

# Deep Learning CS 583 Fall

## Homework 4: Project [10 Points]

Bonus [20 Points]

Course instructor: Jia Xu

TA: Yu Yu

Choose one from the following tasks and build a Deep Learning system with the help of the task description. You can work alone or build a team of two. Scoring is based on accuracy of the system. The accuracy is only reported on the test set of the given dataset. You are free to improve the baseline models. The accuracy improvement with additional models will count as bonus points, up to 20 points, which means that the full score of this project is 10, but you can gain up to 30 points from this project including the bonus points. Note that the full score of the course is 100. Details of the scoring are outlined for each task.

The students whose interest is beyond the listed tasks have two options on homework 4:

1. You can propose your own project by emailing the course instructor with your one-page task description. Note, only appropriate projects will get accepted.
2. You can join a linguistic crowd-sourcing project, but there are no bonus points there.

1. [Named Entity Recognition by Bi-LSTM-CRF](#)
2. [Named Entity Recognition by fine-tuning BERT](#)
3. [Intent Classification A](#)
4. [Intent Classification B](#)
5. [Intent Classification C](#)
6. [Intent Classification D](#)
7. [Dialog Act Classification A](#)
8. [Dialog Act Classification B](#)
9. [Question Detection](#)
10. [Sentiment Classification-IMDB-A](#)
11. [Sentiment Classification-IMDB-B](#)
12. [Sentiment Classification-sentiment140-A](#)
13. [Sentiment Classification-sentiment140-B](#)
14. [Emotion Classification-emotionLines](#)
15. [Topic Classification-ConCET](#)
16. [Dialog Response Generator A](#)
17. [Dialog Response Generator B](#)
18. [Dialog Response Generation C](#)
19. [Dialog Response Generation D](#)
20. [Dialog Response Generation E](#)

## 1.Named Entity Recognition by Bi-LSTM+CRF

- Task description: Named Entity Recognition (NER) is an NLP problem, which involves locating and classifying named entities (people, places, organizations etc.) mentioned in unstructured text. This problem is used in many NLP applications that deal with use-cases like machine translation, information retrieval, chatbots and others. Briefly speaking, for each training sentence, we'd like to predict the "tag" of each token.
- Model: Bi-LSTM+CRF (<https://arxiv.org/pdf/1508.01991.pdf>)
- Dataset: [ner\\_dataset.csv](#)  
The tags used in datasets are in the form of "B-geo", "I-org", "O", the I-prefix indicates that the tag is inside a chunk (i.e. a noun group, a verb group etc.); the O-prefix indicates that the token belongs to no chunk; the B-prefix indicates that the tag is at the beginning of a chunk that follows another chunk without O tags between the two chunks

The entity tags used in this dataset:

Tag	Meaning	Example
geo	Geography	Britain
org	Organisation	IAEA
per	Person	Thomas
gpe	Geopolitical Entity	Pakistani
tim	Time	Wednesday
art	Artifact	Pentastar
eve	Event	Armistice
nat	Natural Phenomenon	H5N1

- Output: report the accuracy score of your model. [6 Points]
- Grading (one of the score applies based on test set accuracy):

A complete model with no accuracy report will gain 6 points, if you achieve acc as below, you will gain bonus points:

Accuracy	Score
>0.8	10 [reach the full score of the homework 4, total score 10]
>0.9	15 [gain 5 bonus points, total score 15]
>0.95	20 [gain 10 bonus points, total score 20]

**Bonus Question:** add new models to your existing systems and further boost your accuracy to more than 0.98 [20 bonus points] - gain 20 bonus points, total score 30.

## 2. Named Entity Recognition by fine-tuning BERT

- Task description: Named Entity Recognition (NER) is an NLP problem, which involves locating and classifying named entities (people, places, organizations etc.) mentioned in unstructured text. This problem is used in many NLP applications that deal with use-cases like machine translation, information retrieval, chatbots and others. Briefly speaking, for each training sentence, we'd like to predict the "tag" of each token.
- Model: BERT. Bidirectional Encoder Representations from Transformers (BERT) is a transformer-based machine learning technique for natural language processing (NLP) developed by Google.  
([https://huggingface.co/transformers/model\\_doc/bert.html#bertfortokenclassification](https://huggingface.co/transformers/model_doc/bert.html#bertfortokenclassification))
- Dataset: [ner\\_dataset.csv](#)  
The tags used in datasets are in the form of "B-geo", "I-org", "O", the I-prefix indicates that the tag is inside a chunk (i.e. a noun group, a verb group etc.); the O-prefix indicates that the token belongs to no chunk; the B-prefix indicates that the tag is at the beginning of a chunk that follows another chunk without O tags between the two chunks

The entity tags used in this dataset:

Tag	Meaning	Example
geo	Geography	Britain
org	Organisation	IAEA
per	Person	Thomas
gpe	Geopolitical Entity	Pakistani
tim	Time	Wednesday
art	Artifact	Pentastar
eve	Event	Armistice
nat	Natural Phenomenon	H5N1

- Output: report the accuracy or f1 score of your model. [6 Points]
- Grading:

A complete model with no accuracy report will gain 6 points, if you achieve acc as below, you will gain bonus points:

Either accuracy or F1 satisfying the following:

Accuracy	F1	Score
>0.8	>0.6	10 [reach the full score of the homework 4, total score 10]
>0.9	>0.7	15 [gain 5 bonus points, total score 15]
>0.95	>0.8	20 [gain 10 bonus points, total score 20]

**Bonus Question:** add new models to your existing systems and further boost your accuracy to more than 0.98 or F1>0.9 Score 30 - [gain 20 bonus points + full score 10 = total score 30].

### 3.Intent Classification A

- Task Description: Intent Classification is the task of correctly labeling a natural language utterance from a predetermined set of intents. Simply speaking, utterances (or sentences) are the input and we want to predict the intent tag for each utterance.
- Model: SVM (linear support vector machine)
- Dataset:  
Name: clinc150 dataset  
Access: you can find raw data here: <https://github.com/clinc/oos-eval>  
You can find the full list of intents [here](#)
- Output: report the (1) accuracy over the 150 intents, or (2) recall on out-of-scope utterances [6 Points]
- Grading:

A complete model with no accuracy report will gain 6 points, if you achieve acc as below, you will gain bonus points:

150-intents accuracy	Out-of-scope recall	Score
>70	>5	10 [reach the full score of the homework 4]
>80	>7	15 [gain 5 bonus points, total score 15]
>88	>10	20 [gain 10 bonus points, total score 20]

**Bonus Question:** add new models to your existing systems and further boost your accuracy to more than 0.95. Score 30 - [gain 20 bonus points + full score 10 = total score 30].

## 4. Intent Classification B

- Task Description: Intent Classification is the task of correctly labeling a natural language utterance from a predetermined set of intents. Simply speaking, utterances (or sentences) are the input and we want to predict the intent tag for each utterance.
- Model: MLP(multilayer perceptron)
- Dataset:  
Name: clinc150 dataset  
Access: you can find raw data here: <https://github.com/clinc/oos-eval>  
You can find the full list of intents [here](#)
- Output: report the (1) accuracy over the 150 intents, or (2) recall on out-of-scope utterances [6 Points]

- Grading:  
A complete model with no accuracy report will gain 6 points, if you achieve acc as below, you will gain bonus points:

150-intents accuracy	Out-of-scope recall	Score
>70	>10	10 [reach the full score of the homework 4]
>80	>15	15 [gain 5 bonus points, total score 15]
>90	>40	20 [gain 10 bonus points, total score 20]

**Bonus Question:** add new models to your existing systems and further boost your accuracy to more than 0.95. Score 30 - [gain 20 bonus points + full score 10 = total score 30].

## 5.Intent Classification C

Task Description: Intent Classification is the task of correctly labeling a natural language utterance from a predetermined set of intents. Simply speaking, utterances (or sentences) are the input and we want to predict the intent tag for each utterance.

- Model: CNN (convolutional neural networks)
- Dataset:  
Name: clinc150 dataset  
Access: you can find raw data here: <https://github.com/clinc/oos-eval>  
You can find the full list of intents [here](#)
- Output: report the (1) accuracy over the 150 intents, or (2) recall on out-of-scope utterances [6 Points]
- Grading:  
A complete model with no accuracy report will gain 6 points, if you achieve acc as below, you will gain bonus points:

150-intents accuracy	Out-of-scope recall	Score
>70	>7	10 [reach the full score of the homework 4]
>80	>10	15 [gain 5 bonus points, total score 15]
>90	>15	20 [gain 10 bonus points, total score 20]

**Bonus Question:** add new models to your existing systems and further boost your accuracy to more than 0.95. Score 30 - [gain 20 bonus points + full score 10 = total score 30].

## 6. Intent Classification D

- Task Description: Intent Classification is the task of correctly labeling a natural language utterance from a predetermined set of intents. Simply speaking, utterances (or sentences) are the input and we want to predict the intent tag for each utterance.
- Model: fine-tuning on BERT (Bidirectional Encoder Representations from Transformers is a transformer-based machine learning technique for natural language processing developed by Google)
- Dataset:  
Name: clinc150 dataset  
Access: you can find raw data here: <https://github.com/clinc/oos-eval>  
You can find the full list of intents [here](#)
- Output: report the (1) accuracy over the 150 intents, or (2) recall on out-of-scope utterances [6 Points]

- Grading:

A complete model with no accuracy report will gain 6 points, if you achieve acc as below, you will gain bonus points:

150-intents accuracy	Out-of-scope recall	Score
>70	>10	10 [reach the full score of the homework 4]
>80	>15	15 [gain 5 bonus points, total score 15]
>90	>40	20 [gain 10 bonus points, total score 20]

**Bonus Question:** add new models to your existing systems and further boost your accuracy to more than 0.95. Score 30 - [gain 20 bonus points + full score 10 = total score 30].

## 7. Dialog Act Classification A

- Task Description: Dialogue act classification is the task of classifying an utterance with respect to the function it serves in a dialogue, i.e. the act the speaker is performing.
- Model: CNN

Dataset: Switchboard dataset

Switchboard is a collection of about 2,400 two-sided telephone conversations among 543 speakers (302 male, 241 female) from all areas of the United States. A computer-driven robot operator system handled the calls, giving the caller appropriate recorded prompts, selecting and dialing another person (the callee) to take part in a conversation, introducing a topic for discussion and recording the speech from the two subjects into separate channels until the conversation was finished. About 70 topics were provided, of which about 50 were used frequently.

You can access switchboard dataset by using the convokit package:

```
[ ] from convokit import Corpus, download
    corpus = Corpus(filename=download("switchboard-corpus"))

[ ] df = corpus.get_utterances_dataframe()

[ ] df.head()
```

	timestamp		text	speaker	reply_to	conversation_id	meta.tag	vectors
id								
4325-0	None		Okay. /{D So, }	1519	None	4325-0	[[Okay. /, o], [{D So, }, qw]]	[]
4325-1	None		[ [ I guess, +	1632	4325-0	4325-0	[[[ [ I guess, +, qy^d]]	[]
4325-2	None	What kind of experience [ do you, + do you ] h...		1519	4325-1	4325-0	[[[What kind of experience [ do you, + do you ]...	[]
4325-3	None	I think, ] + {F uh, } I wonder ] if that worke...		1632	4325-2	4325-0	[[[I think, ] + {F uh, } I wonder ] if that wor...	[]
4325-4	None		Does it say something? /	1519	4325-3	4325-0	[[Does it say something? /, qy]]	[]

The “meta.tag” field contains the pair of [text, tag]

Some of the tags represent:

```
q      Question
s      Statement
b      Backchannel/Backwards-Looking
f      Forward-Looking
a      Agreements
%      indeterminate, interrupted, or contains just a floor holder (see manual)
(^u   [on anything] unrelated response (first utt is NOT response to previous q)
*      comment (followed by "[comment...]" after transcription to explain)
+      continued from previous by same speaker
@,o@,+@ incorrect transcription (can add comment to specify problem further)
^2     collaborative completion
^c     about-communication
^d     declarative question (question asked like a structural statement)
^e     [on statements] elaborated reply to y/n question
^g     tag question (question asked like a structural statement with a question tag at end)
^h     hold (often but not always after a question) ('let me think'; question in response to a question)
^m     mimic other
^q     quotation
^r     repeat self
^t     about-task
```

You can find this full tag list [here](#)

- Output: split the data into train and test set, then report your accuracy on the test set [6 Points]



- Grading:  
A complete model with no accuracy report will gain 6 points, if you achieve acc as below, you will gain bonus points:  

Accuracy	Score
>50	10 [reach the full score of the homework 4]
>65	15 [gain 5 bonus points, total score 15]
>70	20 [gain 10 bonus points, total score 20]

**Bonus Question:** add new models to your existing systems and further boost your accuracy to more than 0.80. Score 30 - [gain 20 bonus points + full score 10 = total score 30].

## 8. Dialog Act Classification B

- Task Description: Dialogue act classification is the task of classifying an utterance with respect to the function it serves in a dialogue, i.e. the act the speaker is performing.
- Model: LSTM
- Dataset: Switchboard dataset  
Switchboard is a collection of about 2,400 two-sided telephone conversations among 543 speakers (302 male, 241 female) from all areas of the United States. A computer-driven robot operator system handled the calls, giving the caller appropriate recorded prompts, selecting and dialing another person (the callee) to take part in a conversation, introducing a topic for discussion and recording the speech from the two subjects into separate channels until the conversation was finished. About 70 topics were provided, of which about 50 were used frequently.  
You can access switchboard dataset by using the convokit package:

```
[ ] from convokit import Corpus, download
    corpus = Corpus(filename=download("switchboard-corpus"))

[ ] df = corpus.get_utterances_dataframe()

[ ] df.head()
```

	timestamp		text	speaker	reply_to	conversation_id	meta.tag	vectors
id								
4325-0	None		Okay. /{D So, }	1519	None	4325-0	[[Okay. /, o], [{D So, }, qw]]	[]
4325-1	None		[ [ I guess, +	1632	4325-0	4325-0	[[[ [ I guess, +, qy^d]]	[]
4325-2	None	What kind of experience [ do you, + do you ] h...		1519	4325-1	4325-0	[[What kind of experience [ do you, + do you ]...	[]
4325-3	None	I think, } + {F uh, } I wonder ] if that worke...		1632	4325-2	4325-0	[[I think, } + {F uh, } I wonder ] if that wor...	[]
4325-4	None		Does it say something? /	1519	4325-3	4325-0	[[Does it say something? /, qy]]	[]

The “meta.tag” field contains the pair of [text, tag]  
Some of the tags represent:

```

q      Question
s      Statement
b      Backchannel/Backwards-Looking
f      Forward-Looking
a      Agreements
%      indeterminate, interrupted, or contains just a floor holder (see manual)
(^u   [on anything] unrelated response (first utt is NOT response to previous q)
*      comment (followed by "[comment...]" after transcription to explain)
+      continued from previous by same speaker
@,o@,+@ incorrect transcription (can add comment to specify problem further)
^2     collaborative completion
^c     about-communication
^d     declarative question (question asked like a structural statement)
^e     [on statements] elaborated reply to y/n question
^g     tag question (question asked like a structural statement with a question tag at end)
^h     hold (often but not always after a question) ('let me think'; question in response to a question)
^m     mimic other
^q     quotation
^r     repeat self
^t     about-task

```

You can find this full tag list [here](#)

- Output: split the data into train and test set, then report your accuracy on the test set [6 Points]
- Grading:  
A complete model with no accuracy report will gain 6 points, if you achieve acc as below, you will gain bonus points:  

Accuracy	Score
>50	10 [reach the full score of the homework 4]
>65	15 [gain 5 bonus points, total score 15]
>70	20 [gain 10 bonus points, total score 20]

**Bonus Question:** add new models to your existing systems and further boost your accuracy to more than 0.80. Score 30 - [gain 20 bonus points + full score 10 = total score 30].

## 9. Question detection

- Task Description:  
Given a sentence, predict if the sentence is a question or not  
There are two steps: 1) detect a question and 2) detect the type of the question. We provide the code for step1, and require you to finish step 2.  
Implementation for step1:

```
import pandas as pd
from pycorenlp import StanfordCoreNLP

class isQuestionBasic():

    # Init Constructor
    # Initialize StanfordCore NLP local instance on port 9000
    def __init__(self):
        self.nlp = StanfordCoreNLP('http://localhost:9000')

    # Input: Sentence to be predicted
    # Processing: 1. Uses Stanfors NLP's 'annotate' method to create Parse Tree
    # 2. Checks for occurence of 'SQ' or 'SBARQ' in the parse tree
    # Return: 1 - If sentence is question | 0 - If sentence is not a question
    def isQuestion(self, sentence):
        if '?' in sentence:
            return 1
        output = self.nlp.annotate(sentence, properties={
            'annotators': 'parse',
            'outputFormat': 'json',
            'timeout': 1000,
        })

        if ('SQ' or 'SBARQ') in output['sentences'][0]['parse']:
            return 1
        else:
            return 0
```

- Model for step2: we use Multinomial Naive Bayes and linear kernel Support Vector Machine (check [https://scikit-learn.org/stable/modules/generated/sklearn.naive\\_bayes.MultinomialNB.html](https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html) and <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>)  
Hint: use the TfidfVectorizer from sklearn to preprocess your data
- Dataset: [questions.txt](#)
- Output: split the dataset into training and test set and report the accuracy on test set [6 Points]

- Grading:  
A complete model with no accuracy report will gain 6 points, if you achieve acc as below, you will gain bonus points:

Either SVM or MNB achieves following accuracy:

Accuracy	Grade
>0.5	10 [reach the full score of the homework 4]
>0.7	15 [gain 5 bonus points, total score 15]

>0.9                      20 [gain 10 bonus points, total score 20]

**Bonus Question:** add new models to your existing systems and further boost your accuracy to more than 0.95. Score 30 - [gain 20 bonus points + full score 10 = total score 30].

## 10. Sentiment Classification-IMDB-A

- Task Description:  
We want to predict whether the review is positive (label 1) or negative (label 0) by using different classification models.
- Model: Bi-LSTM
- Dataset: [IMDB dataset](#)
- Output: split the dataset to train and test set, report the accuracy on test set [6 Points]
- Grading  
A complete model with no accuracy report will gain 6 points, if you achieve acc as below, you will gain bonus points:

Acc	Grade
>0.6	10 [reach the full score of the homework 4]
>0.75	15 [gain 5 bonus points, total score 15]
>0.85	20 [gain 10 bonus points, total score 20]

**Bonus Question:** add new models to your existing systems and further boost your accuracy to more than 0.90. Score 30 - [gain 20 bonus points + full score 10 = total score 30].

## 11. Sentiment Classification-IMDB-B

- Task Description:  
We want to predict whether the review is positive (label 1) or negative (label 0) by using different classification models.
- Model: Bi-GRU
- Dataset: [IMDB dataset](#)
- Output: split the dataset to train and test set, report the accuracy on test set [6 Points]
- Grading  
A complete model with no accuracy report will gain 6 points, if you achieve acc as below, you will gain bonus points:

Acc	Grade
>0.6	10 [reach the full score of the homework 4]
>0.75	15 [gain 5 bonus points, total score 15]
>0.85	20 [gain 10 bonus points, total score 20]

**Bonus Question:** add new models to your existing systems and further boost your accuracy to more than 0.90. Score 30 - [gain 20 bonus points + full score 10 = total score 30].

## 12. Sentiment Classification-sentiment140-A

- Task Description:  
We want to predict whether the review is positive (label 4), neutral (label 2), or negative (label 0) by using different classification models.
- Model: Bi-GRU
- Dataset: [sentiment140](#)
- Output: split the dataset to train and test set, report the accuracy on test set [6 Points]
- Grading  
A complete model with no accuracy report will gain 6 points, if you achieve acc as below, you will gain bonus points:

Acc	Grade
>0.55	10 [reach the full score of the homework 4]
>0.7	15 [gain 5 bonus points, total score 15]
>0.80	20 [gain 10 bonus points, total score 20]

**Bonus Question:** add new models to your existing systems and further boost your accuracy to more than 0.90. Score 30 - [gain 20 bonus points + full score 10 = total score 30].

### 13. Sentiment Classification-sentiment140-B

- Task Description:  
We want to predict whether the review is positive (label 4), neutral (label 2), or negative (label 0) by using different classification models.
- Model: Bi-LSTM
- Dataset: [sentiment140](#)
- Output: split the dataset to train and test set, report the accuracy on test set [6 Points]
- Grading  
A complete model with no accuracy report will gain 6 points, if you achieve acc as below, you will gain bonus points:

Acc	Grade
>0.55	10 [reach the full score of the homework 4]
>0.7	15 [gain 5 bonus points, total score 15]
>0.80	20 [gain 10 bonus points, total score 20]

**Bonus Question:** add new models to your existing systems and further boost your accuracy to more than 0.90. Score 30 - [gain 20 bonus points + full score 10 = total score 30].

### 14. Emotion Classification-emotionLines

- Task Description: We want to predict the emotion of an utterance. This is a multi-label classification problem.
- Model: CNN (convolutional neural network)
- Dataset: [Emotionlines-Friends](#)  
The utterances in the dataset were scraped from TV series Friends.  
The total seven emotion label are:Neutral, Joy, Sadness, Fear, Anger, Surprise, and Disgust.  
You can find more about the dataset [here](#)
- Output: report accuracy on the test set [6 Points]
- Grading  
A complete model with no accuracy report will gain 6 points, if you achieve acc as below, you will gain bonus points:

Acc	Grade
>20	10 [reach the full score of the homework 4]
>30	15 [gain 5 bonus points, total score 15]
>40	20 [gain 10 bonus points, total score 20]

**Bonus Question:** add new models to your existing systems and further boost your accuracy to more than 0.60. Score 30 - [gain 20 bonus points + full score 10 = total score 30].

## 15. Topic Classification-ConCET

- Task Description: The topic classifier is trained to classify a sentence into 1 of 17 classes: attraction, celebrities, chitchat, fashion, fitness, food, games, joke, literature, movie, music, news, other, pets animals, sports, tech, and weather.
- Model: Fine-tuning on pre-trained RoBERTa model (check [https://huggingface.co/transformers/model\\_doc/roberta.html#robertaforsequenceclassification](https://huggingface.co/transformers/model_doc/roberta.html#robertaforsequenceclassification) )
- Dataset: [ConCET](#)
- Output: report either accuracy on test set or macro f1 score of your model [6 Points]
- Grading:  
A complete model with no accuracy report will gain 6 points, if you achieve acc as below, you will gain bonus points:

Acc	F1	Score
>0.65	>0.5	10 [reach the full score of the homework 4]
>0.85	>0.75	15 [gain 5 bonus points, total score 15]
>0.95	>0.9	20 [gain 10 bonus points, total score 20]

**Bonus Question:** add new models to your existing systems and further boost your accuracy to more than 0.98. Score 30 - [gain 20 bonus points + full score 10 = total score 30].

## 16. Response Generation A

- Task Description: Train a “chatbot” that can generate responses given an input utterance.
- Dataset: [DailyDialog dataset](#) (Scroll down to **Download** section)
- Model: fine-tuning on DialogGPT (<https://huggingface.co/microsoft/DialoGPT-medium>)  
DialogGPT is a large, tunable neural conversational response generation model trained on 147M conversation-like exchanges extracted from Reddit comment chains over a period spanning from 2005 to 2017. DialoGPT is based on GPT-2. The GPT-2 model is built out of the transformer language model and leverages a stack of masked multi-head self-attention layers.
- Output: report the perplexity score or accuracy of your model [6 Points]

- Grading:

A complete model with no accuracy report will gain 6 points, if you achieve acc as below, you will gain bonus points:

Acc	Perplexity	Score
>80	<40	10 [reach the full score of the homework 4]
>90	<25	15 [gain 5 bonus points, total score 15]
>95	<20	20 [gain 10 bonus points, total score 20]

**Bonus Question:** add new models to your existing systems and further boost your accuracy to more than 0.98. Score 30 - [gain 20 bonus points + full score 10 = total score 30].



## 17. Response Generation B

- Task Description: Train a “chatbot” that can generate responses given an input utterance.
- Model: fine-tuning on BlenderBot  
(<https://huggingface.co/facebook/blenderbot-400M-distill>)  
BlenderBot is a large pre-trained chatbot developed by facebook AI. We want to finetune on a lightweight distilled version of blenderBot.
- Dataset: [DailyDialog dataset](#) (Scroll down to **Download** section)
- Output: report the accuracy of your model [6 Points]
- Grading:  
A complete model with no accuracy report will gain 6 points, if you achieve acc as below, you will gain bonus points:

Acc	Score
>80	10 [reach the full score of the homework 4]
>90	15 [gain 5 bonus points, total score 15]
>95	20 [gain 10 bonus points, total score 20]

**Bonus Question:** add new models to your existing systems and further boost your accuracy to more than 0.98. Score 30 - [gain 20 bonus points + full score 10 = total score 30].

## 18. Response Generation C

- Task Description: Train a “chatbot” that can generate responses given an input utterance.
- Model: use transformer architecture to train a chatbot from scratch
- Dataset: [DailyDialog dataset](#) (Scroll down to **Download** section)
- Output: report the accuracy of your model [6 Points]
- Grading:  
A complete model with no accuracy report will gain 6 points, if you achieve acc as below, you will gain bonus points:

Acc	Score
>80	10 [reach the full score of the homework 4]
>90	15 [gain 5 bonus points, total score 15]
>95	20 [gain 10 bonus points, total score 20]

**Bonus Question:** add new models to your existing systems and further boost your accuracy to more than 0.98. Score 30 - [gain 20 bonus points + full score 10 = total score 30].

## 19. Response Generation D

- Task Description: Train a “chatbot” that can generate responses given an input utterance.
- Model: use transformer architecture to train a chatbot from scratch
- Dataset: [Empathetic Dialog](#)
- Output: report the accuracy of your model [6 Points]
- Grading:  
A complete model with no accuracy report will gain 6 points, if you achieve acc as below, you will gain bonus points:

Acc	Score
>80	10 [reach the full score of the homework 4]
>90	15 [gain 5 bonus points, total score 15]
>95	20 [gain 10 bonus points, total score 20]

**Bonus Question:** add new models to your existing systems and further boost your accuracy to more than 0.98. Score 30 - [gain 20 bonus points + full score 10 = total score 30].

## 20. Response Generation E

- Task Description: Train a “chatbot” that can generate responses given an input utterance.
- Model: use transformer architecture to train a chatbot from scratch
- Dataset: [Topical Chat](#)
- Output: report the accuracy of your model [6 Points]
- Grading:  
A complete model with no accuracy report will gain 6 points, if you achieve acc as below, you will gain bonus points:

Acc	Score
>80	10 [reach the full score of the homework 4]
>90	15 [gain 5 bonus points, total score 15]
>95	20 [gain 10 bonus points, total score 20]

**Bonus Question:** add new models to your existing systems and further boost your accuracy to more than 0.98. Score 30 - [gain 20 bonus points + full score 10 = total score 30].