— Slide 1: Title & Overview —

Good afternoon, I'm Jaewon Lee, and I'll present our NLP term project titled *A Centralized Mixture-of-Experts using Domain-Specialized Small Language Models.*

Our question is simple:
Can multiple **small, specialized models** collectively outperform a **single general-purpose model**?

Large language models are powerful but expensive and inefficient to run.
Small models, though limited, are efficient and surprisingly strong within their domains.
Our goal is to combine these strengths — building a system that's both **efficient and intelligent**.

Instead of one all-knowing model, we propose a **team of smaller experts**, each focused on a specific domain and coordinated by a central controller.
We call this a **Centralized Mixture-of-Experts (Centralized MoE)** — aiming for **efficiency, flexibility, and scalability**.

— Slide 3: Proposed Architecture —

At the center is the **Gateway Small Language Model**, which acts as the controller.
When a user sends a query, the gateway identifies its domain — biomedical, legal, or programming — and routes it to the most relevant expert.
That expert generates a domain-specific response, which the gateway then refines into a coherent final answer.

This setup is:

- **Modular**, since experts can be added or replaced easily,
- **Efficient**, as only the relevant experts are activated, and
- **Scalable**, because new domains can be integrated without retraining everything.

— Slide 4: Domain Setup and Model Design —

Our prototype includes one gateway and three experts: biomedical, legal, and programming.
Each expert is a small or medium-sized model fine-tuned for its domain.
The gateway is optimized for fast and accurate routing.

A key feature is **dynamic loading** — the system activates only the experts needed for a query, saving both memory and computation.

— Slide 5: Gateway Finetuning —

Since the gateway determines routing accuracy, we fine-tuned it on about six thousand labeled queries in three categories:

1. **Domain-specific queries** — clear examples from biomedical, legal, and programming contexts.

2. **Boundary queries** — intentionally ambiguous inputs mixing domains, to improve robustness.

3. **General queries** — everyday inputs, allowing fallback responses when no expert fits.

— Slide 6: Evaluation Strategy —

We evaluate each expert on standard domain benchmarks:

- **Biomedical:** MedQA and PubMedQA

- **Legal:** LexGLUE and CaseLaw

- **Programming:** HumanEval and GSM8K

We also test the gateway on synthetic boundary queries to measure routing accuracy.
All datasets are standardized for fair comparison across domains.

— Slide 6.5: Evaluation Datasets —

Before running evaluations, we prepared our datasets carefully to ensure consistency and fairness across domains.

Step 1: Query Ambiguity Adjustment.
We paraphrased benchmark questions and injected *boundary queries* — inputs that blur domain boundaries — to test how well the gateway distinguishes overlapping topics.

Step 2: Input and Output Standardization.
We unified all datasets into a single format: *(Query, Reference Answer, Domain Label).*
For models that originally output labels, like Legal-BERT, we mapped those labels into short natural-language sentences so every expert produces text-based answers.

Step 3: Text Normalization.
Finally, we applied standard preprocessing — lowercasing, removing special characters, and tokenizer-specific filtering — to keep input quality consistent across all experts.

This preprocessing step ensures that our evaluations measure the models' reasoning, not formatting or dataset inconsistencies.

— Slide 7: Future Plan —

Our next steps:

1. **Finalize soft routing**, allowing partial or weighted routing for mixed-domain queries.
   We aim for **>85% routing accuracy** on validation.

2. **Run full evaluations** across all domains, targeting **≥70% average accuracy**.

3. **Compare with baselines** — a single SLM and a random router — expecting better performance with about **60% less computation**.

4. **Optimize latency** through quantization and efficient dynamic loading, aiming for **under 1.5 seconds per query**.

— Slide 8: Risks and Mitigation —

We identified three main risks:

1. **Routing overfitting:** We'll add a semantic fallback using dense embeddings when confidence is low.

2. **Expert underperformance:** We'll apply extra fine-tuning with LoRA adapters.

3. **Latency issues:** We'll mitigate through quantization and optimized loading.

— Slide 9: Expected Contributions —

We expect three key outcomes:

1. Demonstrate that multiple domain SLMs can outperform a single general-purpose model of similar size.

2. Provide a **lightweight, modular framework** extensible to new domains.

3. Release an **open-source prototype** to support future research on efficient multi-expert NLP.

— Slide 10: Conclusion —

In short, our project rethinks language modeling —
instead of one massive model trying to know everything, we build a **team of small, focused experts** coordinated by a **smart gateway**.

We believe this makes NLP systems more **efficient, adaptable, and sustainable**, bridging specialized knowledge with general reasoning.

Thank you for listening — we'll be happy to take your questions.
