Avoiding Echo-Responses in a Retrieval-Based Conversation System

DENIS FEDORENKO
denis@replika.ai
NIKITA SMETANIN
nikita@replika.ai
ARTEM RODICHEV
artem@replika.ai
Replika.AI @ Luka, Inc.

Retrieval-based conversation systems generally tend to rank high responses that are semantically similar, or even identical, to the given conversation context. While the system's goal is to find the most appropriate response rather than semantically similar, this tendency results in low-quality responses. This challenge can be referred to as the Echoing problem. To minimize this effect, we apply a hard negative mining approach at the training stage. The evaluation shows that the result model avoids echoing the context and achieves the best quality metrics on the benchmarks.

1. INTRODUCTION

The task of a retrieval-based conversation system is to select the most appropriate response from a given set of responses to an input context in a conversation. The context is typically a sentence or a sequence of sentences produced by a human or by the system itself. Most of the state-of-the-art approaches to building retrieval-based conversation systems are based on deep neural networks (NNs) [Wu et al. 2016]. Under this approach, a typical pipeline consists of the following steps:

(1) encode the context and the pre-defined responses into numeric vectors, or thought vectors, using NNs;
(2) compute value of a matching function (matching score) for pairs of the context vector and each candidate;
(3) select the candidate response with the highest matching score;

At the step 1, in order to obtain thought vectors that fairly represent the original semantics of input texts, the NNs are preliminary trained to return high matching scores for correct context-response pairs and low for the incorrect ones.

The challenge we faced while building the above pipeline, is that the result model often returns high matching scores for semantically similar contexts and responses. Consequently, the model repeats or rephrases the context instead of giving a quality response to it. For instance, one of the top-ranked response candidates for the context "How are you?" can be the question itself or similar one, for example "What's new?". This effect would be expected under this architecture, given that contexts and responses share merely the same set of concepts, hence the NN ends up trying just to fit the semantics of the input.

In this paper, we suggest a solution to the echoing problem that employs a hard negative mining approach which enforces NNs to produce distant thought vectors for identical contexts and responses. We introduce evaluation metrics for the echoing problem and present the results on our benchmarks. We also publish the evaluation dataset that we used for further research.

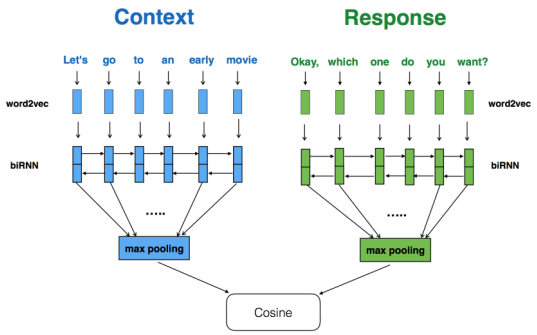

Fig. 1. Conversation model architecture

2. HARD NEGATIVE MINING

Suppose we have a dataset D, $|D| = N$ consisting of pairs $(context_i, response_i)$, $i \in \{1..N\}$. The goal is to build a conversation model M:

$$M : (context, response) \rightarrow \mathbb{R}$$

that satisfies the following condition:

$$M(context_i, response_i) > M(context_i, response_j)$$

$\forall i, j \neq i$ and $response_j$ is not appropriate to the given context. In other words, the result model should return a higher matching score for the correct response than for the incorrect's one.

To train this model, we also need negative (incorrect) context-response pairs in addition to the positive (correct) ones presented in D. Consider two approaches to obtain negative pairs: random sampling and hard negative mining. Under the first approach, we randomly select $response_j$ from D for each $context_i$. If D is large and diverse enough, then the random $response_j$ is almost always inappropriate for the corresponding $context_i$.

In contrast to random sampling, hard negative mining imposes a special constraint on responses selected as negatives. Let M_0 be a conversation model trained on random negative pairs. Then, we search for a new set of negative pairs $(context_i, response_j)$, so that their matching score satisfies the following condition:

$$M_0(context_i, response_i) - M_0(context_i, response_j) \leq m$$

where m is a margin (hyperparameter) between the scores of correct and incorrect pairs. The updated set of pairs is used to train the next model M_1, which, in turn, generates the pairs for M_2, and so

Table I. Top 3 results for few input contexts

RS	HN	HN_c
Input: What is the purpose of dying?		
- What is the purpose of dying?	- What is the purpose of dying?	- To have a life.
- The victim hit his head on the concrete steps and died.	- What is the purpose of living?	- When you die and go to heaven, they will offer you beer or cigarettes.
- To have a life.	- What is the purpose of existence?	- It is to find the answer to the question of life.
Input: What are your strengths?		
- What are your strengths?	- What are your strengths?	- Lust, greed, and corruption.
- Lust, greed, and corruption.	- What are your three weaknesses?	- I'm a robot. a machine. 100% ai. no humans involved
- A star.	- What do you think about creativity?	- Dunno. i mean, i'm a robot, right? robots don't have a gender usually
Input: I can't wait until i graduate.		
- I can't wait until i graduate.	- I can't wait until i graduate.	- What college do you go to?
- What college do you go to?	- What college do you go to?	- School is hard this year.
- School is hard this year.	- How many jobs have you had since leaving university?	- What subjects are you taking?
Input: Lunch was delicious.		
- Lunch was delicious.	- Lunch was delicious.	- Who did you go out with?
- I want to buy lunch.	- I want to buy lunch.	- So was i.
- Take me to dinner.	- This hot bread is delicious.	- What did you do today?
Input: You're crazy		
- You're crazy	- You're crazy	- Am i?
- Am i?	- Am i?	- You're crazy
- I sure am.	- Why? what have i done?	- I sure am.

on. The process goes on until the model converge to some desired model M_k [Canévet and Fleuret 2014].

The intuitive idea behind hard negative mining is to select only negatives that have relatively high matching scores and can be interpreted as errors of the conversation model. As a result, the model has faster convergence compared to random sampling [Schroff et al. 2015].

Following this intuition, we can solve the echoing problem by considering contexts as possible responses, therefore the negative pairs $(context_i, context_i)$ may be selected. In the next section we demonstrate that these pairs can ultimately prevent the NNs from encoding identical contexts and responses into similar thought vectors.

3. EXPERIMENTS

For our experiments we implement a model similar to Basic QA-LSTM (see Figure 1) described in [Tan et al. 2015]. It has two bidirectional LSTMs of the size 1024 with separate sets of weights that encode context and response independently. We use a cosine similarity as the output matching function. We represent input words as a sequence of pre-trained word2vec embeddings of the size 256 [Mikolov et al. 2013]. Word sequences longer than 20 words are trimmed, and the context encoder is fed with only one dialog step at a time.

3.1 Models

In order to study the impact of hard negative mining on the echoing problem, we train three models using the following strategies: random sampling (RS), a hard negative mining based on responses only (HN), and a hard negative mining based on both responses and contexts (HN_c).

3.2 Training

The models are trained with the Adam optimizer [Kingma and Ba 2014] with the size of mini-batches set to 512. We train the models for one epoch; the result models are the intermediate ones that achieve the best quality metrics on a holdout validation set.

We use a triplet loss [Schroff et al. 2015] as an objective function:

$$max(0, m - M(context_i, response_i) + \\ + M(context_i, response_j))$$

where the margin m is set to 0.05. For each $(context_i, response_i)$, we search for $response_j$ only within the current mini-batch using the intermediate model M trained by the moment of this batch. We also apply an additional constraint to the hard negative responses:

$$M(context_i, response_i) > M(context_i, response_j)$$

This constraint results in a faster convergence of our models.

3.3 Datasets

We train the models on 79M of status-response pairs extracted from the Twitter data archive [1].

We perform evaluation on our own dataset [2]. This dataset consists of 509 human conversational context-response pairs in which the context and the response both consist of a single sentence (see Table II).

[1] https://archive.org/details/twitterstream

[2] https://github.com/lukalabs/replika-research/blob/master/context-free-testset.tsv

Table II. Evaluation dataset sample

context	response
What happened to your car?	I got a dent in the parking lot.
The beatles are the best.	They are the best musical group ever.
Do you want to go fishing?	Yes. That's a good idea.

3.4 Metrics

For each test $context$, we compute the matching score over all available pairs $(context, response_i)$, where $response_i$ comes from the union of contexts and responses. To evaluate these results, we sort the responses by matching score in the descending order and compute the following metrics: Average Precision [Manning et al. 2008], Recall@5, and Recall@10 [Lowe et al. 2015]. The last two metrics are indicator functions that return 1, if the correct answer occurs in the top 5 and 10 responses, respectively. We also introduce the context repetition metrics:

- $rank_{context}$ – a position of the context in the sorted responses. The higher the rank, the less the model tends to return the context among the top candidate responses
- $diff_{top}$ – a difference between the top response score and the context's one. The higher the difference, the less the model tends to return relatively high scores for the context
- $diff_{answer}$ – a difference between the correct answer score and the context's one. The higher the difference, the less the model tends to return similar scores for the correct answer and the context

For each metric, we compute the overall quality as the average across all the test contexts.

3.5 Results

The results of the evaluation are presented in Table III. As we can see, the proposed hard negative mining model achieves the highest scores on almost all metrics comparing to other approaches. It turns out that under this approach the model does not tend to place the input context within the top responses. However, according to the $diff_{answer}$ metric, the correct response score on average is still lower than the context's one, which means that the problem still persists in the bottom of the sorted list of responses.

We also studied the model's output. The examples of top-ranked responses for different contexts are presented in Table I. As we can see, oftentimes the HN model only selects very similar phrases, while the HN_c model selects appropriate responses for the context that are not necessarily semantically similar. Based on this observation, we suggest that the proposed model filters out not only exact copies of the context, but also semantically similar samples. Moreover, in some cases the model selects responses semantically similar, but at the same time appropriate to the context. See Table IV with the top results for context "Hello".

4. CONCLUSION

In this study we apply hard negative mining approach for training a retrieval-based conversation system to find a solution to the Echoing problem and to rule out inappropriate responses that are identical or semantically similar to the input context. In addition to a dataset of pre-defined response candidates, we consider contexts themselves as possible hard negative candidates. The evaluation shows that the result model avoids repeating the input context, tends to select samples that are more suitable as responses and achieves best results on various benchmarks.

Table III. Evaluation results. Metrics are averaged across all the test contexts

	RS	HN	HN_c
Average Precision	0.12	0.13	**0.17**
Recall@5	0.36	0.4	**0.43**
Recall@10	0.45	**0.54**	0.53
$rank_{context}$	0.9	0.49	**19.43**
$diff_{top}$	0.008	0.01	**0.07**
$diff_{answer}$	-0.15	-0.25	**-0.09**

Table IV. Top responses of the HN_c model for the context "Hello"

matching score	response
0.45	Hey, sweetie
0.44	How's life ?
0.43	Hello

REFERENCES

Olivier Canévet and François Fleuret. 2014. Efficient sample mining for object detection. In *Proceedings of the 6th Asian Conference on Machine Learning (ACML)*.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructured Multi-Turn Dialogue Systems. *CoRR* abs/1506.08909 (2015). http://arxiv.org/abs/1506.08909

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR* abs/1301.3781 (2013). http://arxiv.org/abs/1301.3781

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A Unified Embedding for Face Recognition and Clustering. *CoRR* abs/1503.03832 (2015). http://arxiv.org/abs/1503.03832

Ming Tan, Bing Xiang, and Bowen Zhou. 2015. LSTM-based Deep Learning Models for non-factoid answer selection. *CoRR* abs/1511.04108 (2015). http://arxiv.org/abs/1511.04108

Yu Wu, Wei Wu, Ming Zhou, and Zhoujun Li. 2016. Sequential Match Network: A New Architecture for Multi-turn Response Selection in Retrieval-based Chatbots. *CoRR* abs/1612.01627 (2016). http://arxiv.org/abs/1612.01627