**Report and implementation**

For Part 1 the report and the implementation have to be submitted. The grade is based on both the content of the report and the functionality and code quality of the implementation.

**Report**

The report for Part 1 has the structure of an academic technical report, which describes the design of the dialog system and the results of the machine learning experiments, and briefly discusses the approach and the limitations. The report should be a maximum of 12 pages. You need to use the document template provided in Appendix B in the folder for Part 2. Please use single column for Part 1, for LaTeX this means the first line should be \documentclass[nonacm, review]{acmart}. The following list contains required sections and the topics that need to be discussed in each section.

* Title
* Abstract
* Introduction
* Data
* Machine learning
* Dialog manager
* Reasoning
* Configurability
* Conclusion
* Overview of contributions of individual group members

*Title*

A title with team information (team number & members).

*Abstract*

Provide a short summary of the goal of the project, the implementation approach, the machine learning results, the functionality of the dialog system, and possible improvements (250 words maximum).

*Introduction*

A brief introduction to the system design and implementation process, with details on the approach. Discuss the scope of the project, the type of the system and the type of user interaction and the components of the system. Refer to the following sections (using the section numbers) when discussing aspects of the project and system.

*Data*

* Description of the dataset: domain, number of dialogs, dialog format, discussion of the language and grammar used in the data and whether this affects system performance or not, and other properties you consider relevant
* The dialog state transition diagram
* Some example dialog snippets to illustrate some of the states, choice points etc. with explicit references (i.e., "this example sentence is related to dialog state 5 in the model because of X")

*Machine learning*

* Describe how you developed your baseline systems (including any preprocessing and parameter settings)
* Design/layout of the machine learning model
* Data preprocessing, including statistics on the data (label distribution, utterance length, number of out of vocabulary words, etc.)
* Quantitative evaluation for various models and parameter settings
* Error analysis (i.e., for which utterances is the classification wrong, are there specific problematic dialog acts, specific vocabulary, influence of utterance length, could any improvements be made in preprocessing, etc.)
* Discussion of difficult instances with examples

*Dialog manager*

* A description of the state transition function, including example dialogs and references to the state diagram
* Description of system utterance templates, including examples of generated sentences

*Reasoning*

* A description of the reasoning component, including example dialog snippets
* A discussion on how to handle possible contradictions, with examples

*Configurability*

* A list of configuration options
* A high-level description of how the options are set by the user and how they are implemented in code

*Conclusion*

* Discussion of the approach for the dialog system in general:
  + General overview of the project, including a summary of the approach and the main findings
  + Limitations
  + Possible improvements
  + Alternative approaches
* Discussion of the approach for dialog act classification:
  + General overview of the experimental methodology
  + Limitations
  + Possible improvements
  + Choice of machine learning algorithm and alternative approaches

*Team member contributions*

Overview of the contributions of individual group members to the project as a table containing tasks, the person that performed the task, and the amount of time spent on this task. Include a brief reflection on the planning of the project and the individual contributions. Add the total amount of hours per person. Some tasks are likely to be performed by everybody, such as reading dialogs at the start of the project to understand the data. However, try to limit the number of entries assigned to everyone to a minimum, and instead split the task to reflect what each person has actually done. Example table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Alice | Bob | Camille | Dave |
| Keyword baseline | 2hr |  |  |  |
| Error Analysis |  | 1hr |  | 0.5hr |
| Report Data section |  |  | 1hr |  |
| [...] |  |  |  |  |
| Total | 60hr | 55hr | 51.5hr | 61hr |

**Implementation**

For the implementation grade **both functionality and code quality** are taken into account. Functionality means that the code performs the required tasks. Code quality is graded on the following aspects:

* Organization: the code is organized and uses the right amount of modularity, the order of the functions is logical. The number of source code files is limited and the files have sensible names.
* Readability: function and variable names are informative, one statement per line, whitespace between functions.
* Efficiency: loops are used sensibly, the right data structures are used for the task, file i/o is used efficiently.
* Comments: comment each function at the start and, if needed, non-trivial steps within functions, don't overuse comments.
* Consistency: use the same scheme for variable names, capitalization, return statements etc. throughout the code.
* Complexity: make sure the code is not unnecessarily complex, avoid for example functions used as parameters/arguments for other functions.
* Validity: make sure that the code does not contain syntax errors and that the program does not crash during execution (also check that unexpected input is handled without crashes).
* Portability: remove any references to local file path names from the code.
* Redundancy: remove functions and other parts of the code that are not actually used during execution.

Furthermore **report guidelines** are taken into account in the grade. The following guidelines are in effect:

* The template (Word or LaTeX) must be used correctly (single column, numbered sections, maximum 12 pages)
* Python-level descriptions of code ("we used a dictionary for X", "the CSV is loaded with Pandas") are **not allowed**
* Images of text (screenshots of data, screenshots of code, screenshots of tables from the assignment) are **not allowed**
* Each figure or table must be referenced (with the figure/table number) in the text at least once. Section references must always be numbered ("as discussed in Section 2, [...]")
* The report should use an academic writing style: sufficiently formal language use, don't try to promote/sell the program that you have written, sensible use of sections and paragraphs, no excessive italics/bold/colors.

Code quality and report guidelines will account for 10% of the final grade for Part 1.

**Deliverables**

The implementation has to be handed in on Blackboard as an archive file (zip, tar.gz, 7z, tar.xz) containing the report as a pdf file and all code necessary (including any data files) to run the final system. If needed, provide a README file with instructions on how to run the system - although it is of course preferred that running the system is self-explanatory, i.e., do not use too many files or a complex folder structure and choose self-explaining filenames in case you use multiple files. The following components should be submitted for Part 1 (deadline: October 8th, 2023):

* (Part 1a) Python code that implements a majority class baseline and a keyword matching baseline
* (Part 1a) Python code that implements two or more machine learning classifiers, with and without data deduplication
* (Part 1b) A working dialog system interface implementing a state transition function
* (Part 1b) As components of the dialog system:  
  + (Part 1b) An algorithm identifying user preference statements
  + (Part 1b) A lookup function that retrieves suitable restaurant suggestions from the CSV database
* (Part 1c) As components of the dialog system:
  + (Part 1c) Implementation of implication rules
  + (Part 1c) An implementation of dialog model configurability
* The report as a pdf file following the requirements as outlined above

A more detailed explanation of the requirements for each deliverable can be found in the description of the corresponding parts.