finding that our approach outperforms ROUGE.

1 Introduction

Automatic text evaluation reduces the need for human evaluations, which can be expensive and time-consuming to collect, particularly when evaluating long, multi-sentence texts. Automatic metrics allow faster measures of progress when training and testing models and easier development of text generation systems.

However, existing automatic metrics for evaluating text are problematic. Due to their computational efficiency, metrics based on word-matching are common, such as ROUGE (Lin, 2004) for summarization, BLEU (Papineni et al., 2002) for machine translation, and METEOR (Banerjee and Lavie, 2005) or CIDER (Vedantam et al., 2015) for image captioning. Nevertheless, these metrics often fail to capture information that has been reworded or reordered from the reference text, as shown in Kilickaya et al. (2017) and Table 1.[1] They have also been found to correlate weakly with human judgments (Liu et al., 2016; Novikova et al., 2017).

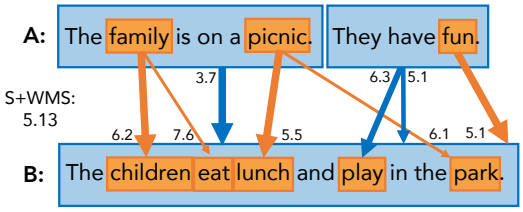


Figure 1: An illustration of S+WMS (a sentence mover similarity metric that uses both word and sentence embeddings) between two documents. This metric finds the minimal cost of "moving" both the word embeddings (orange) and the sentence embeddings (blue) in Document A to those in Document B. An arrow's width is the proportion of the embedding's weight being moved, and its label is the Euclidean distance. Here we show only the highest weighted connections.

To avoid these shortcomings, word mover's distance (WMD; Kusner et al., 2015) can be used to evaluate text in a continuous space using pretrained word embeddings instead of relying on exact word matching. WMD has been used successfully for tasks including image caption evaluation (Kilickaya et al., 2017), automatic essay evaluation (Tashu and Horváth, 2018), and affect detection (Alshahrani et al., 2017). This bag-of-embeddings approach is flexible but fails to reflect the grouping of words and ideas, a shortcoming that becomes more problematic as the length of the document grows.

We modify WMD for evaluating multi-sentence texts by basing the score on sentence embeddings (§3), giving it access to higher-level representa-

[*] Work done while author was at Microsoft Research.

[1]For readability, we scale ROUGE scores by a factor of 100 and sentence mover's metrics by a factor of 1000.

Reference passage. the only thing crazier than a guy in snowbound massachusetts boxing up the powdery white stuff and offering it for sale online ? people are actually buying it . for $ 89 , self-styled entrepreneur kyle waring will ship you 6 pounds of boston-area snow in an insulated styrofoam box – enough for 10 to 15 snowballs , he says .

Summary	ROUGE-L	WMS	SMS	S+WMS
Human summary. a man in suburban boston is selling snow online to customers in warmer states . for $ 89 , he will ship 6 pounds of snow in an insulated styrofoam box .	39.30	57.85	99.98	24.06
Word order. in suburban boston , a man is selling snow online to customers in warmer states . he will ship 6 pounds of snow in an insulated styrofoam box for $ 89 .	31.44 (↓ 20%)	57.85 (=)	99.98 (=)	24.06 (=)
Repetition. a man in suburban boston is selling snow is selling snow online to customers in warmer states in warmer states . for $ 89 , he will ship he will ship 6 pounds 6 pounds of snow in an insulated styrofoam box in a styrofoam box .	35.07 (↓ 11%)	57.31 (↓ 1%)	89.40 (↓ 11%)	22.81 (↓ 5%)

Table 1: A comparison of scores for three different summaries for a reference passage (the first lines of a news article). The human summary has been permuted with its clauses rearranged (Word order) and repeated (Repetition). Word order changes negatively affect ROUGE-L more than repetition; the other metrics are unaffected by word order choices but, to varying degrees, penalize repetition.

tions of the text. We introduce two new metrics: **sentence mover's similarity** (SMS), which relies only on sentence embeddings, and **sentence and word mover's similarity** (S+WMS), which uses word and sentence embeddings, as in Figure 1.

In §4, we find that sentence mover's similarity metrics significantly improve correlation with human evaluations over ROUGE-L (the longest common subsequence variant of ROUGE) and WMD when scoring automatically generated summaries (averaging 3.4 sentences). We also automatically evaluate human-authored essays (averaging 7.5 sentences) and find smaller but significant gains. We compute sentence mover's similarity metrics with type-based embeddings and contextual embeddings and find these results hold regardless of embedding type, with no significant difference caused by the choice of embedding.

Finally, we show in §5 that sentence mover's similarity metrics can also be used when learning to generate text. Generating summaries using reinforcement learning with sentence mover's similarity as the reward results in higher quality summaries than those generated using a ROUGE-L or WMD reward, according to both automatic metrics and human evaluations.

2 Background: Word Mover's Distance

Earth mover's distance (EMD, also known as the Wasserstein metric; Rubner and Guibas, 1998) is a measure of the distance between two probability distributions. Word mover's distance (WMD; Kusner et al., 2015) is a discrete version of EMD that evaluates the distance between two sequences (e.g., sentences, paragraphs, etc.), each represented with relative word frequencies. It combines (1) item similarity[2] on bag-of-word (BOW) histogram representations of text (Goldberg et al., 2018) with (2) word embedding similarity.

For any two documents A and B, WMD is defined as the minimum cost of transforming one document into the other. Each document is represented by the relative frequencies of words it contains, i.e., for the ith word type,

$$d_{A,i} = \text{count}(i)/|A| \tag{1}$$

where $|A|$ is the total word count of document A, and $d_{B,i}$ is defined similarly.

Now let the ith word be represented by $\mathbf{v}_i \in \mathbb{R}^m$, i.e., an m-length embedding,[3] allowing us to define distances between the ith and jth words, denoted $\Delta(i,j)$. V is the vocabulary size. We follow Kusner et al. (2015) and use the Euclidean distance $\Delta(i,j) = \|\mathbf{v}_i - \mathbf{v}_j\|_2$. The WMD is then the solution to the linear program:

$$\text{WMD}(A,B) = \min_{\mathbf{T} \geq \mathbf{0}} \sum_{i=1}^{V} \sum_{j=1}^{V} \mathbf{T}_{i,j} \Delta(i,j) \tag{2a}$$

s.t.

$$\forall i, \sum_{j=1}^{V} \mathbf{T}_{i,j} = d_{A,i}, \tag{2b}$$

[2]The similarity can be defined as cosine, Jaccard, Euclidean, etc.

[3]Our evaluation scores depend on pretrained word embeddings, which can be type-based or contextual. Our experiments consider both; see §4 and §5. When using contextual embeddings, we treat each token as its own type, as each word will have a different embedding depending on its context.

$$\forall j, \sum_{i=1}^{V} \mathbf{T}_{i,j} = d_{B,j} \tag{2c}$$

$\mathbf{T} \in \mathbb{R}^{V \times V}$ is a nonnegative matrix, where each $\boldsymbol{T}_{i,j}$ denotes how much of word i (across all its tokens) in A is assigned to tokens of word j in B, and the constraints ensure the flow of a given word cannot exceed its weight. Specifically, WMD ensures that the entire outgoing flow from word i equals $d_{A,i}$, i.e., $\sum_j \boldsymbol{T}_{i,j} = d_{A,i}$. Additionally, the amount of incoming flow to word j must match $d_{B,j}$, i.e., $\sum_i \boldsymbol{T}_{i,j} = d_{B,j}$. Following the example of Kilickaya et al. (2017), we transform WMD into a similarity (WMS):

$$\text{WMS}(A, B) = \exp(-\text{WMD}(A, B)) \tag{3}$$

WMS measures two documents' similarity by minimizing the total distance to move words between two documents, combining the strengths of BOW and word embedding-based similarity metrics. In Figure 1, WMS would calculate the cost of moving from Document A to Document B using only the word embeddings, denoted in orange. WMS is symmetric, and $\text{WMS}(A, A) = 1$ when word embeddings are deterministic.

Empirically, WMD has improved the performance of NLP tasks (see §6), specifically sentence-level tasks, such as image caption generation (Kilickaya et al., 2017) and natural language inference (Sulea, 2017). However, its cost grows prohibitively as the length of the documents increases, and the BOW approach can be problematic when documents become large as the relation between sentences is lost. By only measuring word distances, the metric cannot capture information conveyed by the grouping of words, for which we need higher-level document representations (Dai et al., 2015; Wu et al., 2018).

3 Sentence Mover's Similarity Metrics

We modify WMS to measure the similarity between two documents using sentence embeddings, which we call a sentence mover's similarity approach. We introduce two new metrics: Sentence Mover's Similarity (SMS) and Sentence and Word Mover's Similarity (S+WMS). SMS replaces the word embeddings in WMS with sentence embeddings (§3.1), while S+WMS combines the two metrics and uses both word and sentence embeddings (§3.2). Our code (an extension of an existing WMD implementation[4]) and datasets are publicly available.[5]

3.1 Sentence Mover's Similarity

Sentence Mover's Similarity (SMS) performs the same linear optimization problem in Eq. 2a as WMS, except now each document is represented as a bag of sentence embeddings rather than a bag of word embeddings. In Figure 1, SMS considers only the sentence embeddings, denoted in blue.

To get the representation of a sentence in a document, we combine the sentence's word embeddings. Sentence representations based on averaging or pooling word embeddings perform competitively on tasks including sentence classification, recognizing textual entailment, and paraphrase detection (Conneau and Kiela, 2018). We use sentence representations that are the average of their word embeddings, as this approach outperformed pooling methods in preliminary results.

While in WMS word embeddings are weighted according to their frequency in the document (see Eq. 1), SMS weights each sentence embedding by the number of words ($|A|$) it contains.[6] So a sentence i in document A will receive a weight of:

$$d_{A,i} = |i|/|A| \tag{4}$$

We solve the same linear program, Eq. 1, by calculating the cumulative distance of moving a document's sentences to match another document. Now the vocabulary is the set of sentences in the documents instead of the words, as in Figure 2.

3.2 Sentence and Word Mover's Similarity

Sentence and Word Mover's Similarity (S+WMS) combines WMS and SMS and represents each document as a collection of both words and sentences. Each document is now a bag of both word and sentence embeddings (as seen in Figure 1), where each word embedding is weighted according to its frequency and each sentence embedding is weighted according to its length. Now the bag of words and sentences representing document A is normalized by $2|A|$, so that:

[4] https://github.com/src-d/wmd-relax

[5] https://github.com/eaclark07/sms

[6] Preliminary results showed count-based sentence weightings performed better than uniform weightings. Other weighting options, such as frequency-based weighting as done in BERTScore (Zhang et al., 2019), are a direction for extending this work.

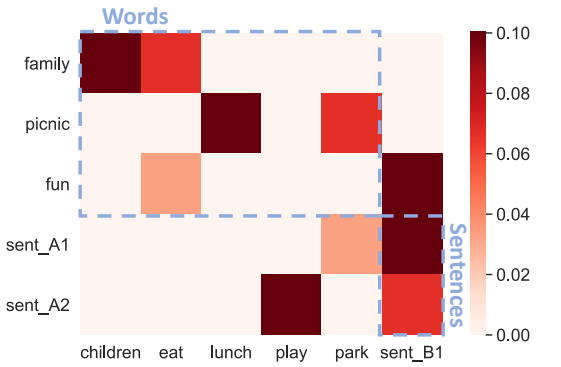


Figure 2: The S+WMS $\boldsymbol{T}$ matrix for documents A and B from Figure 1 (with empty rows/columns removed). Contrarily, WMS's $\boldsymbol{T}$ matrix only maps between words and has the dimensions of the dashed region labeled "Words," and SMS's maps between sentences in the shape of the dashed region "Sentences." Best viewed in color.

$$d_{A,i} = \begin{cases} count(i)/2|A|, & \text{if } i \text{ is a word} \\ |i|/2|A|, & \text{if } i \text{ is a sentence} \end{cases} \quad (5)$$

As in WMS and SMS, the same linear program in Eq. 1 is solved, this time calculating the cumulative distance of moving both a document's words and sentences to match another document. The vocabulary is the set of sentences and words in the documents (see Figure 2). The sentence embeddings are treated the same as word embeddings in the optimization; the only difference is their length-based weights.

This means a sentence embedding can be mapped to a word embedding (e.g., "They have fun." maps to "play" in Figure 1) or vice versa. It also means that a sentence's words do not have to move to the same word or sentence embedding(s) that their sentence moves to (as seen in Figure 1); a sentence in document A could be transported to an embedding in document B and have none of its words moved to the same embedding. More constraints could be introduced to further control the flow between documents, which we leave to future work.

4 Intrinsic Evaluation

To test the performance of the SMS and S+WMS metrics, we first examine their usefulness as evaluation metrics. (In §5, we evaluate their performance as cost functions for an extrinsic task, abstractive summarization.)

We measure the correlations between the scores assigned to texts by various automatic metrics (ROUGE-L, WMS, SMS, S+WMS) and the scores assigned by human judges. We are interested in *multi-sentence* texts, both machine- and human-generated. Therefore, we consider subsets of two corpora that have been judged by humans: a collection of automatically generated summaries of articles in the CNN/Daily Mail news dataset (alongside reference summaries; see Section 4.1; Chaganty et al., 2018; Hermann et al., 2015; Nallapati et al., 2016) and student essays from the Hewlett Foundation's Automated Student Assessment Prize (Section 4.2).[7] Statistics describing the datasets are in A.1.

Because the word and sentence mover's similarity metrics are based on pretrained representations, we explore the effect of varying the word embedding method. We present results for two different types of word embeddings: GloVe embeddings (Pennington et al., 2014) and ELMo embeddings[8] (Peters et al., 2018; Gardner et al., 2018). We obtain GloVe embeddings, which are type-based, 300-dimensional embeddings trained on Common Crawl,[9] using spaCy,[10] while the ELMo embeddings are character-based, 1,024-dimensional, contextual embeddings trained on the 1B Word Benchmark (Chelba et al., 2013). We use ELMo to embed each sentence, which produces three vectors for each word, one from each layer of the model. We average the vectors to get a single embedding for each word in the sentence.

All correlations are Spearman correlations (Elliott and Keller, 2014; Kilickaya et al., 2017), and significance in the improvement between two metrics' correlations with human judgment is calculated using the Williams (1959) significance test.[11]

4.1 Summaries Dataset Evaluation

To understand how the sentence mover's similarity metrics evaluate automatically generated text, we use the subset of the CNN/Daily Mail dataset for which Chaganty et al. (2018) collected human annotations. Annotators evaluated summaries (generated with four different neural models) on a scale

[7] https://www.kaggle.com/c/asap-eas
[8] https://allennlp.org/elmo
[9] http://commoncrawl.org/the-data/
[10] https://spacy.io/models/en#en_core_web_md
[11] https://github.com/ygraham/nlp-williams

	Summaries		Essays	
ROUGE-L	0.117		0.441	
	GloVe	ELMo	GloVe	ELMo
WMS	**0.180	**0.160	0.429	0.443
SMS	****0.258**	****0.253**	0.457	0.451
S+WMS	**0.214	**0.204	***0.488**	***0.490**

Table 2: Spearman correlation of metrics with human evaluations. Asterisks indicate significant improvement over ROUGE-L, with (*) for $p < 0.05$ and (**) for $p < 0.01$.

from –1 to 1. We consider the subset of summaries scored by two or more judges, taking the average to be the summary's score. The automatic evaluation metrics score each generated summary's similarity to the human-authored reference summary from the CNN/Daily Mail dataset.

Table 2 shows each metric's correlation with the human judgments. SMS correlates best with human judgments, and both sentence-based metrics outperform ROUGE-L and WMS. We find that the difference between GloVe and ELMo's scores is not significant.[12]

Discussion Two examples of generated summaries and their scores are shown in Table 3. Because the scores cannot be directly compared between metrics, we distinguish scores that are in the top quartile for their metric (i.e., the highest rated) and in the bottom quartile (i.e., the lowest rated).

The first example in Table 3 is highly rated by metrics using word and sentence embeddings, but judged to be a poor summary by ROUGE-L because information is reworded and reordered from the reference. For example, the phrase "*asked for medical help*" is worded as "*sought medical attention*" in the hypothesis summary. Nevertheless, exact word matching can be important for ensuring factual correctness. While the generated hypothesis summary states "*six officers have been suspended with pay*", the reference states they were actually "*suspended without pay.*"

The second example, which was generated with a seq2seq model, was one of the best summaries according to ROUGE-L but one of the worst according to SMS and S+WMS. It also received low human judgments, most likely due to its nonsensical repetitions. While the short, repeated phrases like "*three different flavours*" match the reference summary well enough to score well with ROUGE-L, the overall sentence representations are distant from those in the reference summary, resulting in low SMS and S+WMS scores.

4.2 Essays Dataset Evaluation

To test the metrics on human-authored text, we use a dataset of graded student essays that consists of responses to standardized test questions for tenth graders. We use a subset of Question #3 from the exam, which asks the test-taker to synthesize information from a reading passage, where student responses contain 5–15 sentences. Graders assigned the student-authored responses with scores ranging from 0 to 3. For the reference essay, we use a top-scoring sample essay, which the graders had access to as a reference while assigning scores. The full reference essay is in A.2.

Table 2 shows the correlation of each metric with the evaluators' scores. As in the summarization task, SMS outperforms both ROUGE-L and WMS. However, in this case, having the sentence representations in the metric gives the best result, with S+WMS correlating best with human scores, significantly better than ROUGE-L. This is consistent across embedding type; once again, the choice of embedding does not create a significant difference between the sentence mover's metrics.[13]

Discussion Aside from the length of the text, the Essays dataset presents the metrics with several challenges not found in the Summaries dataset. For example, the dataset contains a large number of spelling mistakes, due to both author misspellings and errors in the transcription process. One essay begins, "The setting of the story had *effected* the *cycle's becuse* if it was *sub earbs* he could have *stoped any where* and got water ..."

The tone and style of the essay can also vary from the reference essay. (For example, the author of Sample #3 in A.2 ends their essay by reflecting on how they would respond in the protagonist's place.) Embedding-based metrics may be more forgiving to deviations in writing style from the reference essay, such as the use of first person.

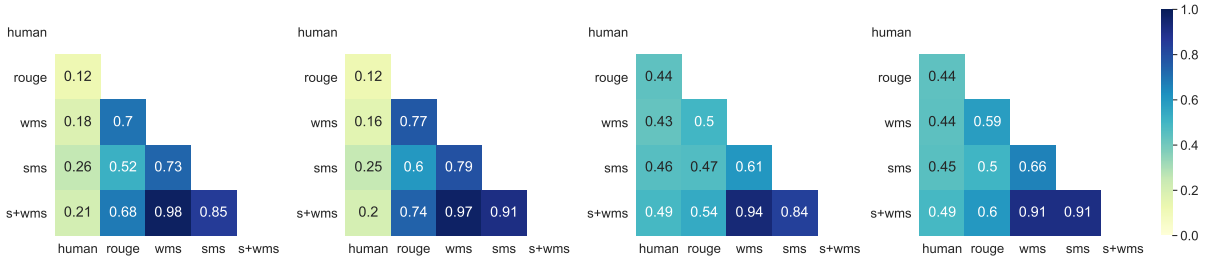
While Table 2 indicates sentence mover's similarity metrics significantly improve correlation with human judgments over standard methods, there is still enough disagreement that we believe automatic metrics should not replace human evaluations. Rather, they should complement human evaluations as an automatic proxy that can be used

[12] Williams test: $p = 0.35$ (SMS) and $p = 0.16$ (S+WMS)

[13] Williams test: $p = 0.33$ (SMS) and $p = 0.46$ (S+WMS)

Samples	Summaries	Metric	Score
Sample #1	**Reference.** Freddie Gray, who is black, asked for medical help but was denied during 00-minute police car ride, eventually paramedics were called. Deputy police commissioner Kevin Davis conceded their failure. But chief commissioner refuses to resign over the death. Six officers are suspended without pay during an investigation.	Human ROUGE-L	0.00 *12.44*
	Hypothesis. Baltimore Police Commissioner Anthony Batts ruled out his resignation despite that fact that his deputy admitted they should have sought medical attention for Freddie Gray. Six officers have been suspended with pay as local police and federal authorities investigate. Commissioner Anthony Batts has ruled out the possibility of his resignation.	WMS SMS S+WMS	**21.41** **128.91** **47.89**
Sample #2	**Reference.** Choc on Choc's chocolates come in three different flavours. The face of each politician is emblazoned on milk Belgium chocolate bars. Cameron's has blueberries, Clegg is honeycomb and Miliband is raspberry.	Human ROUGE-L	-0.5 **34.57**
	Hypothesis. UNK lollies on 273 invalid chocolates come in three different flavours. Contains three different flavours - the colours associated with each leader. David Cameron, Nick Clegg, Nick Clegg, Nick Clegg and David Cameron.	WMS SMS S+WMS	5.08 *51.39* *12.25*

Table 3: Two examples from the Summaries dataset along with the scores they received (using GloVe) comparing reference (human summary) to hypothesis (model generated summary). Scores that are in the top quartile for a given metric are in green and **bold**. Scores in the bottom quartile are in red and *italics*. Human scores range from –1 to 1. Please see A.2 for details.



(a) Summaries with GloVe embeddings (b) Summaries with ELMo embeddings (c) Essays with GloVe embeddings (d) Essays with ELMo embeddings

Figure 3: Spearman correlation with each metric and human evaluations using GloVe and ELMo embeddings on the Summaries and Essays datasets. (Best viewed in color.)

for intermediate evaluation and as a reward signal when learning, as we show in §5.

5 Extrinsic Evaluation

In addition to automatically evaluating text, we can also use sentence mover's metrics as rewards while learning text generation models. To demonstrate this, we train an encoder-decoder model on the CNN/Daily Mail dataset to generate summaries using reinforcement learning (RL). Instead of maximizing likelihood, policy gradient RL methods can directly optimize discrete target evaluation metrics that are non-differentiable, such as ROUGE (Paulus et al., 2018; Jaques et al., 2017; Pasunuru and Bansal, 2017; Wu et al., 2016; Celikyilmaz et al., 2018; Edunov et al., 2018). Here, we learn policies to maximize WMS/SMS/S+WMS metrics, guiding the model to learn semantic similarities, while policies trained using ROUGE rely only on word n-gram matches between generated and ground-truth text.

Model We encode the input document using 2-layered bidirectional LSTM networks and a 2-layered LSTM network for the decoder. We use the attention mechanism (Bahdanau et al., 2015; See et al., 2017) to force the decoder model to learn to focus (i.e., attend) on specific parts of the input sequence when decoding, instead of relying only on the hidden vector of the decoder's LSTM. We also include pointer networks (See et al., 2017; Cheng and Lapata, 2016), which point to elements of the input sequence at each decoding step.

To train our policy-based generator, we use a mixed training objective that jointly optimizes multiple losses, which we describe below.

MLE Our baseline model uses maximum likelihood training for sequence generation. Given $y^* = \{y_1^*, y_2^*, ..., y_T^*\}$ as the ground-truth summary for a given input document d, we compute the loss as:

$$L_{\text{MLE}} = -\sum_{T=1}^{N} \log p(y_t^* \mid y_1^* \ldots y_{t-1}^*, d) \quad (6)$$

by taking the negative log-likelihood of the target word sequence.

Model Loss w/ Reward Metric	ROUGE-1	ROUGE-2	ROUGE-L	WMS	SMS	S+WMS
MLE+Pgen [1] (no reward)	36.44	15.66	33.42	-	-	-
MLE+Pgen+RL Mixed w/ ROUGE-L [2]	38.01	16.43	35.49	-	-	-
MLE+Pgen+RL+Intra-Attn Mixed w/ ROUGE-L [3]	39.87	15.82	36.90	-	-	-
MLE+Pgen (no reward) (re-trained baseline)	36.95	15.56	34.00	13.02	90.05	32.15
MLE+Pgen+RL Mixed w/ ROUGE-L	37.46	16.10	34.39	13.07	86.48	31.87
MLE+Pgen+RL Mixed w/ WMS	38.17	**16.52**	34.97	14.52	95.68	34.77
MLE+Pgen+RL Mixed w/ SMS	**38.52**	**16.52**	**35.33**	**15.15**	**96.65**	**35.50**
MLE+Pgen+RL Mixed w/ S+WMS	37.20	15.67	34.15	13.32	91.09	32.64

Table 4: Evaluation on summarization task when various metrics are used as rewards during learning. Columns show average score of each model's generated summaries according to various metrics. Previously reported results (upper block): [1] MLE training with pointer networks (Pgen) (See et al., 2017) ; [2] Mixed MLE and RL training with Pgen (Celikyilmaz et al., 2018), [3] Mixed MLE and RL training with Pgen and intra-decoder attention (Paulus et al., 2018). The lower block reports re-trained baselines and our models with new metrics. **Bold** indicates best among the lower block.

Reinforcement Learning (RL) Loss The decoder generates the summary sequence $\hat{y}$, which is then compared against the ground truth sequence y^* to compute the reward $r(\hat{y})$. Our model learns using a *self-critical training* approach (Rennie et al., 2016), by exploring new sequences and comparing them against the best greedily decoded sequence. For each training example d, we generate two output sequences: $\hat{y}$, which is sampled from the probability distribution at each time step, $p(\hat{y}_t \mid \hat{y}_1 \dots \hat{y}_{t-1}, d)$, and $\tilde{y}$, the baseline output, which is greedily generated by argmax decoding from $p(\tilde{y}_t \mid \tilde{y}_1 \dots \tilde{y}_{t-1}, d)$. Our mixed training objective is then to minimize:

$$L_{\text{RL}} = (r(\tilde{y}) - r(\hat{y})) \sum_{t=1}^{T} \log p(\hat{y}_t \mid \hat{y}_1 \dots \hat{y}_{t-1}, d) \quad (7)$$

It ensures that, with better exploration, the model learns to generate sequences $\hat{y}$ that receive higher rewards than the baseline $\tilde{y}$, increasing the overall reward expectation of the model.

Mixed Loss While training with only MLE loss will learn a better language model, it may not guarantee better results on discrete performance measures such as WMS and SMS. Similarly, optimizing with only RL loss using SMS as a reward may increase the reward gathered at the expense of diminished readability and fluency of the generated summary. A combination of the two objectives can yield improved task specific scores while maintaining a good language model:

$$L_{\text{MIXED}} = \gamma L_{\text{RL}} + (1 - \gamma) L_{\text{MLE}} \quad (8)$$

where γ is a hyperparameter balancing the two objective functions. We pre-train models with MLE loss, and then continue with the mixed loss.

We train four different models on the CNN/Daily Mail dataset using mixed loss (MLE+RL) with ROUGE-L, WMS, SMS, and S+WMS as the reward functions. Training details are in A.3 and A.4.

5.1 Generated Summary Evaluation

We evaluate the generated summaries from each model with ROUGE-L, WMS, SMS, and S+WMS in Table 4. While we include previously reported numbers, we re-trained the mixed loss models using ROUGE-L and use those as our baseline, as previously trained models should be heavily optimized and use more complex networks than ours. For fair comparison, we kept the encoder-decoder network type, structure, hyperparameters, and initialization the same for each model, changing only the reward. We pre-trained an MLE model ("MLE+Pgen (no reward) (re-trained baseline)" in Table 4) and used it to initialize the mixed loss models with different reward functions.

Across all metrics, the models trained using WMS and SMS metrics as the reward outperform models trained with ROUGE-L as the reward function. S+WMS models lag behind ROUGE-L. The SMS model outperforms all other models across all metrics on the abstractive summarization task, consistent with SMS's performance at evaluating summaries in §4.1.

Table 5 shows summaries generated from each of the mixed loss models.

5.2 Human Evaluation

We collected human evaluations for 100 summaries generated by the mixed loss models to compare ROUGE-L as a reward to WMS, SMS, and S+WMS. Amazon Mechanical Turkers chose between two generated summaries, one from the ROUGE-L model and one from WMS, SMS, or

Human Summary	the 69 - year - old collaborated with nbc 's today show to launch a contest for an elvis - obsessed couple to win the ' ultimate wedding ' . the winning duo will get married in the brand new elvis presley 's graceland wedding chapel at the westgate hotel on thursday , april 23 . while she agreed to make an appearance , the woman who wed elvis in 1967 made one thing clear before unveiling the latest wedding chapel to bear his name : no impersonators .
Model	**Generated Summary**
ROUGE-L	priscilla presley will serve as a witness at the first wedding to be held at an all - new chapel of love in las vegas . the 69 - year - old collaborated with nbc 's today show to launch a contest for one elvis - obsessed couple to win the ' ultimate wedding ' . elvis performed more than 830 sold - out shows .
WMS	the 69 - year - old collaborated with nbc 's today show to launch a contest for one elvis - obsessed couple to win the ' ultimate wedding ' . the winning duo – announced next monday – will tie the knot at elvis presley 's graceland wedding chapel inside the westgate hotel on thursday , april 23 .
SMS	priscilla presley will tie the knot at elvis presley 's graceland wedding chapel inside the westgate hotel on thursday , april 23 . the 69 - year - old collaborated with nbc 's today show to launch a contest for one elvis - obsessed couple to win the ' ultimate wedding ' .
S+WMS	priscilla presley will serve as a witness at the first wedding to be held at an all - new chapel of love in las vegas . the 69 - year - old collaborated with nbc 's today show to launch a contest for one elvis - obsessed couple to win the ' ultimate wedding ' .

Table 5: Summaries generated from the mixed MLE+RL loss models with ROUGE-L, WMS, S+WMS, and SMS metrics as rewards, along with the corresponding human-authored reference summary.

	ROUGE-L vs. WMS				ROUGE-L vs. SMS				ROUGE-L vs. S+WMS			
Criteria	ROUGE-L	WMS	=	% ↑	ROUGE-L	SMS	=	% ↑	ROUGE-L	S+WMS	=	% ↑
non-redundancy	76	**122**	102	61%	64	**144**	92	69%	66	**132**	102	66%
coherence	102	**158**	40	60%	83	**170**	47	67%	83	**166**	51	66%
focus	99	**161**	40	61%	79	**174**	47	68%	84	**166**	50	66%
overall	108	**160**	32	59%	85	**179**	36	67%	84	**179**	37	68%

Table 6: Human evaluations on a random subset of 100 summaries. The frequencies from the head-to-head comparison of models trained with ROUGE-L against WMS/SMS/S+WMS are shown. Each summary is evaluated by 3 judges (300 summaries per criteria). '=' indicates no difference. All improvements are statistically significance at $p < 0.001$.

S+WMS. They selected one of the two summaries based on: (1) *non-redundancy*, fewer repeated ideas, (2) *coherence*, clearly expressed ideas, (3) *focus*, ideas free of superfluous details, and (4) *overall*, the summary effectively communicates the article's content. These criteria help evaluate the impact of the metrics used as reward. (Task details are in A.5.)

Results We asked human judges to evaluate the output of the mixed loss model trained with a ROUGE-L reward versus models trained with WMS, SMS, and S+WMS the reward. The results are shown in Table 6.

Human judges significantly prefer summaries produced by models optimized with WMS, SMS, and S+WMS over ROUGE-L. SMS and S+WMS were preferred over ROUGE-L more often than WMS was. There is no significant difference between the evaluations of SMS and S+WMS. Among all other metrics, SMS was rated the highest on the non-redundancy question (69% improvement over the ROUGE-L score), indicating that the model learns to generate summaries that contain less repetition between sentences.

While the SMS model's output was highly-scored by both the automatic and human evaluations, removing word-level scoring does come with a cost, as seen in the example in Table 5. The SMS summary contains a mistake, stating that "*priscilla will tie the knot*" instead of "*serve as a witness*". This issue may be mitigated by a better encoder for the summarization task and better sentence and word representations. As future work, we will investigate summarization models with more complex sentence embeddings and encoder structures (e.g., self-attention models).

6 Related Work

Evaluation has been among the most discussed topics of the natural language generation (NLG) research area (Lapata and Barzilay, 2005; Belz and Reiter, 2006; Reiter and Belz, 2006; Barzilay and Lapata, 2008; Reiter and Belz, 2009; Reiter, 2011; Novikova et al., 2017). There are three main ways to evaluate NLG methods: (1) automatic metrics to compare NLG texts against refer-

ence texts, (2) task-based (extrinsic) evaluation to measure the impact of a NLG system on a downstream task, and (3) human evaluations, which ask people to rate generated texts. In this work we introduce new automatic evaluation metrics for long text generation and evaluation.

Automatic evaluation metrics compare generated text against reference texts using word overlap metrics such as: BLEU (Papineni et al., 2002); ROUGE (Lin, 2004); NIST (Doddington, 2002), a version of BLEU; METEOR (Lavie and Agarwal, 2007), unigram precision and recall; CIDER (Vedantam et al., 2015), the average n-gram cosine similarity; *cosine similarity* between the average word embedding; and WMD, which calculates the word embedding-based "travel cost". Though all have strengths and weaknesses, ROUGE metrics (particularly ROUGE-L) are common for multi-sentence text evaluations. Textual metrics that consider specific qualities in the system outputs, like complexity and diversity, are also used to evaluate NLG systems (Dusek et al., 2019; Hashimoto et al., 2019; Sagarkar et al., 2018; Purdy et al., 2018).

Word mover's distance has recently been used for NLP tasks like learning word embeddings (Zhang et al., 2017; Wu et al., 2018), textual entailment (Sulea, 2017), document similarity and classification (Kusner et al., 2015; Huang et al., 2016; Atasu et al., 2017), image captioning (Kilickaya et al., 2017), document retrieval (Balikas et al., 2018), clustering for semantic word-rank (Zhang and Wang, 2018), and as additional loss for text generation that measures the optimal transport between the generated hypothesis and reference text (Chen et al., 2019). We investigate WMD for multi-sentence text evaluation and generation and introduce sentence embedding-based metrics.

7 Conclusion

We present SMS and S+WMS, sentence mover's similarity metrics for automatically evaluating multi-sentence texts. We find including sentence embeddings in automatic metrics significantly improves scores' correlation with human judgments, both on automatically generated and human-authored texts. The metrics' gain over ROUGE-L is consistent across word embedding types; there is no significant difference between type-based and contextual embeddings. Moreover, we find these metrics can be used to generate text; summaries generated with SMS as a reward are of better quality than ones generated with ROUGE-L, according to both automatic and human evaluations.

Acknowledgments

This research was supported in part by Microsoft Research, a NSF graduate research fellowship, and the DARPA CwC program through ARO (W911NF-15-1-0543). The authors also thank Antoine Bosselut, Dinghan Shen, and Shuai Tang for their feedback, the anonymous reviewers for their useful comments, and the participants who took part in our study.

References

Mohammed Alshahrani, Spyridon Samothrakis, and Maria Fasli. 2017. Word mover's distance for affect detection. *2017 International Conference on the Frontiers and Advances in Data Science*.

Kubilay Atasu, Thomas P. Parnell, Celestine Dünner, Manolis Sifalakis, Haralampos Pozidis, Vasileios Vasileiadis, Michail Vlachos, Cesar Berrospi, and Abdel Labbi. 2017. Linear-complexity relaxed word mover's distance with GPU acceleration. In *IEEE International Conference on Big Data*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Georgios Balikas, Charlotte Laclau, Ievgen Redko, and Massih-Reza Amini. 2018. Cross-lingual document retrieval using regularized Wasserstein distance. *CoRR*, abs/1805.04437.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *IEEvaluation@ACL*.

Regina Barzilay and Mirala Lapata. 2008. Modeling local coherence: An entity-based approach. In *Computational Linguistics*.

Anja Belz and Ehud Reiter. 2006. Comparing automatic and human evaluation of NLG systems. In *EACL*.

Asli Celikyilmaz, Antoine Bosselut, Xiadong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. In *NAACL*.

Arun Tejasvi Chaganty, Stephen Mussmann, and Percy Liang. 2018. The price of debiasing automatic metrics in natural language evaluation. In *ACL*.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, and Phillipp Koehn. 2013. One billion word benchmark for measuring progress in statistical language modeling. In *INTERSPEECH*.

Liqun Chen, Yizhe Zhang, Ruiyi Zhang, Chenyang Tao, Zhe Gan, Haichao Zhang, Bai Li, Dinghan Shen, Changyou Chen, and Lawrence Carin. 2019. Improving sequence-to-sequence learning via optimal transport. In *ICLR*.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *ACL*.

Alexis Conneau and Douwe Kiela. 2018. SentEval: An evaluation toolkit for universal sentence representations. In *LREC*.

Andrew M. Dai, Christopher Olah, and Quoc V. Le. 2015. Document embedding with paragraph vectors. In *NeurIPS Deep Learning Workshop*.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Second International Conference on Human Language Technology Research*.

Ondrej Dusek, Jekaterina Novikova, and Verena Rieser. 2019. Evaluating the state-of-the-art of end-to-end natural language generation: The E2E NLG challenge. In *Computational Linguistics*.

Sergey Edunov, Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Classical structured prediction losses for sequence to sequence learning. In *NAACL-HLT*.

Desmond Elliott and Frank Keller. 2014. Comparing automatic evaluation measures for image description. In *ACL*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *ACL workshop for NLP Open Source Software*.

Yoav Goldberg, Graeme Hirst, Yang Liu, and Meng Zhang. 2018. Neural network methods for natural language processing. *Computational Linguistics*, 44(1).

Tatsunori B. Hashimoto, Hugh Zhang, and Percy Liang. 2019. Unifying human and statistical evaluation for natural language generation. In *NAACL*.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *NeurIPS*.

Gao Huang, Chuan Guo, Matt J. Kusner, Yu Sun, Fei Sha, and Kilian Q. Weinberger. 2016. Supervised word mover's distance. In *NeurIPS*.

Natasha Jaques, Shixiang Gu, Dxmitry Bahdanau, Jose Miguel Hernandez-Lobato, Richard E. Turner, and Douglas Eck. 2017. Counterfactual multi-agent policy gradients. In *ICML*.

Mert Kilickaya, Aykut Erdem, Nazli Ikizler-Cinbis, and Erkut Erdem. 2017. Re-evaluating automatic metrics for image captioning. In *EACL*.

Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. 2015. From word embeddings to document distances. *ICLR*, 37.

Mirella Lapata and Regina Barzilay. 2005. Automatic evaluation of text coherence: Models and representations. In *IJCAI*.

Alon Lavie and Abhaya Agarwal. 2007. METEOR: An automatic metric for MT evaluation with high levels of with human judgments. In *Second Workshop on Statistical Machine Translation*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *ACL*.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *EMNLP*.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *LREC*.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. SummaRuNNer: A recurrent neural network based sequence model for extractive summarization of documents. In *AAAI*.

Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Caglar Gülehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *CoNLL*.

Jekaterina Novikova, Ondrej Dusek, Amanda Cercas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for NLG. In *EMNLP*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Ramakanth Pasunuru and Mohit Bansal. 2017. Reinforced video captioning with entailment rewards. In *EMNLP*.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *ICLR*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL*.

Christopher Purdy, Xinyu Wang, Larry He, and Mark O. Riedl. 2018. Predicting generated story quality with quantitative measures. In *AIIDE*.

MarcAurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. In *ICLR*.

Ehud Reiter. 2011. Task-based evaluation of NLG systems: Control vs real-world context. In *UCNLG+Eval*.

Ehud Reiter and Anja Belz. 2006. GENEVAL: A proposal for shared-task evaluation in NLG. In *INLG*.

Ehud Reiter and Anja Belz. 2009. An investigation into the validity of some metrics for automatically evaluating natural language generation systems. In *CVPR*.

Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. 2016. Self-critical sequence training for image captioning. In *CVPR*.

Tomasi C. Rubner, Y. and L. J. Guibas. 1998. A metric for distributions with applications to image databases. In *IEEE*.

Manasvi Sagarkar, John Wieting, Lifu Tu, and Kevin Gimpel. 2018. Quality signals in generated stories. In **SEM 2018*.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *ACL*.

Octavia-Maria Sulea. 2017. Recognizing textual entailment in Twitter using word embeddings. In *2nd Workshop on Evaluating Vector-Space Representations for NLP*.

Tsegaye Misikir Tashu and Tomás Horváth. 2018. Pair-Wise: Automatic essay evaluation using word mover's distance. In *CSEDU*.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. CIDEr: Consensus-based image description evaluation. In *CVPR*.

Evan J. Williams. 1959. *Regression Analysis*, volume 14. Wiley.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Machine Learning*.

Lingfei Wu, Ian En-Hsu Yen, Kun Xu, Fangli Xu, Avinash Balakrishnan, Pin-Yu Chen, Pradeep Ravikumar, and Michael J. Witbrock. 2018. Word movers embedding: From word2vec to document embedding. In *EMNLP*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. ArXiv:1609.08144.

Hao Zhang and Jie Wang. 2018. Semantic WordRank: Generating finer single-document summarizations. In *IDEAL*.

Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. 2017. Earth mover's distance minimization for unsupervised bilingual lexicon induction. In *EMNLP*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. BERTScore: Evaluating text generation with BERT. *CoRR*, abs/1904.09675.

A Appendix

A.1 Datasets

Summaries and Essays: For the intrinsic tasks in §4, we use two types of human-evaluated texts: machine-generated summaries and human-authored essays. We follow Kusner et al. (2015) and remove punctuation and stopwords. (For contextual embeddings, these are removed after the embeddings are obtained.) The details of the subsets we used are in Table 7.

	Summaries	Essays
# documents	2,085	1,088
# tokens	255,609	164,776
# types	12,882	6,381
average length (tokens)	65	151
average length (sent.)	3.4	7.5

Table 7: Corpora statistics.

CNN/Daily Mail: CNN/Daily Mail dataset (Nallapati et al., 2017; Hermann et al., 2015) is a collection of online news articles along with multi-sentence summaries. We use the same data splits as in Nallapati et al. (2017). Earlier work anonymized entities by replacing each named entity with a unique identifier (e.g., *Dominican Republic*→`entity15`). In this work we used the non-anonymized version.

Stats	CNN/DM
Avg. # tokens document	781
Avg. # tokens summary	56
Total # train doc-summ. pair	287,229
Total # validation doc-summ. pair	13,368
Total # test doc-summ. pair	11,490
Input token length	400/800
Output token length	100

Table 8: Summary statistics of CNN/Daily Mail (CNN/DM) Datasets.

A.2 More Examples

In Table 9, we show samples of the summaries that we used to perform intrinsic evaluations in the main text.

A.3 Extrinsic Model Training Details

We use 128 dimensional bidirectional 2-layered LSTMs for the encoder and 128 unidirectional LSTMs for the decoder. For both datasets, we limit the input and output vocabulary size to the 30,000 most frequent tokens in the training set. We initialize word embeddings with FastText[14] (Mikolov et al., 2018) 300-dimensional vectors and finetune them during training. For WMS, SMS and S+WMS embeddings, we use the GloVe word embeddings described in §4. We train using Adam with a learning rate of 0.001 for the MLE models and 10^{-5} for the MLE+RL models. We select the MLE models with the lowest cross-entropy loss and the MLE+RL models with the highest reward on a sample of validation data to evaluate on the test set. At test time, we use beam search of width 5 on all our models to generate final predictions. For the Mixed RL trained models, we initialize the weights with pre-trained MLE model, and we start with $\gamma = 0.97$ and gradually increase its value. We train our models for ∼25 epochs which took 1–2 days on an NVIDIA V100 GPU machine.

A.4 Policy Gradient Reinforce Training

Maximum likelihood-based training of sequence generation models poses exposure bias issues since the model is evaluated by comparing the model to empirical distribution, whereas at test time we use automatic metrics to evaluate the model generated text (Ranzato et al., 2015). Reinforced based policy gradient approach is used to address this issue by learning to optimize discrete target evaluation metrics that are non-differentiable. We use REINFORCE (Williams, 1992) to learn a policy p_θ defined by the model parameters θ to predict the next action (word). The RL loss function is defined as:

$$L_{RL} = \mathbb{E}_{\hat{y}\sim p_\theta}[r(\hat{y})] \quad (9)$$

where $\hat{y}$ is the sequence of sampled words. The derivative of the the objective function based on Monte Carlo sampling yields:

$$\bigtriangledown_\theta L_{RL} = -(r(\hat{y}) - b) \bigtriangledown_\theta \log p_\theta(\hat{y}) \quad (10)$$

The baseline b is a bias estimator and is used for variance reduction in RL training. In this work we use *self-critical training* and use the reward obtained from a sequence that is generated by greedily decoding, $\tilde{y}$, as a baseline:

$$\bigtriangledown_\theta L_{RL} = -(r(\hat{y}) - r(\tilde{y})) \bigtriangledown_\theta \log p_\theta(\hat{y}) \quad (11)$$

A.5 Human Evaluations

Evaluation Procedure We randomly selected 100 samples from the CNN/Daily Mail test set

[14] https://fasttext.cc/docs/en/english-vectors.html

Samples	Summaries
Sample #1	**Reference.** Freddie Gray, who is black, asked for medical help but was denied during 00-minute police car ride, eventually paramedics were called. Deputy police commissioner Kevin Davis conceded their failure. But chief commissioner refuses to resign over the death. Six officers are suspended without pay during an investigation.
	Hypothesis. Baltimore Police Commissioner Anthony Batts ruled out his resignation despite that fact that his deputy admitted they should have sought medical attention for Freddie Gray. Six officers have been suspended with pay as local police and federal authorities investigate. Commissioner Anthony Batts has ruled out the possibility of his resignation.
Sample #2	**Reference.** Choc on Choc's chocolates come in three different flavours. The face of each politician is emblazoned on milk Belgium chocolate bars. Cameron's has blueberries, Clegg is honeycomb and Miliband is raspberry.
	Hypothesis. UNK lollies on 273 invalid chocolates come in three different flavours. Contains three different flavours - the colours associated with each leader. David Cameron, Nick Clegg, Nick Clegg, Nick Clegg and David Cameron.
Sample #3	**Reference Essay.** The setting seems to be as formidable an opponent as the actual workout. It seems as if everything is against the cyclist, including nature. As the day progresses, and the cyclist's journey continues, the setting becomes harsher and harsher. After passing the first "town", the "sun was beginning to beat down." In need of water, all a cruel pump gives him is "a tarlike substance." His sufferings continue, increasingly pummeled by his surroundings and his thirst for water. If dehydration was not enough, the flat terrain gave way to "rolling hills", which would only punish his legs more. Reaching possible salvation, his hopes are crushed when the "Welch's Grape Juice Factory" turns out to be abandoned. All these events are enough to destroy anyone's spirit. The cyclist almost gives up hope to accept certain death. He has become ferociously beaten by his very surroundings. It appears as if he is fated to die alone in the blistering heat. Although he hangs his head in despair, he still continues on the path of disappointment. In a twist of fate, he encounters a thriving store where he halts and drinks. Finally encountering his salvation, this particular setting brings new hope and relief to the cyclist who has finally survives his trek through nature.
	Hypothesis. The features of the setting affect the cyclist alot. The hot sun beating down on him makes him sweat and makes him thirsty. The bumpy roods and hills make him work harder. The abandoned places make him lose hope. If faced with these obstacles I would have been affected in the same way. As I believe any human would be.

Table 9: Examples of human generated and model generated summaries from Summaries and Essays datasets

and use workers from Amazon Mechanical Turk as judges to evaluate them on the four criteria (redundancy, focus, coherence, and overall). Following DUC (Document Understanding Conferences) style evaluations (https://duc.nist.gov/), we performed a head-to-head evaluation and randomly showed Turkers two model-generated summaries. We asked the human annotators to rate each summary on the same metrics as before without seeing the source document or ground truth summaries.