Good afternoon, everyone, I come from the Athlone Institute of Technology, Ireland. Today, my sharing topic is improved Muti-round chatbot using data augmentations.

My topic can be divided into six modules. And sharing will follow this list.

First, I want to introduce our motivation for this work. Our work aims to improve the smooth between different topics in the chatbot, to achieve the desired result, we add knowledge graph and data augmentations into our training process. The data augmentation including text matching, ner generation and conversation selection, these three methods contribute to the data augmentation.

To take advantage of knowledge graph, we draw on the knowledge selection model proposed by Lian, you can find the reference details in our paper. The knowledge manager model in charge of the prior and posterior knowledge selection. Prior knowledge distribution only includes utterances, on the contrast, posterior knowledge distribution contains response and utterances at the same time. Knowledge manager try to approximate these two models during training, as we know , the difference between training and testing is that testing does not have response. But if we can map prior and posterior module, even we do not know porsterior when testing, we can still use prior to achieve the posterior function. Because we build the link of these two modules. So this is the knowledge manager main idea.

since the different sub models, we defined three cost function. The fist one is the generation loss, we should familiar with this part, I guess. we choose cross entropy of the real response and the generated response as the first cost function. This cost function is common among text generation tasks.

Then the second cost function is the similarity between the prior and posterior knowledge modules. The last one is cross entropy between real response and selected knowledge, this cost function wish to train our model to select the most relevant knowledge according to the response.

Here we can see our training data structure. It is in json format. It has three keys, there are goal, knowledge, and conversation. The topics or domain among the conversation belong to the goal, the related knowledge graph belongs to knowledge. And the average round length of conversation is 12.

So next, I introduce the first method in our data augmentation, inspired by the dmss model, we build the knowledge selection model, wish to select the highly related knowledge to put into our backend neural network. however, you can use other short text matching algorithms to do this process.

Then I introduce the ner generation part. The entities in our training dataset, such as person name, movies name, even in the same category, the entity has various forms, our model are supposed to learn the abstract representations of the entity in the specific context, rather than remembering the entity itself. Therefore, we generalized the entities of the same category with a unique string. For instance, we replaced person name Ma long with the string of “person name”

The last one data augmentation is conversation selection, we want produce more training dataset by selecting short conversation. For example, if we set the conversation sliding window as 2 , the sample can produce 5 sub training sample for training. However, this method will cut down the information of our inputs.

Here, we want to share some tricks during the during, as the picture shows, the three cost functions have three loss value, and their value are very different. To smooth the difference, we spitted the training process into several sub process. I mean, we can train the largest loss value at first, then train three cost function together. The reason for this trick is that if we train the largest cost function at first , it can avoid our network only update big loss functions if we put them together.

To validate our idea, we did another test based on liner regression. We define mes and abs loss function. The k varies from 10 to 1000, for example, if the k equels 100, it means, mes loss value is 100 times of abs loss value, .so the higher k value, the more difference between the two loss values. We can conclude that the more difference between loss values, the degree of convergence is less effective. So it is necessary to add Regularization.

And we also conclude that more epchs to train can help the loss curve to converge. the yellow line represents epochs number is 5000 thansand, and the convergence is the best one.

Finally, we did 5 benchmarks as the figure shows. For instance, the D 4 data set implements entity generation, knowledge matching and loss regularization. And the D 1 dataset just implement entity generation.

The results are showed by this table. The best performance is the D 4 dataset.

The more results can be found here, D 4 is the best one. Comparing D 2 and D 1, loss regularization is proved effective, because it can avoid over-fitting. This result is consistent with our linear regression result

The comparison between D 2 and D 3, entity generation can improve our model generalization ability.

From D 2 and D4 , knowledge matching certainly improve the performance.

Comparing D 5 and D4, the conversation selection did not boost our model performance, because we reduce the utterance information, so it shoud be under-fitting.