



Joint Retrieval-Extraction Training for Evidence-Aware Dialog Response Selection

Hongyin Luo¹, James Glass¹, Garima Lalwani², Yi Zhang², Shang-Wen Li²

¹MIT CSAIL, USA

²Amazon AI, USA

{hyluo, glass}@mit.edu, {glalwani, yizhngn, shangwel}@amazon.com

Abstract

Neural dialog response selection models infer by scoring each candidate response given the dialog context, and the cross-encoder method yields state-of-the-art (SOTA) results for the task. In the method, the candidate scores are computed by feeding the output embedding of the first token in the input sequence, which is a concatenation of response and context, to a linear layer for making prediction. However, the embeddings of the other tokens in the sequence are not modeled explicitly, and inferring the candidate scores only with the first token makes the result not interpretable. To address the challenge, we propose a Retrieval-EXtraction encoder (REX) for dialog response selection. We augment the existing first-token- or sequence- based retrieval approach with an extraction loss. The loss provides gradient signal from each token during training and allows the model to learn token-level evidence and to select response based on important keywords. We show that REX achieves the new SOTA in the dialog response selection task. Also, our qualitative analysis suggests that REX highlights evidence it infers selections from and makes the inference result interpretable.

1. Introduction

Dialog response selection is an important task in a complete dialog system to retrieve human generated texts or re-rank machine generated responses. Given an input query and candidate texts, an encoder is applied to encode, score, and rank the query-candidate pairs. To achieve this goal, the models are optimized for assigning higher scores to the ground-truth candidates as compared to negative candidates during training [1, 2].

State-of-the-art (SOTA) dialog response selection models employ the Transformer architecture [3] with pre-trained parameters on different data for various downstream tasks, such as BERT [4] and ConveRT [5]. The model architecture that achieves the best selection performance with a pre-trained transformer model is called cross-encoder [6]. By concatenating the input query with each candidate into a sequence and encoding them jointly with a transformer network, cross-encoder yields better representations for the query-candidate pairs. The candidate score of a sequence is obtained by feeding the output embedding of the first token of the entire sequence (e.g. [CLS] in BERT) into a linear layer. The sequence which is assigned with the highest score among all candidates is selected as the retrieved result. Under this setting, the information encoded in positions (or tokens) other than the first is not used explicitly, which makes it hard to interpret the prediction result.

Conversely, extractive question answering (QA) and reading comprehension [7, 8] models apply similar architecture for sequence scoring but utilize output embeddings of all the tokens during training and inference. Each question and context passage



: I just watched the movie "The Wandering Earth".

Oh, is its story authored by **Liu Cixin**? 👍

Sounds good! 😊

It's a good restaurant on the Main st! 🤔



Figure 1: An example of the dialog response selection, where dialog history and three candidate responses are shown. The phrase in red is the evidence supporting the first candidate as the best response for the given dialog history (Liu Cixin is the author of the The Wandering Earth). Human tell the best response by reasoning the evidence in candidate.

are concatenated into a sequence, and embedding of each token in the sequence is fed into feed-forward and softmax layers to compute token score. These token scores are used to extract the span of tokens that answer the question given the context. The question is predicted as not answerable given the context if the [CLS] token yields the maximum score. The architecture suggests that token-level (i.e., span extraction) and sequence-level (i.e., answerable or not) tasks can be modeled jointly.

Motivated by the joint modeling of token- and sequence-level information, we propose a novel Retrieval-EXtraction (REX) training strategy for dialog response selections. In REX, we design two objectives for retrieval and extraction, which guides the model to select the right responses and attend to the relevant token-level evidence respectively. As compared to the conventional cross-encoder, the evidence attention improves both performance and interpretability. However, most dialog response selection corpora annotate no token-level evidence. To resolve the issue, we generate extraction labels from the response retrieval task unsupervisedly to encourage the model attending to tokens properly. Our approach is evaluated with the ConveRT and DSTC7 Track 1 challenges. The results show that REX significantly outperforms previous SOTA. In summary, our main contributions are 3-fold: 1) We propose a combined retrieval and extraction training method for cross-encoder models (REX-encoder). 2) We design an unsupervised extraction-label generation to encourage the attention to token-level evidence. 3) We show REX achieves the new SOTA in dialog response selection corpora, ConveRT and DSTC7, and the learned evidence attention improves interpretability.

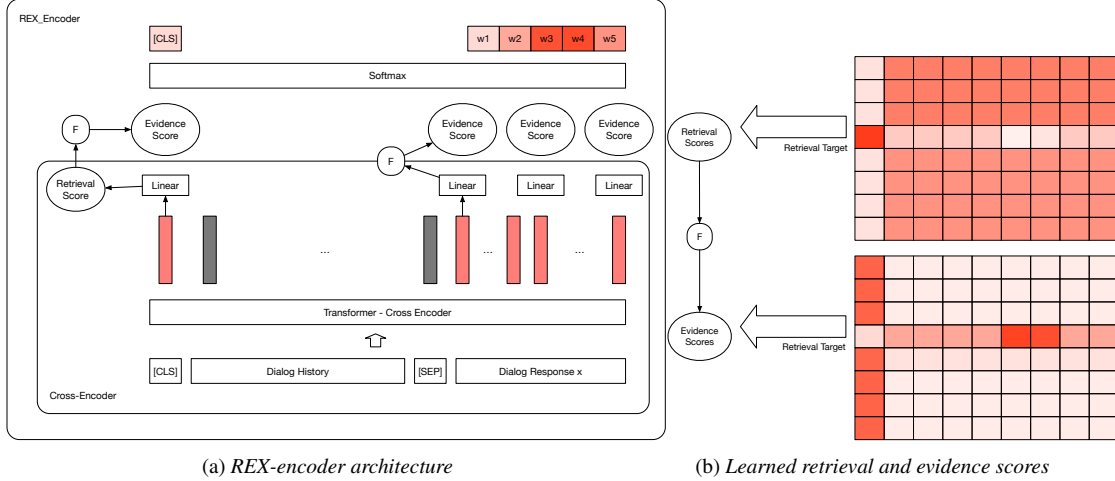


Figure 2: The REX-encoder and an example of learned retrieval and extraction scores. 2a. The inner block is the typical cross-encoder architecture. The REX encoder consists of a cross-encoder model and an unsupervised extraction module. 2b. Retrieval and extraction are opposite to each other. The first token of a true response should receive a high retrieval score and a low extraction score.

2. Related Work

There are many tasks that adopt the setup of scoring or ranking a set of candidates given some query (or context). One of the most common tasks is Dialog Response Selection, where the query is the dialog history and the candidates are the potential responses. Recently, it has become popular to use the text retrieval task as one of the pre-training strategies for learning powerful pre-trained representations that are leveraged in conversational systems beyond BERT. [9, 5, 6] use dialogue response selection objective and huge Reddit conversational corpus to pre-train conversation encoders. Experiments show the encoders yield better performance as compared to general BERT representations pre-trained with Wikipedia data. [10] train networks with a combination of response contrastive loss and masked language modeling loss to learn a more task-oriented dialog pre-trained model initialized from BERT parameters. The resulting pre-trained models are widely applied in the domain of goal-oriented dialog systems [11, 12, 13, 14, 15, 16], end-to-end dialog generation [17, 18, 19], and dialog state tracking [20, 21].

Besides dialogs, sequence scoring is directly applicable to Open Domain QA [22, 23]. In the task, the first step is to retrieve top relevant context passages or documents that may contain the answer, and then applying machine comprehension models to extract token span for answers [24, 25]. Recent approaches like [23, 26] leverage pre-trained transformer models to encode question and documents, and retrieve relevant documents using cosine similarity between encodings of documents and questions. More recently, sequence scoring is also used in obtaining language agnostic sentence representations [27] by ranking various target sentences in cross-lingual embedding space.

3. REX Encoder

3.1. Bridging Retrieval and Extraction

In this work, we propose REX encoder, an encoder combining supervised **R**etrieval and unsupervised **E**xtraction losses based on cross encoders [6], for dialog response selection. The retrieval and extraction task respectively guides the model to rank candidate texts and to attend to supporting evidence. In REX, we start with concatenating a candidate response R^i to the

dialog history H as a sequence s^i , and encoding s^i for its embedding $E^i = T([CLS], Q, [SEP], R^i, [SEP])$, where T is a transformer encoder network¹. $E^i = [e_0^i, e_1^i, \dots, e_{|Q|+|R^i|+2}^i]$ is a list of output embedding for each token in s^i . REX then learns retrieval- and extraction- probabilities jointly by assuming that when the training converges, both retrieval and extraction probabilities (P_R and P_E) meet the following two constraints. First, if s^i contains the true (i.e., positive) response, $P_R(i) = P_E(k)^i = 1, \forall k > 0$. Here, $P_R(i)$ is the probability of s^i containing the true response, and $P_E(k)^i = 1, \forall k > 0$ indicates that a token from the response or history (other than [CLS]) is attended as the supporting evidence. Second, if s^i contains a negative response (i.e., R^i does not fit into H), the retrieval probability is 0 and the model extracts the [CLS] token as the evidence. That is, $P_R(i) = 0; P_E(0)^i = 1$.

The retrieval probability of s_i and extraction probabilities of token j in s_i are

$$P_R(i) = \frac{\exp(x_0^i)}{\sum_k \exp(x_k^i)}; P_E(j)^i = \frac{\exp(f(x_j^i))}{\sum_k \exp(f(x_k^i))} \quad (1)$$

where $x_k^i = W \cdot e_k^i$ and W is a learnable linear layer. Here, $f(x)$ is a transformation function that converts retrieval logits x_k^i to extraction logits. Given that the positive response is expected to have large $P_R(i)$ but small $P_E(0)^i$, both based on the logits from the [CLS] token, we use $f(x) = -x$ in this work.

3.2. Unsupervised Extraction Learning

In most retrieval tasks and corpora, only the ground-truth retrieval targets are annotated and there are no explicit token-level labels for supporting evidence. Thus, it is difficult to obtain supervised extraction learning signals. With this limitation, we introduce an unsupervised extraction learning, to train the model to focus on the [CLS] tokens for sequences with negative responses and attend to evidence keywords for positive sequences.

For sequences with negative responses, we expect the extraction model to focus on the [CLS] token, because there is no evidence that supports the candidate responses in the sequences

¹This architecture is the same as the one used in cross-encoder.

to be good retrieval targets. To achieve this goal, we first generate a one-hot label, $l = [1, 0, 0, \dots, 0]$, that has 1 in the position corresponding to the [CLS] token and 0 for the rest. Then, we train the extraction model with a cross-entropy loss

$$l_E^{i,neg} = -\log \frac{\exp(f(x_0^i))}{\sum_k \exp(f(x_k^i))} = -\log \frac{\exp(f(x_0^i))}{Z_E^i} \quad (2)$$

Here, the label l is used to make the model attend to the [CLS] token when negative sequences are provided.

Regarding the sequences with positive responses, the goal of the extraction model is to extract evidence keywords from the sequences instead of paying attention to the [CLS] token. In this case, a trained model should assign small values to $f(x_0^i)$, and high extraction scores to other tokens $j > 0$ in the positive sequence s^i . Since there is not evidence annotation available, in this case we train the extraction model with the following loss function, as a soft supervision to encourage the model to attend to the texts instead of the [CLS] token.

$$l_E^{i,pos} = \log Z_E^i - \log Z_E^{i,[1:N]} \quad (3)$$

where $Z_E^{i,[1:N]} = Z_E^i - \exp(f(x_0^i))$. Equation 3 is equivalent to generate a label $l = [0, 1, 1, \dots, 1]$ and training the model with cross-entropy. As we can see in both cases, we generate l automatically based on the retrieval label, and thus the extraction learning is unsupervised. To sum up, for a dialog history and a set of candidates, which contain one true response i and others as negative candidates, we propose the overall loss function for unsupervised extraction learning

$$l_E = l_E^{i,pos} + \sum_{j \neq i} l_E^{j,neg} \quad (4)$$

3.3. Overall Architecture

As mentioned above, we train the REX encoder with a combination of a supervised-retrieval and an unsupervised-extraction task. We show the complete architecture of the proposed REX-encoder model in Figure 2a along with an example of the learned retrieval and extraction scores in Figure 2b. The overall loss function for REX, $L = \alpha \cdot l_R + \beta \cdot l_E$, is a weighted combination of retrieval and extraction losses with coefficients α and β . Here l_E has been shown in Equation 4 and l_R is the standard retrieval loss based on cross-entropy².

4. Experiments

4.1. Task and Corpus

We evaluate REX and baselines with the dialog response selection performance on ConvAI2 [1] task and DSTC7 challenge Track 1 Ubuntu dialog corpus [2]. ConvAI2 is based on the persona-chat dataset [28] where each participant of a conversation is given a persona. The DSTC7 corpus contains conversations of questions and answers among users of the Ubuntu community about the system usage. ConvAI2 and DSTC7 consist of 17,878 and 135,078 training dialog threads, respectively. The annotations of both corpora only contain true responses, and the token-level evidence is not provided. We conduct supervised retrieval training with the annotated ground-truth responses and unsupervised extraction training following the approach described in Section 3.2.

² $\alpha = 1, \beta = 0$ is the standard cross-encoder approach for response selection or retrieval.

4.2. Implementation Details

We implement our model with the ParlAI framework³ based on its cross-encoder implementation. We use the BERT-large model pre-trained on the Reddit dialog corpus [29] provided by ParlAI as the default Transformer architecture and parameter initialization. Following [30], we apply the Adam [31] optimizer with 0.01 weight decay rate. We initialize the learning rate as $5e-5$ with 1000 warmup steps and the decay rate for every half epoch is 0.4. In our experiment, we set $\alpha = 1$ and $\beta = 5$. Both values are obtained by tuning on a held-out set. The authors of [30] trained the SOTA model with 8 Volta 100 GPUs using a batch size of 16. Due to the limited computation resources, we train our model on 2 Volta 100 GPUs and the largest batch size we can reach is 8. As a result, we compare our model with both the reported performance in [30] and our reproduced SOTA baseline performance using our own machines. We conduct all experiments with the settings described above if not explicitly specified. Our implementation is publicly available at <https://github.com/luohongyin/rex-encoder.git>.

4.3. Response Selection Performance

We evaluate our model on the dialog response selection task with two metrics, Recall@1 (R@1) given C candidates and mean reciprocal rank (MRR). Here C is 20 and 100 for ConvAI2 and DSTC7, respectively. The experiment results are summarized in Table 1. Since the test set of ConvAI2 is not publicly available, we evaluate our model on the development set. For fair comparison, we used officially recommended hyper-parameter settings⁴ without any tuning. Experiments show that the REX encoder outperforms the original SOTA model on ConvAI2 Dev set, and achieves the new state-of-the-art performance on the DSTC7 Track 1 challenge although we have limited computational resources and smaller batch size.

Table 1: Retrieval performance of the baseline and the proposed REX encoder on ConvAI2 and DSTC7. We compare both the reported performance trained with 8 Volta 100 GPUs and batch size 16, and the performance reproduced by training on our machine with 2 Volta 100 GPUs and batch size 8. The latter is denoted as Cross-encoder (Ours). The experiment results of bi-, poly-, and cross-encoders are reported in [30].

Models	R@1	MRR
<i>ConvAI2-Dev</i>		
Cross-encoder (Ours)	90.0	94.1
Rex-encoder	90.5	94.3
<i>DSTC7</i>		
[32]	60.8	69.1
[33]	64.5	73.5
Bi-encoder	70.9	78.1
Poly-encoder	71.4	78.3
Cross-encoder	71.7	79.0
Cross-encoder (Ours)	70.8	78.0
Rex-encoder	72.9	79.4

Analyzing the performance difference between approaches based on R@1 scores on DSTC7 corpus suggests that REX significantly outperforms others. The improvement achieved by

³<https://parl.ai/>

⁴<https://parl.ai/projects/polyencoder/>

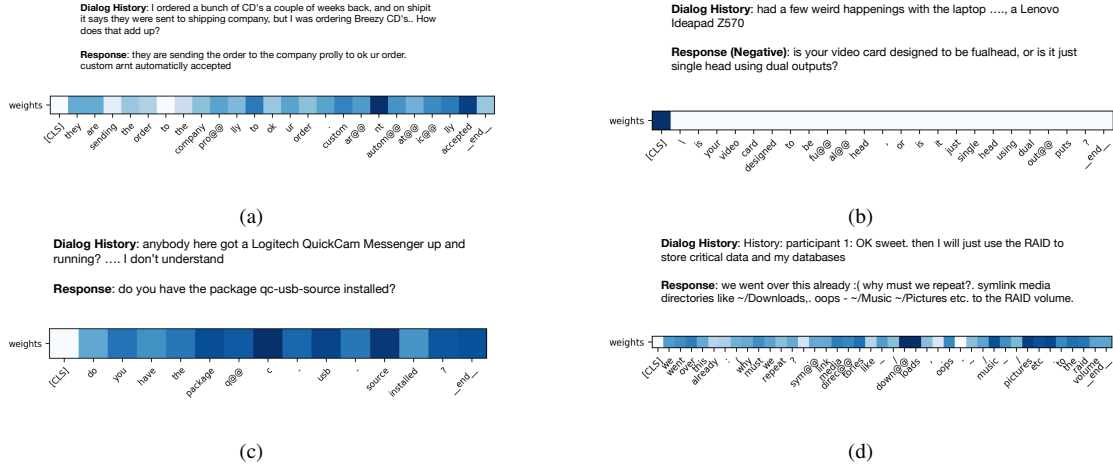


Figure 3: Visualization of the learned extraction scores on examples randomly sampled from the DSTC7 test set. 3a, 3c, 3d. The model attends to the evidence in the true responses. 3b. The model attends to the [CLS] token for a negative candidate.

Table 2: Comparing the REX-encoder with baseline models that also explicitly use all the output embeddings of the cross-encoder with different pooling methods.

Strategies	R@1	MRR
[CLS]	70.8	78.0
S-mean	70.1	77.2
S-max	69.8	77.3
E-mean	71.1	78.4
E-max	70.7	78.2
REX	72.9	79.4

applying the poly-encoder instead of bi-encoder is 0.5%, while the SOTA cross-encoder model outperforms the bi-encoder by 0.8%. Meanwhile, REX outperforms our implementation of cross-encoder by 2.1%, and even the reported SOTA performance of cross-encoders by 1.2%, which is still larger than both the reported Cross- vs. Bi-encoder and Poly- vs. Bi-encoder performance gaps.

4.4. REX vs. Pooling

As we discussed above, one advantage of the REX-encoder model is that REX utilizes the output embedding of each token during training. There are other simple ways to use all the embeddings such as pooling. To show that the unsupervised extraction learning provides additional benefit as compared to the simple approach, we compare the REX-encoder to cross-encoder with the following pooling methods,

- Scores mean and max pooling (S-mean/max): feed the output embedding of each token to a linear scorer, and calculate the retrieval score of the sequence by pooling the output scores of the linear layer.
- Embedding mean and max pooling (E-mean/max): pool the output embedding of each token, and calculate the retrieval score of the sequence by feeding the pooled sequence embedding to a linear layer.

The above strategies make it possible for the cross-encoder model to use the output embedding of each token from the input sequence explicitly. Also, we use [CLS] pooling to denote the original cross-encoder model.

Table 2 shows the experiment results for REX and various pooling strategies on DSTC7. REX significantly outperforms pooling with cross-encoder. Except for the E-mean strategy that yields slightly higher accuracy, all the other mean/max pooling strategies perform worse than the cross-encoder with standard [CLS] pooling. The result suggests that the embedding of each token cannot provide additional information and improve the sequence-level response prediction without the extraction training signal introduced by REX.

4.5. Evidence Extraction

We further analyze the quality of the learned extraction signal. We randomly sample sequences from the test set of DSTC7, calculate the extraction attention on each token from the sequences using the learned REX model, and visualize the attention in Figure 3. The visualization shows that the model attends to the words in response text that are coherent with the dialog history. The attention indicates the evidence used by the model for prediction and improves interpretability. In Example 3a, the model focuses on words "aren't" and "accepted" to answer the shipment question. The solution to the question raised in the dialog of Example 3c is "qc-usb-source", and the model successfully attends to each subwords of them. In Example 3d, the model focuses on the words "download" and "pictures", which are coherent with the dialog discussing about storage of critical data. As for a negative response, the model attends to the [CLS] token as demonstrated in Figure 3b.

5. Conclusions

In this work, we propose a supervised-Retrieval, unsupervised-EXtraction (REX) method based on the cross-encoder transformer neural network to improve the accuracy and interpretability of dialog response selection. Experiments showed that REX significantly outperforms the cross-encoder baseline and achieves the new SOTA performance on ConvAI2 and the DSTC7 Track 1 challenge with the same number of trainable parameters. The proposed unsupervised extraction training yields better performance than common poolings. Visualizing the extraction results demonstrates that the model attends to evidence keywords helping to decide whether the candidate is a good response, and thus enhances interpretability.

6. References

- [1] E. Dinan, V. Logacheva, V. Malykh, A. Miller, K. Shuster, J. Urbanek, D. Kiela, A. Szlam, I. Serban, R. Lowe *et al.*, “The second conversational intelligence challenge (convai2),” in *The NeurIPS’18 Competition*. Springer, 2020, pp. 187–208.
- [2] C. Gunasekara, J. K. Kummerfeld, L. Polymenakos, and W. Lasecki, “DSTC7 task 1: Noetic end-to-end response selection,” in *Proceedings of the First Workshop on NLP for Conversational AI*, Aug. 2019.
- [3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998–6008, 2017.
- [4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019, pp. 4171–4186.
- [5] M. Henderson, I. Casanueva, N. Mrkšić, P.-H. Su, T.-H. Wen, and I. Vulić, “Convert: Efficient and accurate conversational representations from transformers,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, 2020, pp. 2161–2174.
- [6] S. Humeau, K. Shuster, M.-A. Lachaux, and J. Weston, “Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring,” *arXiv: Computation and Language*, 2019.
- [7] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, “Squad: 100,000+ questions for machine comprehension of text,” in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 2383–2392.
- [8] S. Gao, A. Sethi, S. Agarwal, T. Chung, and D. Hakkani-Tur, “Dialog state tracking: A neural reading comprehension approach,” in *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, 2019, pp. 264–273.
- [9] M. Henderson, I. Vulić, D. Gerz, I. Casanueva, P. Budzianowski, S. Coope, G. Spithourakis, T.-H. Wen, N. Mrkšić, and P.-H. Su, “Training neural response selection for task-oriented dialogue systems,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 5392–5404.
- [10] C.-S. Wu, S. C. Hoi, R. Socher, and C. Xiong, “Tod-bert: Pre-trained natural language understanding for task-oriented dialogue,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 2020, pp. 917–929.
- [11] H. Luo, S.-W. Li, and J. Glass, “Knowledge grounded conversational symptom detection with graph memory networks,” *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pp. 136–145, 2020.
- [12] O. Ramadan, P. Budzianowski, and M. Gasic, “Large-scale multi-domain belief tracking with knowledge sharing,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 2018, pp. 432–437.
- [13] H. Luo, M. Mohtarami, J. R. Glass, K. Krishnamurthy, and B. Richardson, “Integrating video retrieval and moment detection in a unified corpus for video question answering,” in *INTERSPEECH*, 2019, pp. 599–603.
- [14] T.-H. Wen, D. Vandyke, N. Mrkšić, M. Gasic, L. M. R. Barahona, P.-H. Su, S. Ultes, and S. Young, “A network-based end-to-end trainable task-oriented dialogue system,” in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, 2017, pp. 438–449.
- [15] H. Luo, S.-W. Li, and J. Glass, “Prototypical q networks for automatic conversational diagnosis and few-shot new disease adaption,” in *INTERSPEECH*, 2020, pp. 3895–3899.
- [16] S.-W. Li, J. Krone, S. Dong, Y. Zhang, and Y. Al-onazian, “Meta learning to classify intent and slot labels with noisy few shot examples,” in *2021 IEEE Spoken Language Technology Workshop (SLT)*, 2021, pp. 1004–1011.
- [17] T. Zhao, K. Xie, and M. Eskenazi, “Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019, pp. 1208–1218.
- [18] B. Liu, G. Tur, D. Hakkani-Tur, P. Shah, and L. Heck, “Dialogue learning with human teaching and feedback in end-to-end trainable task-oriented dialogue systems,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2018, pp. 2060–2069.
- [19] T.-H. Wen, Y. Miao, P. Blunsom, and S. Young, “Latent intention dialogue models,” in *International Conference on Machine Learning*. PMLR, 2017, pp. 3732–3741.
- [20] J. D. Williams, “Web-style ranking and slu combination for dialog state tracking,” in *15th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 2014, p. 282.
- [21] S. Li, J. Cao, M. Sridhar, H. Zhu, S.-W. Li, W. Hamza, and J. McAuley, “Zero-shot generalization in dialog state tracking through generative question answering,” in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021, pp. 1063–1074.
- [22] D. Chen, A. Fisch, J. Weston, and A. Bordes, “Reading wikipedia to answer open-domain questions,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017, pp. 1870–1879.
- [23] K. Lee, M.-W. Chang, and K. Toutanova, “Latent retrieval for weakly supervised open domain question answering,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 6086–6096.
- [24] D. Chen, A. Fisch, J. Weston, and A. Bordes, “Reading wikipedia to answer open-domain questions,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017, pp. 1870–1879.
- [25] H. Luo, S.-W. Li, S. Yu, and J. Glass, “Cooperative learning of zero-shot machine reading comprehension,” *arXiv preprint arXiv:2103.07449*, 2021.
- [26] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, “Dense passage retrieval for open-domain question answering,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 2020, pp. 6769–6781.
- [27] F. Feng, Y.-F. Yang, D. M. Cer, N. Arivazhagan, and W. Wang, “Language-agnostic bert sentence embedding,” *ArXiv*, vol. abs/2007.01852, 2020.
- [28] S. Zhang, E. Dinan, J. Urbanek, A. Szlam, D. Kiela, and J. Weston, “Personalizing dialogue agents: I have a dog, do you have pets too?” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 2018, pp. 2204–2213.
- [29] P.-E. Mazare, S. Humeau, M. Raison, and A. Bordes, “Training millions of personalized dialogue agents,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 2775–2779.
- [30] K. Shuster, J. Urbanek, E. Dinan, A. Szlam, and J. Weston, “Deploying lifelong open-domain dialogue learning,” *arXiv preprint arXiv:2008.08076*, 2020.
- [31] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.
- [32] J.-C. Gu, Z.-H. Ling, Y.-P. Ruan, and Q. Liu, “Building sequential inference models for end-to-end response selection,” *arXiv preprint arXiv:1812.00686*, 2018.
- [33] Q. Chen and W. Wang, “Sequential attention-based network for noetic end-to-end response selection,” *arXiv preprint arXiv:1901.02609*, 2019.