Speaker Sensitive Response Evaluation Model

JinYeong Bak
School of Computing
KAIST
jy.bak@kaist.ac.kr

Alice Oh
School of Computing
KAIST
alice.oh@kaist.edu

Abstract

Automatic evaluation of open-domain dialogue response generation is very challenging because there are many appropriate responses for a given context. Existing evaluation models merely compare the generated response with the ground truth response and rate many of the appropriate responses as inappropriate if they deviate from the ground truth. One approach to resolve this problem is to consider the similarity of the generated response with the conversational context. In this paper, we propose an automatic evaluation model based on that idea and learn the model parameters from an unlabeled conversation corpus. Our approach considers the speakers in defining the different levels of similar context. We use a Twitter conversation corpus that contains many speakers and conversations to test our evaluation model. Experiments show that our model outperforms the other existing evaluation metrics in terms of high correlation with human annotation scores. We also show that our model trained on Twitter can be applied to movie dialogues without any additional training. We provide our code and the learned parameters so that they can be used for automatic evaluation of dialogue response generation models.

1 Introduction

Evaluating the system generated responses for open-domain dialogue is a difficult task. There are many possible appropriate responses given a dialogue context, and automatic metrics such as BLEU (Papineni et al., 2002) or ROUGE (Lin, 2004) rate the responses that deviate from the ground truth as inappropriate. Still, it is important to develop and use an automatic metric because human annotation is very costly. In addition to BLEU and ROUGE, there is a widely-used evaluation metric based on the distributed word representation (Liu et al., 2016), but this metric shows low correlations with human judgments.

One reason for the difficulty in developing an automatic metric that correlates well with human judgements is that the range of appropriate responses for a given context is very wide. Table 1 shows an example of a conversation between Speaker *A* and *B*. While there is a ground truth response "Yeah let's go to the theater," *A* could have also said "That sounds good! Have you seen Thor?" or "Good. What movie?" Note that based on word overlap with the ground truth, these two responses would receive low scores. Responses labeled N#, such as "The weather is no good for walking" are not appropriate. As the Table shows, the existing metrics from BLEU to RUBER are not able to tell apart these appropriate A# responses from the inapproriate N# responses.

Some recent metrics such as ADEM (Lowe et al., 2017) and RUBER (Tao et al., 2018) compute the similarity between a context and a generated response. However, ADEM requires human-annotated scores to train and thus cannot be applied to new datasets and domains. RUBER overcomes this limitation by using the idea that a random response should be used as a "negative sample", but it is not able to distinguish the responses in the example in Table 1, because it uses only one random sample which does not provide sufficient information about appropriate and inappropriate responses.

In this paper, we propose Speaker Sensitive Responses Evaluation Model (SSREM) that analyzes the appropriateness of the responses. We use speaker sensitive responses that are generated by one speaker to train the model. We test SSREM in comparison with other evaluation metrics. First, we make annotated human scores for responses in Twitter conversation data. The evaluation scores of SSREM shows a higher correlation with human scores than other evaluation metrics. And SSREM outperforms other metrics in terms of identifying the ground truth responses given a context. We

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6376–6385
July 5 - 10, 2020.

	Context	*A*: What do you want to do tonight? *B*: Why don't we go see a movie?					
	Ground truth response	*A*: Yeah Let's go to the theater					
	Utterance	BLEU	ROUGE	EMB	RUBER	SSREM	Human
A1	That sounds good! Have you seen Thor?	0.00 (3)	0.00 (3)	0.95 (2)	0.59 (2)	0.64 (1)	5.00 (1)
A2	Good, What movie?	0.00 (3)	0.00 (3)	0.92 (4)	0.55 (4)	0.62 (2)	5.00 (1)
A3	Or hang out in city	0.00 (3)	0.00 (3)	0.89 (6)	0.48 (5)	0.49 (3)	3.80 (3)
N1	The weather is no good for walking	0.32 (1)	0.15 (2)	0.94 (3)	0.47 (6)	0.44 (4)	2.60 (4)
N2	The sight is extra beautiful here	0.32 (1)	0.17 (1)	0.97 (1)	0.64 (1)	0.38 (5)	1.00 (5)
N3	Enjoy your concert	0.00 (3)	0.00 (3)	0.91 (5)	0.57 (3)	0.33 (6)	1.00 (5)

Table 1: Example of appropriate responses (A1 - A3) and non-appropriate responses (N1 - N3) for a given context and ground truth response, and the responses' scores by evaluation metrics. Emb is embedding average and Human is average scores from five people. Ranks are shown in brackets. SSREM has positive correlation with human scores.

show the additional advantage of SSREM: it can be applied to evaluate a new corpus in a different domain. We train SSREM on Twitter corpus and test it on a corpus of movie reviews, and we show that SSREM outperforms other metrics in terms of the correlation with human scores and the task of identifying the ground truth response.

Our contributions in this paper include the following.

- We present SSREM, a new response evaluation model trained with speaker sensitive negative samples (Sec 3).
- We conduct experiments on a Twitter conversation corpus and show that SSREM outperforms the others (Sec 5 and 6). We further show the applicability of SSREM with Movie dialogue corpus that are not using in the training (Sec 7).
- We provide our code and the learned parameters of SSREM which can be used for evaluation of generated responses[1].

2 Related Work

In this section, we describe existing automatic evaluation metrics for dialogue response generation and discuss their limitations.

For task-oriented dialogue models such as airline travel information system (Tur et al., 2010), completing the given task is most important, and the evaluation metrics reflect that (Hastie, 2012; Bordes et al., 2017). But open-domain conversation models do not have specific assigned tasks; the main goal of an open-domain conversation model is generating appropriate responses given a conversation about any topic.

Existing automatic evaluation metrics compare a generated response and the ground truth response. The most widely-used metric are BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) based on the overlap of words between the two responses. A limitation of these word overlap-based metrics is that they cannot identify the synonyms, and to overcome this limitation, the embedding-based metrics use distributed word vector representations (Liu et al., 2016). However, these metrics have poor correlation with human judgments (Liu et al., 2016; Novikova et al., 2017; Gupta et al., 2019) because they still only look at the similarity between the generated response and the ground truth. SSREM is a model with the awareness that a response can be different from the ground truth response but still appropriate for the conversation context.

The responses for a casual conversation can be varied. For example, there are four appropriate responses including ground truth response for a given context in Table 1. Some previous approaches suggest considering the context together with the response such as ADEM (Lowe et al., 2017) and RUBER (Tao et al., 2018). ADEM uses pre-trained VHRED (Serban et al., 2017) to encode the texts and compute the score by mixing similarities among the context, generated response and a ground truth. One limitation of ADEM is that it requires human annotated scores to learn the model. Human labeling is cost-intensive, so it is impractical to apply to a new dataset or domain. RUBER uses negative sampling to overcome this issue, but it uses only one random negative sample against one positive sample which is not ideal (Gutmann and Hyvärinen, 2010). SSREM does not require

[1] https://github.com/NoSyu/SSREM

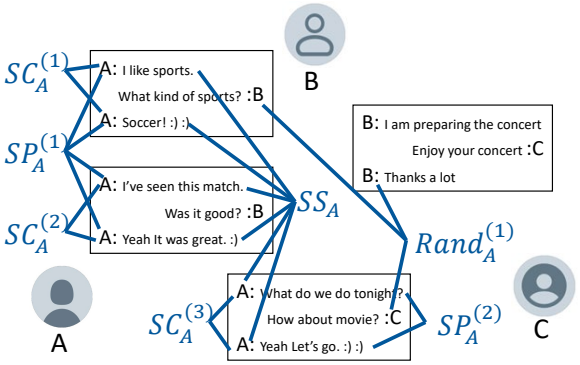


Figure 1: Example of utterance sets for speaker A. SC stands for 'same conversation', SP for 'same partner', SS for 'same speaker', and $Rand$ for 'random'.

SC	SP	SS	$Rand$
.922±1e-4	.919±2e-4	.912±3e-4	.898±2e-3

Table 2: Mean similarity among utterances in SC, SP, SS and $Rand$ sets with a 95% confidence interval

human scores to learn the model and uses many speaker sensitive negative samples.

3 Speaker Sensitive Response Evaluation Model

This section describes our Speaker Sensitive Response Evaluation Model (SSREM) that trains with speaker sensitive utterance samples. SSREM looks at a given context and its ground truth response together to evaluate a generated response. We describe the motivation of SSREM with empirical observations in section 3.1. We present the structure of SSREM in section 3.2. With the motivation, we present a training method of SSREM with speaker sensitive utterance samples in section 3.3.

3.1 Motivation

We are motivated by the assumption that there is varying degree of similarity among utterances in a corpus of conversations containing many speakers and conversations.

1. If we pick a set of random utterances from the corpus, they will not be very similar.

2. If we pick a set of utterances from a single speaker conversing with multiple partners, those utterances will be more similar than the random utterances in 1.

3. If we pick a set of utterances from conversations between a single dyad, even if the conversations are far apart in time, those utterances would be more similar than those in 2.

4. If we pick a set of utterances in a single conversation session, they are the most similar, even more so than those in 3.

To test these assumptions, we first categorize one speaker A's utterances into four types of sets corresponding to the assumptions above.

- Random ($Rand_A$): Random utterances from speakers who are not A
- Same Speaker (SS_A): Speaker A's utterances
- Same Partner (SP_A): A's utterances in conversations with the same partner B
- Same Conversation (SC_A): A's utterances in a single conversation

Figure 1 shows one example of the sets. We make three SC_A sets because A participates in three conversations. We make two SP_A sets because A has conversations with B and C. SS_A is all utterances from A so we create one set of utterances for A. Finally, $Rand_A$ is random utterances from non-A's utterances. We create five sets for each speaker.

From these sets, we compute the similarity among utterances in a set. First, we convert an utterance into a vector by averaging the words in the utterance with GloVe Twitter 200d (Pennington et al., 2014). And we compute the similarity of the vectors by Frobenius norm. Finally, we calculate the mean similarity of each set with a 95% confidence interval. Table 2 shows the results. $Rand$ has the lowest similarity mean value, so it supports the first assumption. SS has higher similarity mean value than $Rand$. It supports the second assumption. The mean similarity value of SP is higher than SS. It supports the third assumption. Finally, SC has the highest mean similarity value. It also supports the last assumption. From the observations, we assume that utterances are clustered by the speakers and addressees.

3.2 SSREM

SSREM evaluates a generated response $\hat{\mathbf{r}}$ from a context $\mathbf{c}$ and a ground truth response $\mathbf{r}$. The output of SSREM is as follows:

$$SSREM(\mathbf{c}, \mathbf{r}, \hat{\mathbf{r}}) = h(f(\mathbf{c}, \hat{\mathbf{r}}), g(\hat{\mathbf{r}}, \mathbf{r})) \quad (1)$$

where $f(\mathbf{c}, \hat{\mathbf{r}}) = tanh(V(\mathbf{c})^T \mathbf{M} V(\hat{\mathbf{r}}))$ is a parametrized function to measure the similarity between the context $\mathbf{c}$ and the generated response $\hat{\mathbf{r}}$. V is a function to convert a sequence of words to a vector. $\mathbf{M}$ is a matrix that weights of the similarity between two vectors. It is the parameter of the f function. $g(\mathbf{r}, \hat{\mathbf{r}})$ is another function to measure the ground-truth response and the generated one. h is a function to mix the values of f and g functions. To normalize each output of the f and g functions, we adopt linear scaling to unit range (Aksoy and Haralick, 2001) which rescale the value x as follows:

$$\tilde{x} = \frac{x - l}{u - l} \tag{2}$$

where u is an maximum and l is minimum of x.

SSREM is similar to RUBER, which computes the similarities among $\mathbf{c}$, $\mathbf{r}$ and $\hat{\mathbf{r}}$ separately and merge it at the end. However, SSREM uses speaker sensitive samples, whereas RUBER takes one positive sample and one negative sample.

3.3 Training with Speaker Sensitive Samples

SSREM has a parametrized function f that takes context $\mathbf{c}$ and a generated response $\hat{\mathbf{r}}$. To train the f function, we define a classification problem to identify the ground truth response $\mathbf{r}$ from a set of candidate responses R_{cand}. The R_{cand} has the ground truth response and some negative samples. A classifier tries to identify the ground truth response with the negative samples. Negative samples are usually selected from the uniform distribution. But we sample the speaker sensitive utterances which described in section 3.1 for SSREM.

Formally speaking, let A be the speaker of the ground truth response $\mathbf{r}_A$. It means it is A's turn to say the response for the context $\mathbf{c}$. The candidate response set R_{cand_A} is given by

$$R_{cand_A} = \{\mathbf{r}_A, \mathbf{sc}_A, \mathbf{sp}_A, \mathbf{ss}_A, \mathbf{rand}_A\} \tag{3}$$

where $\mathbf{sc}_A \in SC_A \setminus \mathbf{c}$, $\mathbf{sp}_A \in SP_A \setminus \mathbf{c}$, $\mathbf{ss}_A \in SS_A \setminus \mathbf{c}$ and $\mathbf{rand}_A \in Rand_A$ are the negative samples from speaker sensitive responses. Then, the probability of a ground truth response $\mathbf{r}_A$ given context $\mathbf{c}$ and R_{cand_A} is as follows:

$$p(\mathbf{r}_A | \mathbf{c}, R_{cand_A}) = \frac{exp(f(\mathbf{c}, \mathbf{r}_A))}{\sum_{\mathbf{r}' \in R_{cand_A}} exp(f(\mathbf{c}, \mathbf{r}'))} \tag{4}$$

We maximize this probability among all context-ground truth response pair. So the loss function of the classification problem is

$$-\sum_{\mathbf{c}} \log \frac{exp(f(\mathbf{c}, \mathbf{r}_A))}{\sum_{\mathbf{r}' \in R_{cand_A}} exp(f(\mathbf{c}, \mathbf{r}'))} \tag{5}$$

This approach is similar to learning the sentence representations (Logeswaran and Lee, 2018), but we use the speaker sensitive negative samples. It is also similar to Noise Contrastive Estimation (NCE) (Gutmann and Hyvärinen, 2010; Mnih and Teh, 2012). But we set the noise distribution to speaker sensitive distribution and only take the data sample term in the objective function of the NCE.

Selecting negative samples is important for learning. When we choose the noise distribution, it would be close to the data distribution, because otherwise, the classification problem might be too easy to learn the data (Gutmann and Hyvärinen, 2010). Mnih and Teh (2012) shows that using samples from the unigram distribution outperforms using samples from a naive uniform distribution for learning a neural probabilistic language model. Likewise, we create negative samples from the speaker sensitive utterances. $\mathbf{sc}_A$ is more similar to the $\mathbf{r}_A$ than any other negative samples. We show the patterns by empirical observations in section 3.1 and experimental results in section 6.2. These speaker sensitive samples make the classification problem harder and lead to learning the function f better than using the naive uniform distributed random samples.

To train SSREM, we need a conversation corpus that has many conversations from one speaker. We choose the Twitter conversation corpus (Bak and Oh, 2019) as it has 770K conversations with 27K Twitter users. We split the data as 80/10/10 for training/validation/test.

4 Annotating Human Scores

To measure the correlation SSREM with human judgments, we first gather human judgments of responses given a conversation context. We use Amazon Mechanical Turk (MTurk) to annotate the scores of the responses. We select 300 conversations from a dataset of Twitter conversations. And we generate responses for annotation using three conversation models and the ground truth response for each conversation.

- Retrieval model (Pandey et al., 2018): A

Human Score	1	2	3	4	5
Twitter	211	258	342	278	71
Movie	279	267	311	217	126

Table 3: Basic statistics of human scores of the responses on Twitter conversation and Movie scripts

BM25 retrieval model (Robertson et al., 2009) that uses TF-IDF vector space.

- VHCR (Park et al., 2018): A variational autoencoder model that has a global variable for a conversation.
- VHUCM (Bak and Oh, 2019): A variational autoencoder model that considers the speakers of a conversation.

Then we ask two questions to the MTurkers. (1) How appropriate is the response overall? (2) How on-topic is the response? These questions are used in (Lowe et al., 2017). The authors show that these questions have high inter-annotator agreement among workers. They suggest using the first question to annotate the human score, and so we follow the suggestion. But we ask the second question to workers to filter out workers who submit random answers. Each worker answers these questions on a five-point Likert scale.

We annotate 1,200 responses in total. One worker answers ten conversations, four responses per conversation for a total of 40 responses. Each response is tagged by five workers for a total of 287 workers of which we retain the responses from 150 workers who passed all the tests. We tag the most selected score as the human score for each response. The inter-annotator Fleiss' kappa (Fleiss, 1971) is $\kappa = 0.61$ which is consistent with the results in (Lowe et al., 2017). Table 3 shows the basic statistics of the annotations.

5 Experiment 1 - Comparing with Human Scores

This section describes the experiment that looks at the correlation between the model scores and the human scores for given contexts and responses.

5.1 Experiment Setup

We use a Twitter conversation corpus (Bak and Oh, 2019) to train and validate SSREM and other baseline models. For the test, we remove the ground truth responses in human-annotated corpus since it always produces the maximum score on BLEU and ROUGE.

We compare SSREM with the following response evaluation methods:

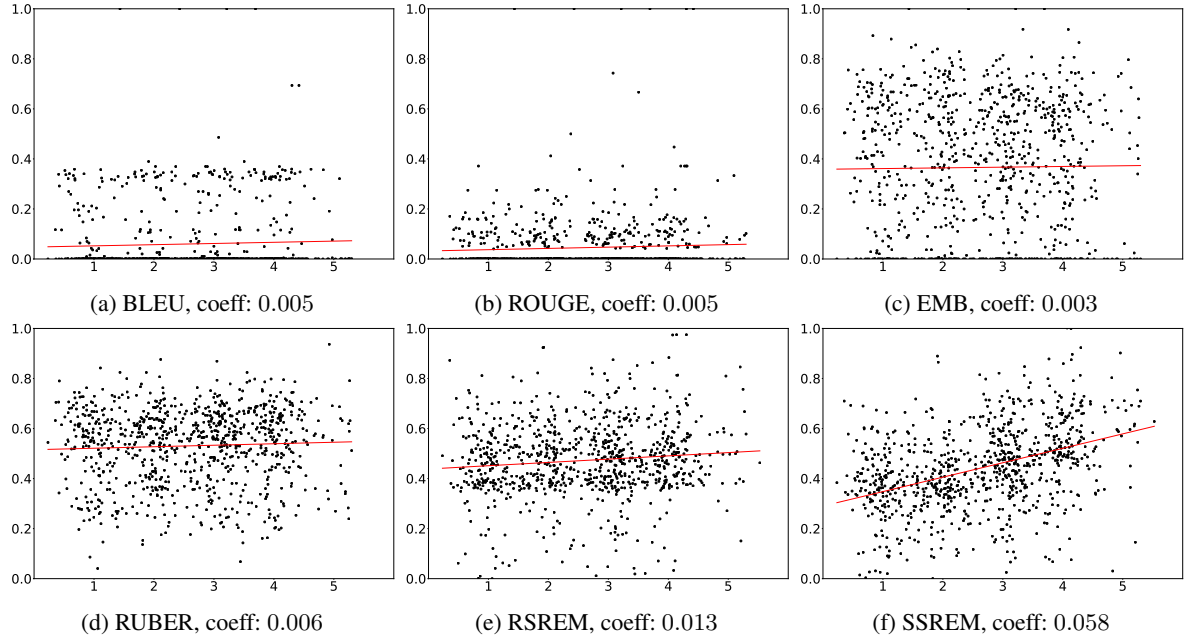
- BLEU (Papineni et al., 2002): We compute the sentence-level BLEU score with the smoothing seven technique (Chen and Cherry, 2014).
- ROUGE (Lin, 2004): We compute the F score of ROUGE-L.
- EMB (Liu et al., 2016): We compute the average cosine similarity between ground truth response and test response in a word embedding[2]. We use pre-trained Google news word embedding (Mikolov et al., 2013) to avoid the dependency between the training data and embedding.
- RUBER (Tao et al., 2018): We train with a random negative sample to train unreferenced metric in RUBER. And we use arithmetic averaging to hybrid the referenced and unreferenced metrics.
- RSREM: We use the same structure of SSREM, but train with uniformly random negative samples, not speaker sensitive samples.

We choose functions in SSREM for the experiment. For V function, We use the word averaging technique that averages the vectors of words in the sequence. We can use advanced methods such as RNN or sentence embeddings (Reimers and Gurevych, 2019). But for the fair comparisons with RUBER, we select a similar approach. We use GloVe Twitter 200d word embedding (Pennington et al., 2014). For g function, we use sentence mover's similarity that is the state of the art evaluating reference-candidate pair of sentences by using word and sentence embeddings (Clark et al., 2019). To avoid dependency between the training data and embedding, we use Elmo embedding (Peters et al., 2018). For h function, we use arithmetic averaging that shows good results in (Tao et al., 2018).

5.2 Results and Discussion

Table 4 shows the Spearman and Pearson correlations between human scores and models scores. First, BLEU, ROUGE, and EMB are not correlated

[2]We experimented with the greedy and extreme embedding for comparison, but these methods were not better than the average embedding.



(a) BLEU, coeff: 0.005 (b) ROUGE, coeff: 0.005 (c) EMB, coeff: 0.003

(d) RUBER, coeff: 0.006 (e) RSREM, coeff: 0.013 (f) SSREM, coeff: 0.058

Figure 2: Scatter plots that show model scores against human scores. We add Gaussian noise drawn from $N(0, 0.3)$ to the human scores to better visualize the density of points (Lowe et al., 2017). The red line is a linear regression line, and the coeff is the coefficient of the line. SSREM shows a higher positive correlation with human judgment than the other models.

Metric	Spearman	Pearson
BLEU	0.024 (0.472)	0.041 (0.227)
ROUGE	0.024 (0.471)	0.052 (0.124)
EMB	0.006 (0.861)	0.012 (0.720)
RUBER	0.044 (0.192)	0.046 (0.177)
RSREM	0.088 (< 0.01)	0.101 (< 0.01)
SSREM	**0.392** (< 0.001)	**0.376** (< 0.001)

Table 4: Correlation between human and model scores. We compute Spearman and Pearson correlation coefficients. p-values are shown in brackets. SSREM shows higher correlation with human judgement than the other models.

with human scores. It means evaluating responses with ground truth only is not useful. These results are the same in previous research (Liu et al., 2016; Lowe et al., 2017; Tao et al., 2018). RUBER shows a higher correlation with human scores than other baselines but has a high p-value that means low statistically significant. RSREM performs better than RUBER and other baselines. It shows using multiple negative samples improves the performance of learning the model. Finally, SSREM outperforms all other methods for two correlations with low p-values. It shows the effectiveness of using speaker sensitive negative samples.

Figure 2 shows scatterplots of the human and model scores. A dot is one response, and a red line is a linear regression line. The x-axis is the human score, and the y-axis is each automatic evaluation metric. To visualize the dots better, we adopt the technique from (Lowe et al., 2017) that adds random number ($N(0, 0.3)$) to x-axis value. But, we train the linear regression with original scores. First, BLEU and ROUGE have many zero values since there are few overlapped words between the generated response and the ground-truth response. The dots in EMB that uses word embedding to overcome the limitation are more distributed. But there are few relationships with human scores, and the linear regression coefficient is flattened. RUBER is better than BLEU, ROUGE, and EMB. RSREM that uses more negative samples shows better than RUBER. Finally, SSREM shows a higher positive correlation with human scores than other baselines.

6 Experiment 2 - Identifying True and False Responses

The second experiment presents the performance of f function in SSREM by comparing it with baselines. RUBER, RSREM, and SSREM compute the score from the context of the conversation and generated responses. To investigate the performance of the score, we set up the task that identifies the true and false responses for a given context. The

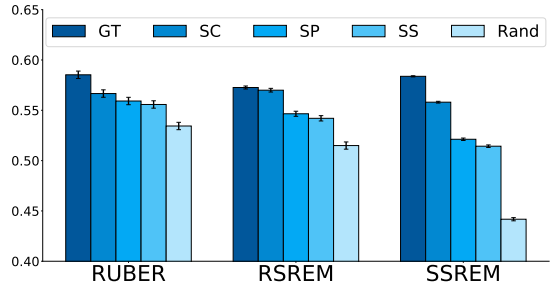


Figure 3: Difference of scores on various responses in Twitter conversation corpus. The range of the vertical error bar is a 95% confidence interval of the values among the responses. SSREM outperforms the other models for identifying true and false responses.

true responses are ground-truth responses, and false ones are four negative samples that are described in section 3.3.

6.1 Experiment Setup

The data for this experiment is the test data of the Twitter conversation corpus. We extract contexts, true and false responses from the data. The true response is the ground-truth response (GT). And the false responses are four types that are described in section 3.3 (SC, SP, SS, $Rand$).

We compare SSREM with RUBER and RSREM that compute the similarity between a context and a response. We take the unreferenced metric score in RUBER. And we take the output of the f function in RSREM and SSREM. We use the same trained models in section 5.

6.2 Results and Discussion

Figure 3 shows the results. The x-axis is the models, and the y-axis is the output of the unreferenced metric or f function. All models perform well on distinguishing between GT utterances and $Rand$ utterances. But RUBER performs poor on identifying SC, SP, and SS. And RSREM cannot identify false responses from SC. Finally, SSREM outperforms the other two models for identifying all cases. It also maximizes the difference between GT and $Rand$ than the other two models. It is another clue for showing the effectiveness of using speaker sensitive negative samples.

One interesting result is that the output scores decrease from GT to $Rand$. It is the same observation about the differences of speaker sensitive utterances in section 3.1. And it also means that identifying GT and SC is a harder problem than GT and $Rand$ pair. It is another evidence for why we use speaker sensitive negative samples, as we discussed in section 3.3.

SC consists of negative samples that are most difficult for the model to distinguish, so it makes sense to consider only SC negative samples. But we include SP and SS for the following two reasons. First, there are only a limited number of SC utterances because they must all come from the same conversation, whereas we need a pretty large number of negative samples to effectively train the model (Mnih and Teh, 2012). Second, we also sample from SP and SS because they represent different degree of similarity to the context utterances. SC utterances are from the same conversation, leading to decreased model generalization.

7 Experiment 3 - Applying New Corpus

In this section, we investigate the applicability of SSREM to a new conversation corpus. SSREM takes the speaker sensitive samples from Twitter. But there are many open-domain conversation corpora such as Movie scripts (Danescu-Niculescu-Mizil and Lee, 2011). Tao et al. (2018) run a similar experiment with RUBER, but they use the similar domain of data, Chinese online forum (Training from Douban and testing on Baidu Tieba). We choose the Movie scripts corpus because it is written by the script writers whereas Twitter is personal causal online conversations. We present the performance of SSREM on the new corpus.

7.1 Experiment Setup

First, we annotate 1,200 responses to the movie dialog corpus. We use HRED (Sordoni et al., 2015) rather than VHUCM. The next procedure of annotation is the same when we create human scores for Twitter conversation responses in section 4. Two hundred forty-four workers tagged all responses. But, 94 workers failed the attention check question, so we collect the 150 workers' answers. The inter-annotator Fleiss' kappa (Fleiss, 1971) for Movie is κ = 0.63. It is still consistent with the results in (Lowe et al., 2017) and annotated Twitter conversations. The bottom row in Table 3 shows the basic statistics of the annotated responses.

We run two experiments, comparing with human scores and identifying true and false responses. We use the same models in section 5. We use the Twitter conversation corpus to train RUBER, RSREM, and SSREM. And we test the models on annotated movie dialogs. Unlike the Twitter conversation cor-

Metric	Spearman	Pearson
BLEU	0.036 (0.378)	0.063 (0.124)
ROUGE	0.041 (0.322)	0.054 (0.191)
EMB	0.022 (0.586)	0.010 (0.815)
RUBER	0.004 (0.920)	-0.009 (0.817)
RSREM	0.009 (0.817)	0.024 (0.550)
SSSREM	**0.132** (< 0.001)	**0.119** (< 0.005)

Table 5: Correlation between human and model scores with Movie corpus. We compute Spearman and Pearson correlation coefficient. p-values are shown in brackets. SSREM shows higher correlation with human judgement than the other models.

pus, the movie dialogs have a short length of conversations. So we choose SC and Ran only to run the second experiment.

7.2 Results and Discussion

In the experiment on comparing with human scores on the movie dialogs corpus, Table 5 shows the results. First, BLEU, ROUGE, and EMB are not correlated with human scores. RUBER shows worse performance than testing on the Twitter corpus. RSREM performs better than RUBER and other baselines, but it also shows worse performance than testing on the Twitter corpus. Finally, SSREM outperforms all other methods for two correlations with low p-values. It shows the effectiveness of using speaker sensitive negative samples for the new corpus. Figure 2 shows the similar results by plotting scatter plots.

In the experiment on identifying true and false responses with the movie dialogs corpus, Figure 5 shows the results of the identification task. RUBER performs poor on distinguishing between GT and $Rand$ statistically significantly. RSREM performs better than RUBER. And SSREM outperforms the other two models for identifying all cases in the new corpus.

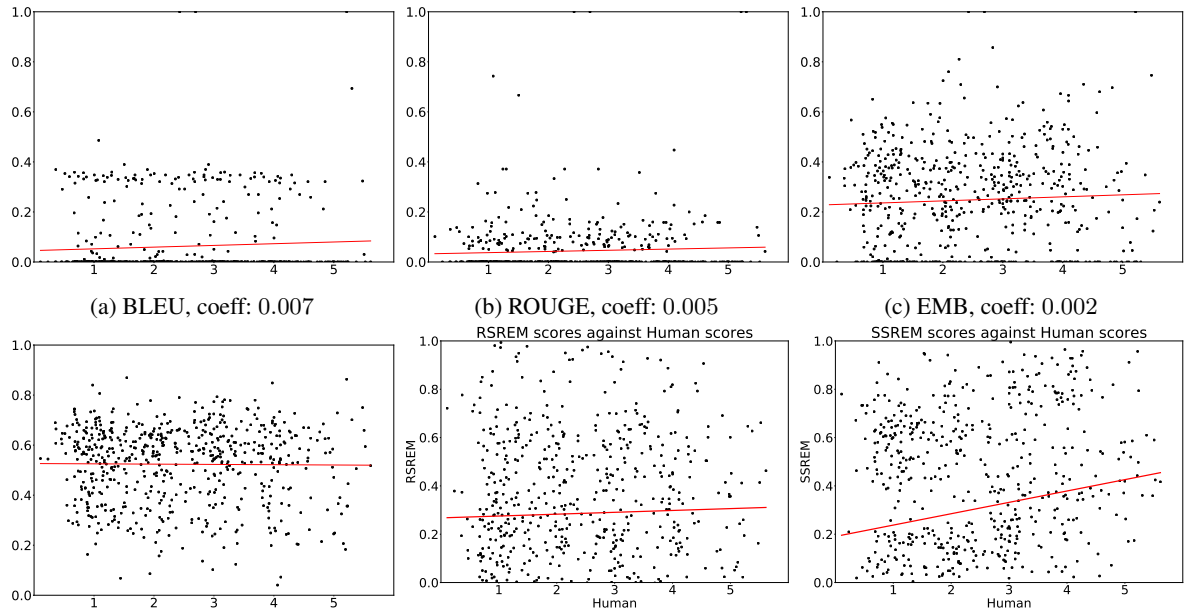
8 Conclusion and Future Work

In this paper, we presented SSREM, an automatic evaluation model for conversational response generation. SSREM looks at the context of the conversation and the ground-truth response together. We proposed negative sampling with speaker sensitive samples to train SSREM. We showed that SSREM outperforms the other metrics including RSREM that uses random negative samples only. We also showed that SSREM is effective in evaluating a movie conversation corpus even when it is trained with Twitter conversations.

There are several future directions to improve SSREM. First, we can make SSREM more robust on adversarial attacks. Sai et al. (2019) shows limitations of ADEM on adversarial attacks such as removing stopwords and replacing words with synonyms. We investigated another type of the adversarial attack named copy mechanism that copies one of the utterances in the context as the generated response. All existing automatic evaluation methods including RUBER that compare the context and the response can be cheated by the copy mechanism. SSREM is also susceptible. However, SSREM is fooled less than other existing models because SSREM learns with negative samples from the set of utterances in the same conversation. SSREM learns to differentiate among utterances in the same context. We show this empirically with an experiment to identify true and false responses (Sec 6.2). When we look at the mean score for the context utterances that shows this copy mechanism compared to the mean score of the ground-truth response (GT), the mean score of context utterances is 0.07 higher by RUBER, but only 0.01 higher by SSREM. SSREM does not give lower scores for the context utterances than GT, but it is not as bad as RUBER. We will make SSREM more robust on the attacks.

Second, we can improve SSREM for a higher correlation with human judgement. We chose to approach SSREM with a classification loss because it is simple and widely used to estimate the models using negative sampling. Although the classification loss is simple, SSREM outperforms all existing automatic evaluation models. However, as Table 2 and Figure 3 are shown, each negative samples has different correlation with the context. We will use ranking loss (Wang et al., 2014; Schroff et al., 2015) to learn the difference among samples. Recently, Zhang et al. (2020) uses BERT (Devlin et al., 2019) to evaluate generated candidate sentences by comparing reference sentence. We used word embeddings to represent an utterance to the vector for the simplicity, but contextual embeddings are much better since it generates more context-related representation than word embeddings. We will use the contextual embedding to represent utterances.

Third, we can extend using SSREM to various conversation corpora such as task-oriented dialogues. We trained and tested SSREM on open-



(a) BLEU, coeff: 0.007 (b) ROUGE, coeff: 0.005 (c) EMB, coeff: 0.002

(d) RUBER, coeff: −0.001 (e) RSREM, coeff: 0.008 (f) SSREM, coeff: 0.047

Figure 4: Scatter plot showing model against human scores with Movie corpus. We add Gaussian noise drawn from $N(0, 0.3)$ to the human scores to better visualize the density of points which is similar to (Lowe et al., 2017).

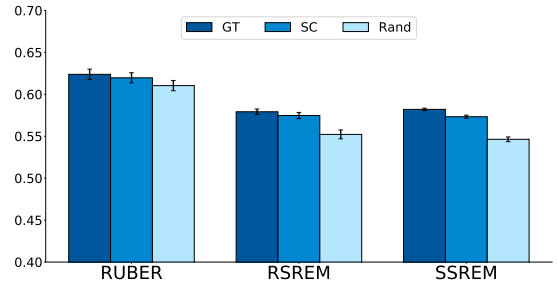


Figure 5: Difference of scores on various responses in Movie corpus. The range of the vertical error bar is a 95% confidence interval of the values among the responses. SSREM outperforms the other models for identifying true and false responses.

domain conversation corpora. However, contextual coherence between the input context and the generated text is important in multi-turn conversations. We will apply SSREM to various conversation tasks for evaluating the generated text automatically. We will explore these directions in our future work.

Acknowledgments

We would like to thank Jeongmin Byun[3] for building the annotation webpage, and the anonymous reviewers for helpful questions and comments. This work was supported by Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2017-0-01779, A machine learning and statistical inference framework for explainable artificial intelligence).

[3] https://jmbyun.github.io

References

Selim Aksoy and Robert M Haralick. 2001. Feature normalization and likelihood-based similarity measures for image retrieval. *Pattern recognition letters*.

JinYeong Bak and Alice Oh. 2019. Variational hierarchical user-based conversation model. In *Proceedings of the EMNLP-IJCNLP*.

Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2017. Learning end-to-end goal-oriented dialog. In *Proceedings of the ICLR*.

Boxing Chen and Colin Cherry. 2014. A systematic comparison of smoothing techniques for sentence-level bleu. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*.

Elizabeth Clark, Asli Celikyilmaz, and Noah A. Smith. 2019. Sentence mover's similarity: Automatic evaluation for multi-sentence texts. In *Proceedings of the ACL*.

Cristian Danescu-Niculescu-Mizil and Lillian Lee. 2011. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the 2nd Workshop on Cognitive Modeling and Computational Linguistics*.

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 NAACL*.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*.

Prakhar Gupta, Shikib Mehri, Tiancheng Zhao, Amy Pavel, Maxine Eskenazi, and Jeffrey Bigham. 2019. Investigating evaluation of open-domain dialogue systems with human generated multiple references. In *Proceedings of the SIGdial*.

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the AISTATS*.

Helen Hastie. 2012. Metrics and evaluation of spoken dialogue systems. In *Data-Driven Methods for Adaptive Spoken Dialogue Systems*. Springer.

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

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike 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 *Proceedings of the EMNLP*.

Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. In *Proceedings of the ICLR*.

Ryan Lowe, Michael Noseworthy, Iulian Vlad Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. 2017. Towards an automatic Turing test: Learning to evaluate dialogue responses. In *Proceedings of the ACL*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the NIPS*.

Andriy Mnih and Yee Whye Teh. 2012. A fast and simple algorithm for training neural probabilistic language models. In *Proceedings of the ICML*.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for NLG. In *Proceedings of the EMNLP*.

Gaurav Pandey, Danish Contractor, Vineet Kumar, and Sachindra Joshi. 2018. Exemplar encoder-decoder for neural conversation generation. In *Proceedings of the ACL*.

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

Yookoon Park, Jaemin Cho, and Gunhee Kim. 2018. A hierarchical latent structure for variational conversation modeling. In *Proceedings of the NAACL*.

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

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

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the EMNLP-IJCNLP*.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*.

Ananya B. Sai, Mithun Das Gupta, Mitesh M. Khapra, and Mukundhan Srinivasan. 2019. Re-evaluating adem: A deeper look at scoring dialogue responses.

F. Schroff, D. Kalenichenko, and J. Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the CVPR*.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Proceedings of the AAAI*.

Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the CIKM*.

Chongyang Tao, Lili Mou, Dongyan Zhao, and Rui Yan. 2018. Ruber: An unsupervised method for automatic evaluation of open-domain dialog systems. In *Proceedings of the AAAI*.

Gokhan Tur, Dilek Hakkani-Tür, and Larry Heck. 2010. What is left to be understood in atis? In *Proceedings of the IEEE Spoken Language Technology Workshop*.

J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. 2014. Learning fine-grained image similarity with deep ranking. In *Proceedings of the CVPR*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *Proceedings of the ICLR*.