Group Project Briefing

How to present (8-min + Live Demo) and how to write the final report (format-compliant, reproducible, well-evaluated)

What you should get from this briefing

- Clarify the project goals and required NLP components
- Know the deliverables, deadlines, and submission rules
- Understand the assessment rubrics and what "good" looks like
- Get a ready-to-use structure for your **project PPT** and **final report**

Note: This deck focuses on **what to include** and **how to organize** your work, not on enforcing one specific model choice.

Agenda (30 minutes)

1. Motivation & task definition
2. Core NLP subtasks & required development steps
3. Deliverables, important dates, submission rules
4. Rubrics: how to score well
5. Recommended structure for the 8-min PPT + Live Demo
6. Recommended structure for the final report
7. Common pitfalls + Q&A

Motivation: why this project

Course-aligned learning outcomes

- **Text data work**: collection, cleaning, structuring, annotation
- **Language generation**: context-aware generation and controllability
- **Interactive systems**: intent understanding, state tracking, real-time response
- **Professional deliverables**: polished demo + clear technical writing

Why an interactive text-adventure setting

- Naturally combines **NLU** (understanding) + **NLG** (generation) + **memory/state**
- Easy to evaluate: branching difference, consistency violations, response time
- Closer to real-world "generative system engineering" than standalone text generation

The focus is not "writing a cool story", but building a **robust NLP-driven interactive narrative system**.

Task definition (StoryWeaver)

AI-powered text adventure with dynamic plot generation

- **Input**: player choices / player text input (e.g., “negotiate with the elder”, “explore the cave”)
- **System**: interpret input, maintain narrative state, generate coherent next plot segments
- **Output**: next plot (narration/dialogue) + next-step options (or guidance for free input)

Key properties

- Dynamic branching
- Contextual coherence
- Multi-turn interaction

Main challenges

- Diverse inputs
- Controllable branching
- Consistency over time

In-class expectations

- 8-min presentation
- Live demonstration
- Explain methods + effectiveness

Core NLP subtasks you should cover

NLU: intent recognition

- Map player input to **action/intent** types (e.g., NEGOTIATE, EXPLORE, ASK)
- Optional: slot/entity extraction (character, location, item, target)
- Low-confidence handling: clarification, fallback, or safer options

NLG: context-aware generation

- Generation must reflect prior events, characters, locations, and constraints
- Recommend structured output: Narration Dialogue Next choices
- Branching must be observable: different choices → different consequences

Plot consistency maintenance

- Track and preserve facts (character status, items, timeline, revealed information)
- Core capability: **detect contradictions** + **repair or refuse** + **explain**

Required development steps (align your report with these)

1) Data preparation

- Collect & organize: adventure scripts, branching narratives, dialogue data, consistency annotations
- Preprocess: cleaning, segment plot units, label narrative logic for consistency
- Goal: training/testing that is reproducible and evaluable

2) Algorithm design

- Context-aware generation, intent recognition, consistency detection, dialogue management
- Efficient real-time pipeline: latency, caching, context window control
- Branching strategy: choices should meaningfully alter plot

3) System implementation

- Suggested stack: Transformers, PyTorch, Gradio
- UI: input → generation → next choices
- Logging: store trajectories for evaluation & writing

4) Performance evaluation

- Narrative quality: coherence/consistency
- Responsiveness: response time
- Choice matching accuracy + user satisfaction

Deliverables overview

Deliverable 1: In-class presentation + Live demo

- Show: task setting, challenges, methods, system functionality, effectiveness
- Mandatory: live demonstration of interactive capability
- PPT must be submitted before

Deliverable 2: Final project report (PDF)

- Must document: task/background, development process, methods, outcomes, evaluation
- Must include: member roles & contributions (names + student IDs)
- Submit to Blackboard (Assessments/Group Project Report)

Practical tip: your demo logs, evaluation tables, and case studies can be directly reused in the report.

Important dates

Item	Date / time	What you should prepare
In-class Presentation + Live Demo	**Apr 8, 2026** (18:30–21:20)	8-min slides + stable demo (prepare a backup demo script)
PPT submission	Before the demo	Upload to Blackboard: Assessments/Group Project PPT
Final project report submission	**Apr 26, 2026** 23:59 (Sunday)	PDF: format-compliant + complete evaluation + contributions

- The PPT must be submitted to Blackboard under *Assessments > Group Project PPT* by **23:59 on April 7**.
- The presentation order for each group will be posted on the Announcement page before the presentation session.

Submission rules (report + PPT)

Final report submission (PDF)

- Blackboard: Assessments/Group Project Report
- File naming: use your Group ID, e.g., `Group10.pdf`
- File size: keep it small (**< 40 MB**)
- Only one member needs to upload; multiple submissions allowed (last one counts)

PPT submission

- Blackboard: Assessments/Group Project PPT
- Submit before the in-class demo (TA will download beforehand)
- After uploading, verify that the submission is saved successfully

> Common pitfall: selecting a file is not the same as a successful submission—always confirm.

Report formatting requirements

Item	Requirement	TA tip
Main text length	Up to **8** A4 portrait pages, single column; no cover page	Use figures/tables to improve readability and density
Font & layout	Times New Roman 12pt, 2.5 cm margins, single spacing	Strict compliance improves perceived professionalism
References	APA format; unlimited space	Cite only what you actually use and discuss
Figures & tables	Up to 2 pages	Every figure/table needs a caption and must be discussed in text
Member contributions	Must include names + student IDs + specific roles	Avoid vague “we all did everything” statements

Assessment rubrics (15% total; 5 criteria × 3%)

Criterion	What “top-level” usually looks like	Typical reasons for losing points
Appropriateness	Task setting, challenges, methods, and functionality are well aligned with the spec	Looks like generic chatting/continuation; weak branching; missing consistency design
Soundness	Clear end-to-end process: data → methods → system → evaluation; reproducible settings	No evaluation; unclear setup; no evidence beyond a demo
Excitement	Engaging and innovative demo points (e.g., controlled branching + consistency repair)	Branches look similar; outputs are monotonous; no highlight
Presentation	Professional structure, clear visuals, stable live demo, good pacing	Overcrowded slides; time overruns; demo instability without contingency
Writing	Clear, concise, format-compliant report with well-explained figures/tables and proper references	Non-compliant format; weak explanations; missing contribution statements

In-class project PPT: recommended structure (8–10 slides)

Goal: explain “what / how / how well / it runs live” within 8 minutes

#	Slide	Must include	Suggested time
1	Title	Group ID, members, 1-line contribution statement	0:20
2	Motivation & challenges	Why it matters + 2–3 key challenges	0:45
3	Task definition (I/O)	Inputs/outputs + system boundary (turns, free input scope)	0:45
4	Data	Sources, scale, preprocessing, schema, train/val/test split	0:45
5	Method overview	Architecture: NLU → State → NLG → Consistency → Output	1:00
6–8	Key modules	Intent / generation control / consistency (1 slide each)	2:15
9	Live demo	Scripted 2–3 turn interaction + show state updates	2:30
10	Evaluation & takeaways	Metrics + 1 failure case + next steps + summary	1:00

Final report: recommended overall structure (within main-text limit)

Suggested section outline (can be used as headings)

1. **Introduction**: problem, task I/O, contributions (3 bullets)
2. **Related Work / Background** (optional): only what you use
3. **Task & Data**: sources, scale, schema, preprocessing, split
4. **Methodology**: NLU, State, NLG, Consistency (clearly defined I/O per module)
5. **System Implementation**: tech stack, inference flow, UI, engineering details
6. **Experiments & Evaluation**: setup, metrics, comparisons/ablations, case studies
7. **Discussion & Limitations**: failure modes and future improvements
8. **Conclusion**: summarize key findings
9. **Contributions**: member roles (names + student IDs + specific work)

How to write the Methodology section (academic + reproducible)

Use a consistent template for each module

- **Input / Output**: define exactly what goes in and what comes out
- **Approach**: model/rules/prompting design (baseline + improvements)
- **Rationale**: why this design addresses your challenges
- **Failure modes**: what inputs still break it

Concrete “evidence” you can include

- State schema + update rules (with one example)
- Consistency checklist (5–10 rules or checks)
- Prompt template / constraint format (appendix or figure)
- Logs: per-turn intent, state, latency, repair flags

```
Example wording:
We formulate player input understanding as intent classification over K action types.
Given the predicted intent and the narrative state, we update a structured memory and
generate the next plot under explicit consistency constraints.
```

What to include in Experiments & Evaluation (most important section)

Minimum set of information (without it, Soundness is hard to justify)

- **Experimental setup**: data split, number of turns, environment/hardware, model config
- **Metric definitions**: how you quantify consistency/coherence, latency protocol, intent accuracy
- **Results table**: at least one (include units and sample sizes)
- **Case study**: one success + one failure (failure + analysis is very academic)

Comparisons / ablations (optional but high value)

- Without consistency vs with consistency
- Prompt design variants
- Decoding strategies (temperature/top-p/beam + rerank)

Presentation tips

- Keep tables narrow and readable
- Use one architecture figure + one results table + one case figure
- Use consistent case format: Input / State / Output / Issue / Fix

Support & Q&A (as specified)

Teaching Assistant contact

Ms. WANG Bingbing
bing-bing.wang@connect.polyu.hk

Email subject suggestion: `COMP5423+Group Project+Group id`
Example: COMP5423+Group Project+Group5

If no reply within 48 hours, follow the spec's instruction to forward to the course instructor.

How to ask efficiently

- Include Group ID + your current progress (data/method/demo/eval)
- Provide reproducible details: example inputs, screenshots, logs, errors
- State your question type: spec clarification / rubric alignment / report format / system design

Q&A

Questions welcome: task interpretation, deliverables, scoring criteria, PPT/report structure, demo design, evaluation protocol.

Lack of Group Members:

- Group 8: Liu Zimeng, LIANG Xiao
- Group_13: RUAN Yilin, YE YI, CHANG Hongtao
- Group_15: ZHI Yong, ZHAN Shi, WU Siyuan
- Word2Vec: HE Ziming
- Group NULL: CHAN Chi Yu, ZHANG Hao, HO Ching Hei, CHAU Wai Hung

There are two Group 17, and can be combined into one.
Group5 and Group 12 don't appear in BB