A Appendices

A.1 Details of Test Sets

Statistics	Wizard of Wikipedia		CMU_DoG
	Test Seen	Test Unseen	Test
Avg. # turns	9.0	9.1	12.4
Avg, # words per turn	16.4	16.1	18.1
Avg. # knowledge entries	60.8	61.0	31.8
Avg. # words per knowledge	36.9	37.0	27.0

Table 6: The statistics of test sets of two benchmarks.

We tested our proposed method on the Wizard-of-Wikipedia (WoW) (Dinan et al., 2019) and CMU_DoG (Zhou et al., 2018a). Both datasets contain multi-turn dialogues grounded on a set of background knowledge and are built with crowd-sourcing on Amazon Mechanical Turk.

In the WoW dataset, one of the paired speakers is asked to play the role of a knowledgeable expert with access to the given knowledge collection obtained from Wikipedia, while the other of a curious learner. The dataset consists of 968 complete knowledge-grounded dialogues for testing. *It is worth noting that the golden knowledge index for each turn is available in the dataset.* Response selection is performed at every turn of a complete dialogue, which results in 7512 for testing in total. Following the setting of the original paper, positive responses are true responses from humans and negative ones are randomly sampled. The ratio between positive and negative responses is $1:99$ in testing sets. Besides, the test set is divided into two subsets: Test Seen and Test Unseen. The former shares 533 common topics with the training set, while the latter contains 58 new topics uncovered by the training or validation set.

The CMU_DoG data contains knowledge-grounded human-human conversations where the underlying knowledge comes from wiki articles and focuses on the movie domain. Similar to Dinan et al. (2019), the dataset was also built in two scenarios. In the first scenario, only one worker can access the provided knowledge collections, and he/she is responsible for introducing the movie to the other worker; while in the second scenario, both workers know the knowledge and they are asked to discuss the content. *Different from WoW, the golden knowledge index for each turn is unknown for both scenarios.* Since the data size for an individual scenario is small, we merge the data of the two scenarios following the setting with Zhao et al. (2019). Finally, there are 537 dialogues for testing. We evaluate the performance of the response selection at every turn of a dialogue, which results in 6637 samples for testing. We adopted the version shared in Zhao et al. (2019), where 19 negative candidates were randomly sampled for each utterance from the same set. More details about the two benchmarks can be seen in Table 6.

A.2 Baselines for Knowledge Selection

To compare the performance of knowledge selection, we choose the following baselines from Dinan et al. (2019) including (1) Random: the model randomly selects a knowledge entry from a set of knowledge entries; (2) IR Baseline: the model uses simple word overlap between the dialogue context and the knowledge entry to select the relevant knowledge; (3) BoW MemNet: the model is based on memory network where each memory item is a bag-of-words representation of a knowledge entry, and the gold knowledge labels for each turn are used to train the model; (4) Transformer: the model trains a context-knowledge matching network based on Transformer architecture; (5) Transformer (w/ pretrain): the model is similar to the former model, but the transformer is pre-trained on Reddit data and fine-tuned for the knowledge selection task.

A.3 Results of Low-Resource Setting

Ration (t)	Wizard Seen			Wizard Unseen		
	R@1	R@2	R@5	R@1	R@2	R@5
0%	89.5	96.7	98.9	69.6	85.8	96.3
10%	90.8	97.1	99.4	73.2	86.9	96.8
50%	91.5	97.1	99.3	73.9	87.9	96.9
100%	92.2	97.6	99.4	74.3	88.1	97.1

Table 7: Evaluation results of our model in the low-resource setting on the Wizard of Wikipedia data.

As an additional experiment, we also evaluate the proposed model for a low-resource setting. We randomly sample $t \in \{10\%, 50\%, 100\%\}$ portion of training data from WoW, and use the data to fine-tune our model. The results are shown in Table 7. We can find that with only 10% training data, our model can significantly outperform existing models, indicating the advantages of our pre-training tasks. With 100% training data, our model can achieve 2.7% improvement in terms of R@1 on the test-seen and 4.7% improvement on the test-unseen.