IMPORTANT: please make sure that you check the [IMPORTANT NOTES](https://www.kaggle.com/competitions/2022-comp5046-a2/overview/important-notes) for the A2

COMP5046 Assignment 2 Overview

This assignment can be done individually or in teams of two. We strongly encourage collaboration. See the [University of Sydney collaboration rules](https://www.sydney.edu.au/students/group-work.html) webpage. If your team member does not follow the rules, please contact the unit coordinator (Dr. Caren Han) with describing the contributions of each collaborator

We strongly recommend to start working early so that you will have ample time to discover stumbling blocks and ask questions.

In this assignment, you will work on proposing and implementing a model/framework for Slot Filling (Tagging) from the in-game chat. The detailed information for the implementation and submission steps are specified in the following sections. Note that lecture notes and lab exercises would be a good starting point and baseline for the assignment [especially, Lecture 9/Lab 9] Named Entity Recognition (word-based entity tagging).

You are free to design the architecture using any of the techniques learned from our lectures and labs. For training and evaluation, we provide you with a benchmark dataset in a split of training, validation and test. You will use the training and validation set for training/validation while using the test set for leaderboard submission.

For this assignment, instead of solely focusing on achieving higher performance, having a good architecture design & implementation with detailed step-by-step justification will be important to show you’ve fully reached our expectations in this assignment.

Award

The team that produced the model with great novelty will receive the 'USydNLP - FortifyEdge Award' with the prize and write an A\*/A International Conference Paper.

For more information, please check the following pages:

1. [Important Dates](https://www.kaggle.com/competitions/2022-comp5046-a2/overview/important-dates): You can find the important due and submission dates  
2. [Data And Input](https://www.kaggle.com/competitions/2022-comp5046-a2/overview/data-and-input): The dataset desc, data pre-processing process, and input embedding process are specified  
3. [Slot Filling/Tagging model](https://www.kaggle.com/competitions/2022-comp5046-a2/overview/ner-model): The process of Slot tagging model design and implementation can be found  
4. [Testing - Leaderboard](https://www.kaggle.com/competitions/2022-comp5046-a2/overview/testing-leaderboard): You can find how to test your Slot Filling/Tagging model and how to submit your result via Kaggle Leaderboard.  
5. [Documentation](https://www.kaggle.com/competitions/2022-comp5046-a2/overview/documentation): The detailed information about writing report can be found.  
6. [Canvas Submission](https://www.kaggle.com/competitions/2022-comp5046-a2/overview/canvas-submission): You can find how to submit your assignment 2 via canvas.  
7. [Important Notes](https://www.kaggle.com/competitions/2022-comp5046-a2/overview/important-notes): Please read this important notes to make sure you are in the right direction!!!!

# Important Dates

The Important date for the assignment 2 can be summarised as follows:

* Assignment 2 **Specification Release**: 10 May 2022
* Assignment 2 [Group Release](https://docs.google.com/spreadsheets/d/1qdCHTiwel2sYcrHaYuvYb7oXDuO9DTwA-3Rl9m1kWWY/edit?usp=sharing): 10 May 2022
* Assignment 2 **Group Revision Due**: 13 May 2022
* Assignment 2 **Leaderboard**: 10 May - 04 June 2022
* Assignment 2 **Submission Due**: 05 June 2022

*All deadlines are 11.59 pm (Sydney Time - AEST).*

#### Assignment 2 Group Release

YOU MUST CONTACT YOUR GROUP MEMBER BY THE GROUP REVISION DUE. Please check the reason in the following section.

#### Assignment 2 Group Revision Due

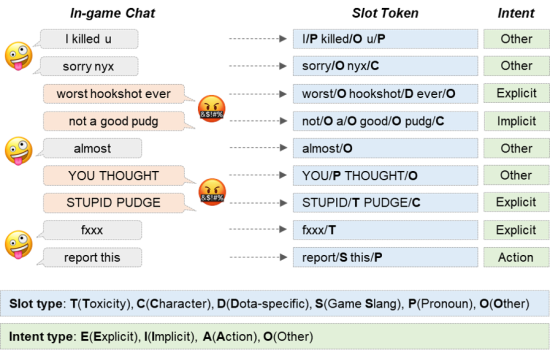
If you want to change your group to individual, you can revise it by Assignment 2 Group Change Due. However, in this case, you should get your team member's agreement.  
*NOTE: If you do not respond to your group member by the Group Revision Due, and your group member requests for working as an individual, you must work on Assignment 2 individually.*

#### Assignment 2 Leaderboard

The leaderboard will be opened from 10 May to 04 June, 2022, 11:59PM, which is one day before the assignment submission due.

# Data And Input

For this assignment 2, you MUST use PART OF the [CONDA Dataset](https://github.com/usydnlp/CONDA) (a CONtextual Dual-Annotated dataset for in-game toxicity understanding and detection) from the University of Sydney NLP Group, which focuses on in-game word slot tagging(filling) from in-game chat. You are provided with three files (train, val, test)



## **Data Download and Load**

Please go to the [Data page](https://www.kaggle.com/competitions/2022-comp5046-a2/data) in order to download and see more detailed information about the dataset.  
You MUST load the dataset via google drive. (If you want to use Jupyter notebook, you can submit your dataset in the zip file)

## **Data Preprocessing**

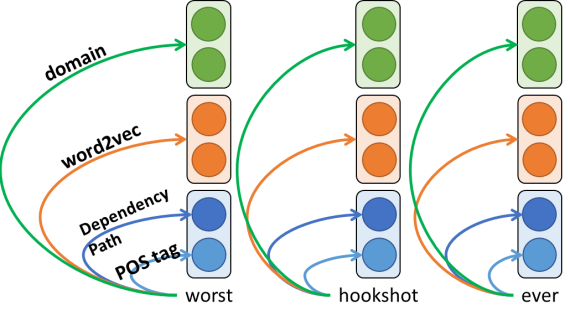
There is no restriction on pre-processing the dataset. You do not have to do data pre-processing if no need. NOTE: There is no deduction even if you don't apply any data pre-processing technique.

## **Input Embedding**

After you download the data files from the [Data page](https://www.kaggle.com/competitions/2022-comp5046-a2/data), you MUST generate the input word vector by using three(3) different types of aspect features (at least one embedding per aspect).

Aspect 1) Syntactic Textual Feature Embedding: PoS tag information, Dependency Path, etc.  
Aspect 2) Semantic Textual Feature Embedding: Word Embeddings (Word2Vec, ELMO, etc.)  
Aspect 3) Domain Feature Embedding: Your own new feature embedding to solve this in-game chat word slot filling(tagging).

You can also find some good feature examples from Lecture 9, page 24.  
For example, in the following figure, the utterance 'worst hookshot ever' is converted to the input vector by concatenating four different aspect feature embeddings - four features in total.



Your best model does not require to use of the input embedding that concatenates all 3 (or more) aspect features. For example, after you apply and test different feature combinations, if you find that using only word2vec is enough or the best. Your best model should be the word embedding only.

Note: However, you MUST submit the ipynb with all the relevant code you used (not only the best one) and the running log that you tried all the features (at least 3 aspect features) for the input embedding.

### Documentation for this section

You need to describe how to preprocess the data and justification. (If you don't have any data pre-processing, you can just skip the section saying 'No Content')

For input embedding, you MUST describe how to construct each feature embedding and justify why you chose these features for the in-game chat word slot filling/tagging task.

Relevant sections in the report: Section 1 (Data Preprocessing), Section 2 (Input Embedding)

### Code for this section

You MUST include the relevant code and running log for this section to the ipynb.

### The marking details can be found in the [documentation](https://www.kaggle.com/competitions/2022-comp5046-a2/overview/documentation) page

# Slot Filling/Tagging Model

You MUST propose and implement the In-game chat slot filling/tagging model by applying any seq2seq architecture, such as Bi-RNN, Bi-LSTM, Bi-GRU, Transformer, even BERT, etc. However, you MUST NOT directly use the existing state-of-art seq2seq architecture as it is but need to modify the seq2seq architecture and propose your own model as follows.

## **Baseline model**

Note: The baseline slot filling/tagging model (Bi-LSTM and CRF) is provided in the [Lab 09 (Bi-LSTM CRF section)](https://colab.research.google.com/drive/1biKtNR4BxPrTbZ_fMy9YwVExAh1E_ypS?usp=sharing)(the original code can be found in the [pytorch official website](https://pytorch.org/tutorials/beginner/nlp/advanced_tutorial.html)). We went through some detailed process in the lecture 9 and the lab 9.  
If you need a more detailed explanation of the code, please come to the live QA session. 😍

## **Model design**

Note: We suggest you to modify the provided baseline model in [Lab 09 (Bi-LSTM CRF section)](https://colab.research.google.com/drive/1biKtNR4BxPrTbZ_fMy9YwVExAh1E_ypS?usp=sharing) above. However, you are also welcome to use any seq2seq architecture (i.e. transformer, or Bi-GRU and MEMM). There are several ways you can modify the model but the following three components MUST be considered in your model. Only the best of your model implementation should be added in the ipynb file. NOTE: CRF should be included in the code even if your best model is that without CRF.

### 1. Stacked Seq2Seq model

You can use any sequence model but you should provide the optimal number of stacked layers.  
(NOTE: at least 2 different numbers should be tested).

If you use the RNN-based seq2seq framework, you MUST choose any of the bi-directional RNN-based models, e.g. bi-LSTM or bi-GRU, and MUST test how many layers can be stacked - check the following sample image [Lecture 4, Lecture 5, Lecture 10].

If you use Transformer [Lecture 11], you MUST decide how many encoders or decoders you would like to use. Note: You should justify the optimal number of stacked layers.

### 2. Attention

For the model, you MUST consider using attention. Please choose the type of attention score calculation, and justify it in the documentation. Moreover, you need to decide and justify the position of the attention. If you use a transformer, you MUST add more attention to different positions.

Note: You MUST justify your attention score calculation method (at least 3 methods should be tested), the position you used.

### 3. CRF Attachment

We use CRF to make use of neighbour tag information in predicting current tags. CRF is to  
focus on sentence-level instead of individual positions. Your Seq2Seq model with Attention (in section 1. Stacked Seq2Seq model and 2. Attention) provides the features for the CRF. [Please refer to Lecture 9]  
NOTE: You MUST test your Slot Filling/Tagging model with CRF/ without CRF. Hence, you MUST include the code to the ipynb.

\*\*Note: You can add other additional components to your model if you feel it is necessary. \*\*

### Documentation for this section

You MUST describe how to design the model (Stack Layers, Attention, and CRF Attachment, optional - additional component section), and MUST include justification (previous literature reference or empirical testing/evaluation). (If you don't have any additional component, you do not need to add the additional component section)

Relevant sections in the documentation: Section 3 (Slot Filling/Tagging model)

### Code for this section

You MUST include all the relevant code and running log for this section to the ipynb.

### 1. Performance Comparison

You MUST present F1 - F1 mean score -same as Kaggle Leaderboard on the given validation dataset. Your model should be compared with the baseline Lab09 - BiLSTM CRF code. Your performance table should visualise the result of columns T-F1, T-F1(T), T-F1(S), T-F1(C), T-F1(D), T-F1(P), and T-F1(O). You MUST follow Table 5 in the paper, [CONDA: a CONtextual Dual-Annotated dataset for in-game toxicity understanding and detection](https://arxiv.org/pdf/2106.06213.pdf)

### 2. Ablation Study - different input embedding model

You MUST present the F1 mean score on the given validation dataset with different input embedding that you used (as described in the [Data and Input page](https://www.kaggle.com/competitions/2022-comp5046-a2/overview/data-and-input)). You MUST follow Table 5 in the paper, [Detect All Abuse! Toward Universal Abusive Language Detection Models](https://arxiv.org/pdf/2010.03776v2.pdf) Wang et al., COLING 2020.

### 3. Ablation Study - different attention strategy

You MUST present F1 on the given validation dataset with different attention score calculation AND position that you used (as described in the [Slot Filling/Tagging model](https://www.kaggle.com/competitions/2022-comp5046-a2/overview/slot-fillingtagging-model)).  
Table

Description automatically generated

### 4. Ablation Study - different Stacked layer or # of encoder/decoder strategy

You MUST present F1 on the given validation dataset with different stacked-layer numbers or usage that you used as described in the [Slot Filling/Tagging model](https://www.kaggle.com/competitions/2022-comp5046-a2/overview/slot-fillingtagging-model). The sample table can be the same as the above 3. Ablation Study - different attention strategy.

### 5. Ablation Study - with/without CRF

You are to present F1 on the given validation dataset with CRF usage that you used as described in the [Slot Filling/Tagging model](https://www.kaggle.com/competitions/2022-comp5046-a2/overview/slot-fillingtagging-model). The sample table can be the same as the above 3. Ablation Study - different attention strategy.

### Documentation for this section

* Evaluation setup: You MUST describe the setup environment and hyper-parameter you used, including input embedding dimension, learning rate, optimiser. YOU MUST USE EPOCH 2 ONLY.
* Evaluation result section: You need to visualise the testing results with different combinations of model specifications, e.g. different features that you used in section Dataset and Slot Filling/Tagging model, via the table or graph, and justification. The justification MUST be based on the previous literature reference (incl. international conferences or journal publications) or empirical evaluation. You MUST explain the trend of performance, and the reason (or your opinion) why the trends show as they do.

Relevant sections in the report: Section 4 Evaluation Setup and Evaluation Result

### Code for this section

The relevant code and running log for this section should be in the ipynb.

### The marking details can be found in the [documentation](https://www.kaggle.com/competitions/2022-comp5046-a2/overview/documentation) page

## **Kaggle Leaderboard**

### You should submit the leaderboard with your group number

The top 20 groups in the private leaderboard will receive 3 bonus marks.  
The top 21 to 50 groups in the private leaderboard will receive 1 bonus mark.

The evaluation metric for this competition is \*\*Mean F1-Score\*\*. The F1 metric weights recall and precision equally, and a good retrieval algorithm will maximize both precision and recall simultaneously. Thus, moderately good performance on both will be favored over extremely good performance on one and poor performance on the other.

The **Mean F1-Score** in kaggle is same to [sklearn.metrics.f1\_score](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html)with average = 'micro'.

## **Leaderboard Submission Format**

**For every token in the dataset**, submission files should contain two columns: Id and Predicted.

The file should contain a header and have the following format:

ID,Predicted  
0,O  
1,O  
2,O  
3,O  
4,T  
etc.

Documentation [20 marks]

Please download the Assignment 2 report template (either [Latex template](https://www.overleaf.com/read/khfjzdjxcyzc) or [Word template](https://usydug.files.wordpress.com/2022/05/a2_template_2022.docx))

You should submit pdf version of the assignment 2 report (10 pages maximum - excluding reference and appendix). The detailed requirement for each section can be found ([Data and Input](https://www.kaggle.com/competitions/2022-comp5046-a2/overview/data-and-input), [Slot Filling/Tagging model](https://www.kaggle.com/competitions/2022-comp5046-a2/overview/slot-fillingtagging-model), [Testing & Leaderboard](https://www.kaggle.com/competitions/2022-comp5046-a2/overview/testing-leaderboard))

NOTE:

* The justification MUST be based on the previous literature reference (incl. international conference or journal publication or lecture notes) or empirical evaluation.
* We do not mark the section (or all assignment 2) if you do not include all the relevant code but described in the report.

Sections and Marking Details in the A2 Report

1. Data preprocessing (details can be found from the [Data and Input](https://www.kaggle.com/competitions/2022-comp5046-a2/overview/data-and-input) page)
2. Input Processing (details can be found from the [Data and Input](https://www.kaggle.com/competitions/2022-comp5046-a2/overview/data-and-input) page)
3. Slot Filling/Tagging model (details can be found from the [Slot Filling/Tagging model](https://www.kaggle.com/competitions/2022-comp5046-a2/overview/slot-fillingtagging-model) page)
4. Evaluation (details can be found from the [Testing & Leaderboard](https://www.kaggle.com/competitions/2022-comp5046-a2/overview/testing-leaderboard) page)

| Part | Points |
| --- | --- |
| Data Pre-processing | Optional |
| Input Embedding | 4 marks |
| Slot Filling/Tagging Model - Stacking Layers | 2 marks |
| Slot Filling/Tagging - Attention | 3 marks |
| Slot Filling/Tagging - CRF | 1 marks |
| Evaluation Setup | 2 marks |
| Evaluation Results | 8 marks |
| Total Points | 20 marks |
| Leaderboard - Bonus | 3 marks |

Canvas Submission

Submission Due: 11:59PM, Sunday 05 June 2022

**Submission:**[**Canvas Assignment 2 Submission Page**](https://canvas.sydney.edu.au/courses/39694/assignments/379218)

For assignment 2, you must submit three files:

1. pdf file: a report (documentation) with the given template  
   (file name: your*groupid*A2.pdf)
2. ipynb file or python packages (.py files): An ipynb file or python package that includes all your implementation (the implementation is described in the following sections - dataset, model, and testing-leaderboard) and all the running log.  
   (file name: your*groupid*A2.ipynb)
3. zip file: A zip file that contains trained models, dataset, readme - if necessary, and all other required files that you used for your program.  
   (file name: your*groupid*A2.zip)

NOTE: if you do not submit with your group\_id, your assignment will not be marked. Only one of your team members can submit the assignment file.

# Important Notes

## **Assignment 2 Rules**

### You must follow the rules below. If not, your assignment will not be marked.

1) You can use the Slot Filling/Tagging model with different types of seq2seq model, such as Bi-RNN, Bi-LSTM, Bi-GRU, Transformer, even BERT, etc. However, you MUST NOT directly use the existing state-of-the-art (SOTA) seq2seq architecture as it is but need to modify the seq2seq architecture and propose your own model. For example, you MUST NOT directly use the SOTA seq2seq model (i.e. BERT) without any modification.

2) You can use any pretrained model from any sources but MUST NOT fine-tune it

3) The model described in the report MUST NOT be different from the submitted code and running log in the submitted ipynb file.

4) You can use packages or code from the Labs. If you use/refer any other package or code, please put the reference at the bottom of the code. Otherwise, it will be considered as plagiarism.

5) You MUST NOT submit the prediction result (to the kaggle leaderboard) that is not from your code

6) Your code MUST NOT use any rule(or condition)-based techniques in the model.

7) You MUST NOT use any testing data when you train your Slot Filling/Tagging model. You are allowed to use only the training and validation sets that are provided

8) You MUST use the given template.

9) You MUST use only given training, validate, test dataset for training and testing.

10) You MUST use only Epoch 2.

## **Useful Notes**

1) Please refer to lecture notes, lab materials, and resources that are covered in week 9, 10, and 11.

2) Please proceed your own way if we do not specify it in the assignment specification.

3) Please come to our live QA session (Wed, 9-10pm) if you have any questions. We are more than happy to go through the baseline code or relevant material, and answer your assignment Qs.