1. Hello. We are Team 1. We will introduce a sentence selection chatbot for Ubuntu dialog corpus.
2. Here is the table of contents.
3. Machine learning developed many breakthrough results like classifying images, object detection, and so on.   
   Additionally, recent researches and products have introduced AI which recognizes the context of the situation.   
   For the sense of the intelligent mechanics, robots have to interact with humans properly, so they have to understand what is needed for humans.   
   As a user-friendly product, many companies launched their chatbots to solve customers requests.   
   For instance, Google launched google assistant for their android phones.   
   Banks of Korea also introduced their own clerks and encourage users to get the answer by text dialogue.
4. There are some properties for chatbots.   
   For instance chatbots have to answer similarly for inputs having similar meaning.   
   Even if we enter exchange rate of 100 dollars rather than the exchange of 100 dollars on the bank clerk, we want to get the amount of Korean won corresponding to 100 dollars.   
   Sometimes, we may enter a not-completed sentences, typos, or even abbreviations for the chatbot, but still want to get a right answer.   
   However, many chatbots don’t have that properties.   
   In this example, I succeeded to ask exchange, but failed to ask exchange rate on this clerk.
5. As the matter of context, we sometimes omit some parts of the sentence when we talk to each other.   
   In the left example, you can see that the clerk didn’t reply according to the context.  
   This clerk also failed to recognize similar usage.   
   When I just added 100 on my text, the clerk replied to make me follow the form of the input to exchange.  
   It is just a rule-based answer, so it means that this chatbot cannot recognize the similar usage.
6. Because of that, we decided to implement a chatbot which can take care about the two-person goal-oriented conversation with wrong grammar, typos, and even needless emoji.   
   Additionally, we took care about expertized terminology from task-specific domain.   
   What we want to do is let the chatbot selects the right reaction from the answer pool.  
   We originally planned to generation problem, but we turned it into the selection problem.  
   Generation task has too wide possible response pool which makes us not to expect a proper response in most cases.  
   However, selection task provides the response set, which makes chatbot to find the more accurate response.
7. For this purpose, we decided to use Ubuntu dialogue corpus.   
   This dataset is the collection of logs from Ubuntu-related chat rooms. The goal of each data is to solve an Ubuntu user’s posted problem from the two-person dialogue.  
   Since our target is to implement response sentence selection model for goal-oriented conversation, we selected this dataset.  
   Here is the formal detail. The given task is to select the next utterance from given candidate set with one hundred of sentences, where the only one of them is the correct answer. The performance of the chatbot will be decided by the accuracy of the results from Top n in 100.
8. Here is an example of the dialogue. In this dialogue, user 1 wants to recognize his ipod on the Ubuntu via gtkpod and write some songs into his device. The proper answer in this case should be a sentence what user2 experienced on that tools.  
   In the dialogue, you can see some typos and wrong grammars on the user1.  
   In the candidate set, there are some possible simple answers like No on number 12, hesitating answer on number 26, and so on.
9. In the same paper that introduced Ubuntu dialogue corpus, they introduced their baseline model to decide the right result.   
   It takes conversations and responses as a vector of words, and the recurrent neural network captures the meaning of each sentences.   
   Then it predicts the response and compares to the actual response by taking dot product.   
   As the dot product gets bigger, the prediction goes similar to the answer.   
   Finally, it converts that value to the probability.   
   Then it sorts the probability for each candidates to get the top n answers.   
   Now we will introduce our model.