A Strategic and Experimental Framework for Hostable Ensemble Models in Computational User Understanding

Part I: The Core Questions for System Design – A Strategic Framework

The development of artificial intelligence capable of understanding human users from free-text input presents a dual challenge. The first is a problem of scientific depth: "understanding" is not a single task but a complex, multi-dimensional construct. The second is a problem of engineering practicality: the resulting model must be "hostable," implying strict adherence to computational budgets for memory, latency, and cost.

This report provides a strategic framework for addressing this dual challenge. It is organized into two parts. Part I deconstructs the problem by formulating the "core questions" that must be answered to design such a system. It translates the abstract goals of "user understanding" and "hostable" into a concrete set of measurable tasks, quantifiable performance budgets, and candidate architectures. Part II then leverages this framework to propose a rigorous, multi-stage AI experimental plan designed to validate these architectures and identify the optimal, deployable model.

Section 1. Deconstructing "User Understanding": A Framework for Multi-Faceted Analysis

The first and most critical question is: *What, precisely, do we mean by "user understanding"?* This term is not a single, solvable Natural Language Processing (NLP) task. It is a portfolio of distinct, albeit related, inference problems.

1.1 The Central Problem: "User Understanding" as a Latent Vector

Attempting to solve "understanding" as a single, monolithic task is a common failure mode, leading to poorly defined objectives and untestable models. A more robust approach is to de-risk the R&D process by disaggregating the concept. We define "user understanding" as a *latent vector* composed of multiple dimensions, each corresponding to an established domain within computational social science.

This vector encompasses:

1. **Stable Traits:** The user's underlying, long-term personality.
2. **Transient States:** The user's immediate, in-the-moment emotions.
3. **Immediate Goals:** The user's specific, functional intent.
4. **Demographic Markers:** The user's background social and demographic profile.

By disaggregating the problem in this way, we create a measurable, modular, and achievable R&D roadmap.

1.2 Dimension 1: Stable Traits (Personality Recognition)

- **Core Question:** What is the most robust and computationally viable model for human personality?

The scientific literature overwhelmingly converges on the **Big Five model**, also known as the Five-Factor Model (FFM) or OCEAN.[1] This model describes personality along five bipolar scales: Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism.[1] This framework is not merely theoretical; these five traits have been shown to be predictive of numerous real-world outcomes, including academic and occupational success, interpersonal relationships, and health outcomes.[4] This predictive power makes it an exceptionally high-utility target for an "understanding" system.

The standard datasets for this task, which will form the basis of our experiments, include the "Essays" corpus (2,479 stream-of-consciousness essays from students) [2] and the "MyPersonality" dataset, which contains a large corpus of Facebook status updates.[6]

However, a critical conflict exists. While the Big Five is the *scientifically preferred* model, the data for it—which requires users to complete validated personality inventories [1]—is scarce, costly, and difficult to collect.[9] An alternative, the Myers-Briggs Type Indicator (MBTI), is also

used.[2] Unlike Big Five labels, MBTI labels are "very common and easy to retrieve from Twitter," which has led to the creation of larger, more accessible datasets.[9]

This presents a core strategic dilemma:

1. Do we adhere to the "gold standard" Big Five model, despite data scarcity?
2. Do we use the "data-rich" MBTI model, despite its lower scientific rigor?
3. Do we attempt a high-risk, high-reward "data fusion" methodology, mapping MBTI data to Big Five traits to synthetically create a larger corpus?[10]

The choice of strategy will heavily influence the required AI architecture. As will be discussed in Section 4, a data-scarce problem (Big Five) strongly suggests the use of a **Boosting**-based ensemble, a technique that has been demonstrated to excel in low-data regimes.[11]

1.3 Dimension 2: Transient States (Emotion Detection)

- **Core Question:** Are we measuring *what* emotion is felt (categorical) or *how strongly* (dimensional/intensity)?

Emotion detection from text is a mature field with established psychological models and NLP benchmarks.[12] The research highlights two primary approaches to modeling emotion:

1. **Categorical Models:** These classify text into discrete emotion "buckets." The most prevalent are **Ekman's six basic emotions** (anger, disgust, fear, joy, sadness, surprise) [13] and **Plutchik's Wheel of Emotions**, which expands this set to eight (adding trust and anticipation) and organizes them by intensity (e.g., "serenity" as low-intensity "joy," "ecstasy" as high-intensity).[13]
2. **Dimensional Models:** These map emotion onto a multi-dimensional space, such as the Valence-Arousal-Dominance (VAD) model, which rates text on scales of positivity (valence), intensity (arousal), and control (dominance).[17]

The primary evaluation ground for this task is the **SemEval** (International Workshop on Semantic Evaluation) shared task series.[18] These tasks provide key benchmarks, such as "Affect in Tweets" (SemEval-2018 Task 1) [17] and, most recently, SemEval-2025 Task 11, "Bridging the Gap in Text-Based Emotion Detection".[20]

The design of this latest SemEval task provides a crucial specification. Track A is defined as **"Multi-label Emotion Detection"**.[22] This means a single text snippet can be labeled with multiple, even contradictory, emotions (e.g., both "joy" and "fear"). This architectural implication rules out a simple single-output classifier (which uses a $softmax$ function). It demands a model capable of multi-label output, such as a final layer with $sigmoid$

activations for each independent emotion class.

1.4 Dimension 3: Immediate Goals (Intent Classification)

- **Core Question:** What is the user *trying to do* with their text?

While personality and emotion describe *who the user is* and *how they feel*, intent classification (or "user intent detection") addresses *what the user wants right now*.[25] This is a foundational component of language understanding (LU) in modern dialogue systems.[27] It involves classifying an utterance into a specific goal, such as $request\_movie(genre=action)$ or ask_for_help.[27]

This dimension acts as the "utility bridge." A system that can infer a user is high in "Neuroticism" [3] but fails to recognize they are "trying to buy a product" is an academic curiosity, not a useful application.

Unlike personality and emotion, intent is often highly domain-specific, with taxonomies developed for "medical domain" [25] or e-commerce. Publicly available datasets for this task include multi-class corpora for news categorization or sentiment analysis [30] and dialogue system benchmarks.[31] The domain-specific nature of this task strongly suggests a **heterogeneous stacking ensemble** [32], an architecture (discussed in Section 4) where one or more base models are "specialists" trained *only* on intent classification for the target domain.

1.5 Dimension 4: Demographic Markers (Author Profiling)

- **Core Question:** What background demographics (e.g., age, gender) can be inferred from the text?

The definitive benchmarks for this task are the **PAN shared tasks at the CLEF** (Conference and Labs of the Evaluation Forum). These competitions have, since 2013, focused on "author profiling".[33] While earlier tasks addressed authorship attribution, many have explicitly targeted "age and gender identification" [33] and even "personality recognition" [33] from social media texts.

The PAN datasets provide a rich, multilingual [35], and "in-the-wild" testbed. The input is not "clean" laboratory essays; it is Twitter feeds.[36] This makes the PAN corpora a perfect, high-difficulty validation set for our final, integrated model, testing its robustness on noisy,

real-world data.

1.6 Synthesis: The "User Understanding Vector"

Based on this analysis, the first core question ("What is user understanding?") is answered. It is not a single score but a *vector of outputs*. Our system must be designed to predict this multi-dimensional vector. The following table provides the concrete engineering specification, translating the ambiguous problem into a defined "Statement of Work" for the experimental phase.

Table 1: The "User Understanding" Vector (Tasks, Models, and Datasets)

Target Construct	Scientific Model	Key Benchmark Datasets	NLP Task Type
Personality	Big Five (OCEAN) [1]	MyPersonality [6], Essays [2]	Multi-Output Regression (5 scores)
Emotion	Ekman/Plutchik [13]	SemEval 2025 Task 11 [20]	Multi-Label Classification
Intent	Domain-Specific Taxonomy [27]	(User-Defined) or Public Dialogue Corpora [30]	Multi-Class Classification
Profiling	Age / Gender [33]	PAN/CLEF Datasets (e.g., PAN 2015) [37]	Multi-Class Classification

---

Section 2. Defining the "Hostable" Constraint: Quantifiable Performance Budgets

The second core question is: *What, precisely, is our "hostable" budget?* Like "user understanding," the term "hostable" is dangerously vague. It could mean "runs on a personal

laptop" [41], "runs on a phone," or "runs efficiently on a single server-grade GPU."

2.1 The Engineering Budget: Moving from "Hostable" to Hard Metrics

To proceed, "hostable" must be defined in terms of a concrete engineering budget. Based on research into efficient LLM deployment [42], we define this constraint using four key efficiency indicators [44]:

1. **Model Size (Storage):** The model size in GB on disk. This dictates storage costs and, more importantly, the time required to load the model into memory (a "cold start" problem).
2. **Peak Memory (Runtime):** The peak memory (VRAM) in GB required for inference. This is often the single most expensive constraint, as it determines the required GPU class (e.g., a 24GB NVIDIA A10G vs. a 40GB/80GB NVIDIA A100).[41]
3. **Latency (Responsiveness):** The time (in ms) from receiving a user's text to returning the "understanding" vector.
4. **Throughput (Capacity):** The number of requests per second (RPS) or tokens per second (TPS) the system can handle, which dictates its ability to serve concurrent users.[45]

2.2 The Critical Latency Insight: Prefill vs. Decode

A critical distinction from recent LLM inference research [44] is the splitting of latency into two distinct phases:

1. **Prefilling Stage (First Token Latency):** The model processes the *input* (the user's free text). This is a compute-intensive operation that scales with the length of the input.
2. **Decoding Stage (Per-Output Token Latency):** The model *generates* its output token-by-token. This is a memory-bound operation that depends on the size of the "Key-Value (KV) cache".[44]

This distinction radically simplifies our problem. For the "user understanding" tasks defined in Table 1 (classification and regression), we are *not* generating long-form text. The output is a small, fixed-size vector of scores or labels. Therefore, the **"Decoding Stage" is negligible.**

Our entire "hostable" budget is consumed by the **"Prefilling Stage."** We do not need to optimize for complex decoding techniques or a large KV cache. We must optimize for *one thing*: the raw speed of processing the input prompt. This makes the task much more

tractable than generative text.

2.3 The Non-Negotiable Reality: Quantization

A "hostable" budget (e.g., < 8-16 GB VRAM) is fundamentally incompatible with modern base models at full precision. The evidence for this is non-negotiable:

- A 7-billion parameter (7B) model in full precision (float32) requires $7 \times 4 = 28$ GB of VRAM just for model weights, which is too large for most commodity GPUs.[47]
- In half-precision (float16 or bfloat16), a 7B model requires $7 \times 2 = 14$ GB of VRAM.[47]
- Specific analysis of the Llama 3 8B model shows it requires approximately 14.96 GiB (just under 16 GB) for its weights alone.[46] This already exceeds the budget of many common "hostable" GPUs.[41]
- In contrast, the smaller Gemma 2B model requires 4.67 GB in FP16.[48]

The conclusion is clear: "hostable" *requires* quantization. Quantization is a model compression technique that converts model weights from high-precision formats (like float16) to low-precision integers (like int8 or int4), drastically reducing memory footprint and often increasing latency.[43]

The core question is not *if* we quantize, but *how far*?

- **8-bit (int8):** The Gemma 2B model requires only 2.33 GB VRAM.[48]
- **4-bit (int4):** The Gemma 2B model requires a mere 1.17 GB VRAM.[48]

This establishes a central trade-off that must be experimentally measured: the "Quantization Tax." For every gigabyte of VRAM we save, what is the percentage-point of accuracy we sacrifice on our nuanced "understanding" tasks?

2.4 Defining the Budget: A "Hostable" Profile

To proceed, we must propose a concrete budget. We will define "hostable" as a system deployable on a single, commodity, last-generation cloud GPU (e.g., an NVIDIA T4 or A10G). This budget serves as the "engineering contract" and the pass/fail criteria for the experimental phase.

Table 2: Proposed "Hostable" Performance Budget

Metric	"Hostable" Target (per-request)	Rationale
Peak VRAM	< 8 GB	Fits on a single, low-cost cloud GPU (e.g., NVIDIA T4) or high-end consumer card. This is our *hardest* constraint.
Model Size (Storage)	< 5 GB	Allows for int8/int4 quantization of a ~7B model. Fast to download and load.[41]
P95 Latency (Prefill)	< 500 ms	The "prefill" or "first token" latency.[44] Must be sub-second for a "real-time" feel in any interactive application.
Throughput (Batch)	> 10 reqs/sec	A baseline capacity for a production service. This will be highly dependent on batching strategy.

Section 3. Analysis of Foundational Model Candidates (Base Learners)

The third core question is: *What are the best "base learners" for our ensemble?* The research has converged on a new generation of "Small Language Models" (SLMs) that demonstrate remarkable performance, often outperforming older, larger models.

3.1 The "Small, Smart" Landscape

The clear front-runners for a "hostable" system, identified in recent benchmarks [50], are:

1. **Llama 3 8B:** A top-performing model in its class, representing the state-of-the-art for general capability.[54]
2. **Mistral 7B:** Renowned for its high performance-per-parameter. It uses efficient attention mechanisms like Grouped-Query Attention (GQA) and Sliding Window Attention (SWA) for faster inference.[55]
3. **Gemma 2B:** The smallest candidate, designed by Google specifically for on-device and CPU use cases, making it an excellent "low-cost" option.[57]

3.2 Baseline Capability: General Reasoning Benchmarks

Before fine-tuning, we must assess the "general intelligence" of these models. General reasoning benchmarks (like MMLU and GSM8K) serve as a proxy for their ability to understand complex, nuanced human language, which is essential for our "understanding" tasks.

Data from the Hugging Face Open LLM Leaderboard [58] and technical reports [54] reveals a clear hierarchy:

- **Llama 3 8B:** MMLU 66.6, GSM8K 45.7 [54]
- **Mistral 7B:** MMLU 62.5, GSM8K 34.5 [54]
- **Gemma 2B:** MMLU ~42.3-46.5 [60]

This hierarchy defines our central R&D trade-off. Llama 3 8B is the clear performance leader.[54] However, this performance comes at a *literal* cost. On a service like Amazon Bedrock, Llama 3 8B is **~64.3% more expensive** than Mistral 7B ($0.0004 vs $0.00015 per 1,000 input tokens).[62]

This poses a core experimental hypothesis: Is the ~4-point MMLU advantage of Llama 3 8B *worth* a more than 2x increase in cost-per-token? The assumption is that this MMLU gain *will* translate to higher accuracy on our personality and emotion tasks, but this *must* be tested.

3.3 Baseline "Hostable" Metrics: VRAM and Cost

When these candidates are measured against our "Hostable" budget (Table 2), the trade-off becomes even starker.

- **Llama 3 8B:** Requires ~16 GB for FP16 weights.[46] This is *outside* our <8GB budget.

- **Mistral 7B:** Requires ~14 GB for FP16 weights.[47] This is also *outside* our budget.
- **Gemma 2B:** Requires 4.67 GB for FP16 weights.[48] This is the *only* model that fits our <8GB budget *even at half-precision*.

This makes Gemma 2B our "low-cost control." It also opens up a unique possibility: an ensemble of *multiple* Gemma models may *still* be "hostable." For example, a 3-model stacking ensemble composed of int8-quantized Gemma 2B models would require approximately $3 \times 2.33 \text{ GB} = 6.99 \text{ GB}$ of VRAM, which fits comfortably within our 8 GB budget.

3.4 Synthesis: Candidate Profile

We have now answered the third core question. We have three strong candidates, each representing a different point on the cost-performance curve. The following table synthesizes data from multiple technical reports [46] to create a "cheat sheet" for model selection. It directly visualizes the project's core trade-off: Llama 3 8B offers the best "General Capability," but Gemma 2B has the best "Hosting Profile."

Table 3: Baseline Performance and Hosting Metrics for Foundational SLMs

Metric	Llama 3 8B	Mistral 7B	Gemma 2B
General Capability			
MMLU Score	**66.6** [54]	62.5 [54]	~42.3-46.5 [60]
GSM8K Score	**45.7** [54]	34.5 [54]	N/A
Hosting Profile (Est.)			
VRAM (FP16)	~16 GB [46]	~14 GB [47]	**4.67 GB** [48]
VRAM (int8)	~8 GB	~7 GB	**2.33 GB** [48]

VRAM (int4)	~4 GB	~3.5 GB	**1.17 GB** [48]
API Cost ($/M-in)	$0.0004 [62]	**$0.00015** [62]	N/A (Hostable)

Section 4. Optimizing for Insight: Ensemble Architecture Strategies

The final core question is: *How do we combine these models?* The user query specifies "ensemble models," which are proven to "outperform a single model".[64] Ensembles enhance robustness, reduce variance, and improve accuracy.[67]

4.1 The Promise and the Peril of Ensembling

This benefit is not free. Ensembles introduce what we will call the "Ensemble Tax"—a significant increase in computational cost. The "BoostingBERT" model, for example, achieves state-of-the-art results but at the cost of "a large number of parameters and long inference time".[11] This "Ensemble Tax" is in *direct conflict* with our "Hostable" budget.

Our core problem is therefore: *How do we gain the accuracy benefits of an ensemble without paying the full latency and memory cost?*

4.2 Strategy 1: Bagging (Bootstrap Aggregating)

- **How it Works:** Train the *same* base model (e.g., Mistral 7B) multiple times on different "bootstrap samples" (random subsets) of the training data. The final prediction is an average or vote from all models.[67] This is a *homogeneous* ensemble.[32]
- **Pros:** Simple, parallelizable, and excellent at reducing model variance (overfitting).
- **Cons:** Extremely high cost. It requires training and, critically, *hosting* N full models. The memory footprint is $N \times (\text{model VRAM})$.

4.3 Strategy 2: Boosting (e.g., BoostingBERT)

- **How it Works:** Train models *sequentially*. Model 1 is trained normally. Model 2 is then trained to focus on the errors (high-weight examples) that Model 1 got wrong. Model 3 focuses on the errors of Model 2, and so on.[67] It is an error-correction chain.[11]
- **The Data Scarcity Solution:** The research on BoostingBERT provides our single most powerful hypothesis. It "significantly outperforms BERT" and is *especially* useful for tasks with "little training data." One experiment showed **"26.77% performance gains"** with only 0.1% of the training data.[11]
- This creates a direct causal link:
  1. In Section 1.2, we established that our gold-standard personality datasets (MyPersonality, Essays) are "scarce".[9]
  2. [11] proves that Boosting excels in "data-scarce" situations.
  3. **Therefore, a Boosting-based ensemble is the *theoretically optimal* choice for the *Personality* dimension of our "understanding" vector.**

4.4 Strategy 3: Stacking (Meta-Learning)

- **How it Works:** Train multiple, *different* "base models" (e.g., a Mistral 7B and a Gemma 2B). Then, train a "meta-learner" (e.g., a simple logistic regression or XGBoost model) that takes the *predictions* of the base models as its *input features* and makes the final decision.[67]
- **The Specialization Solution:** This *heterogeneous* ensemble architecture [32] is perfectly suited for our multi-faceted "User Understanding Vector".
- This creates a second causal link:
  1. Our "understanding" problem has 4 distinct dimensions: Personality (Regression), Emotion (Multi-Label), Intent (Classification), and Profiling (Classification).
  2. These tasks use different datasets.[9]
  3. **Therefore, we can build a *heterogeneous stacking ensemble* of "specialists":**
     - **Base Model 1 (Mistral 7B):** Fine-tuned *only* on Personality.
     - **Base Model 2 (Gemma 2B):** Fine-tuned *only* on Emotion.
     - **Base Model 3 (Gemma 2B):** Fine-tuned *only* on Intent.
     - **Meta-Learner (XGBoost):** Takes the (5 + 6 + N) outputs from the base models as its features to produce the final, holistic "user profile."
  - This "stacking-of-specialists" architecture is far more modular, maintainable, and likely more accurate than a single, monolithic model attempting to learn all tasks at once (multi-task learning).

4.5 The Resolution: Knowledge Distillation

We are now faced with a conflict. The Boosting and Stacking ensembles are theoretically optimal for our *tasks*, but they are *unhostable*. A 3-model Stacking ensemble has ~3x the VRAM footprint and a P95 latency dictated by the slowest model.

The research on BoostingBERT provides the answer: **Knowledge Distillation**.[11]

- **How it Works:** We first build our large, expensive, and unhostable ensemble (the "Teacher"). This could be the "Stacking-of-Specialists" from 4.4. We then train a *single, small, "student" model* (e.g., one Mistral 7B) whose only job is to *mimic the output logits* (the "soft" probabilities) of the Teacher.
- **The Result:** The Student model learns the complex patterns and "dark knowledge" captured by the full ensemble, but it is a *single, "hostable" model* that fits our budget.
- This is our ultimate strategic answer. The "best hostable ensemble model" is *not an ensemble at all* at inference time. It is a *distilled student* that has *learned from* an ensemble during training.

4.6 Synthesis: Architectural Trade-offs

The following table summarizes the trade-offs, justifying the "Distilled Stack" as the final architectural goal.

Table 4: Qualitative Analysis of Ensemble Architectures vs. Hosting Constraints

Architecture	Primary Use Case	"Understanding" Suitability	"Hostable" Conflict	Resolution
Single Model (Fine-Tuned)	Baseline	Low. Struggles with multi-task complexity and data scarcity.	**High** (Meets budget)	N/A (Baseline)
Bagging [32]	Reduce Variance	Medium. Robust, but not specialized.	**Very High** (N*Models Cost)	Knowledge Distillation

Boosting [11]	High Accuracy	**Excellent** for data-scarce tasks (Personality).[9]	**Very High** (N*Models Cost)	Knowledge Distillation
Stacking [67]	Combine Specialists	**Excellent** for multi-faceted vector (Personality + Emotion + Intent).	**Very High** (N*Models Cost)	Knowledge Distillation
Distilled Student [11]	Final Product	**Optimal.** Captures ensemble accuracy (from Teacher).	**High.** Designed to meet the "Hostable" budget.	**This *is* the resolution.**

Part II: Key AI Experiments for Hypothesis Verification

This part of the report transitions from strategic planning to an actionable R&D plan. We now design a comprehensive, 4-stage experimental protocol to test the hypotheses from Part I and build our final, optimized model.

Section 5. Experimental Protocol 1: Baseline Performance and "Quantization Tax"

5.1 Objective

To establish a "cost-vs-performance" baseline for *single* models. This experiment will answer two questions:

1. What is the best-performing *single* model (Llama 3, Mistral, or Gemma) for our

"understanding" tasks?
2. What is the *exact* accuracy penalty (the "Quantization Tax") for quantizing these models to int8 and int4 to make them "hostable"?

5.2 Hypotheses

- **H1.1 (Performance):** The fine-tuned Llama 3 8B (at FP16) will achieve the highest F1-score and lowest Mean-Squared-Error (MSE) on all "User Understanding" tasks, consistent with its superior MMLU/GSM8K scores.[54]
- **H1.2 (Cost):** The fine-tuned Gemma 2B (at int4) will have the lowest latency and VRAM footprint [48], but also the lowest accuracy.
- **H1.3 (Quantization Tax):** int4 quantization will reduce VRAM by ~4x (vs. FP16) but will cause a significant and unacceptable (>5%) drop in F1-score on nuanced tasks like Personality and Emotion. int8 will be the optimal "hostable" trade-off, balancing accuracy and memory.

5.3 Methodology

1. **Tasks:** Select one representative dataset for each "Understanding" dimension from Table 1:
   - *Personality:* MyPersonality (Big Five Regression) [9]
   - *Emotion:* SemEval 2025 Task 11 (Multi-Label Classification) [24]
   - *Profiling:* PAN 2015 (Age/Gender Classification) [37]
2. **Models:** Llama 3 8B, Mistral 7B, Gemma 2B.
3. **Procedure:** Fine-tune each of the 3 models on each of the 3 tasks. Evaluate each resulting model at three precision levels: FP16, int8, and int4. This yields $3 \times 3 \times 3 = 27$ total experimental runs.
4. **Metrics (Accuracy):** F1-Score (for Emotion/Profiling), Mean-Squared-Error (for Personality).
5. **Metrics (Hostable):** Peak VRAM (GB), P95 Prefill Latency (ms).[44]

5.4 Deliverable: The "Pareto Frontier"

This experiment produces a 2D scatter plot: **Accuracy (Y-axis) vs. Peak VRAM (X-axis)**.

Each of our model/quantization combinations will be a point on this graph. This plot will visually demonstrate the "Quantization Tax" and identify the "Pareto frontier" of models that offer the best accuracy for a given VRAM budget. Our "hostable" budget (<8GB) from Table 2 will be a vertical line on this graph. *Any model to the right of this line is disqualified from being the final product.*

Section 6. Experimental Protocol 2: Quantifying the "Ensemble Tax"

6.1 Objective

To precisely measure the latency and memory *cost* of ensemble architectures, *separate from their accuracy*. This experiment tests the "Ensemble Tax" hypothesis [11] and validates the engineering feasibility of our "hostable ensemble" concept.

6.2 Hypotheses

- **H2.1 (Latency):** A 3-model *Stacking* ensemble (running in parallel) will have a P95 Latency ~1.1x that of the *slowest single model* (due to meta-learner overhead). A 3-model *Boosting* ensemble (running sequentially) will have a P95 Latency ~3x that of a single model, making it unviable for real-time use.
- **H2.2 (Memory):** A 3-model ensemble of Mistral 7B (int8) will have a Peak VRAM of $\approx 3 \times 7 \text{ GB} = 21 \text{ GB}$, *exceeding* our 8GB budget by a wide margin.
- **H2.3 (Mitigation):** A 3-model *Gemma 2B* ensemble (at int8, 2.33GB each) will have a Peak VRAM of $\approx 3 \times 2.33 \text{ GB} = 7 \text{ GB}$, *fitting* within our budget and providing a viable (though low-accuracy) "hostable ensemble" baseline.

6.3 Methodology

1. **Models:** Use the most "hostable" models from Protocol 1 (e.g., Gemma 2B int8 and Mistral 7B int8).

2. **Architectures:**
    - *Baseline:* 1x Gemma 2B (int8)
    - *Baseline:* 1x Mistral 7B (int8)
    - *Stacking (Homogeneous):* 3x Gemma 2B (int8) + 1x scikit-learn Logistic Regression (meta-learner) [67]
    - *Stacking (Heterogeneous):* 1x Mistral 7B (int8) + 2x Gemma 2B (int8) + Meta-learner
    - *Boosting (Simulated):* Simulate 3x sequential runs of 1x Mistral 7B (int8)
3. **Procedure:** Load the models into memory and run 1000 "dummy" inference requests (input text only).
4. **Metrics (Hostable):** P95 Latency (ms) and Peak VRAM (GB).

- **Note:** We *do not* care about accuracy for this experiment. This is purely an engineering benchmark to measure the computational overhead of the ensemble structures themselves.

6.4 Deliverable: The "Cost-of-Ensemble" Report

A simple table quantifying the *exact* latency and VRAM "tax" for each ensemble type, proving or disproving our hypotheses. This will inform the *cost* side of our final cost-benefit analysis.

Section 7. Experimental Protocol 3: Ensemble Architecture Validation (Accuracy)

7.1 Objective

Now that we know the *baseline* (P1) and the *cost* (P2), we must find the *benefit*. This experiment tests our core architectural hypotheses from Section 4: *Boosting for Scarcity* and *Stacking for Specialization*.

7.2 Sub-Experiment 3A: Boosting for Data Scarcity

- **Hypothesis (H3.1):** A 3-model *Boosting-Mistral-7B* ensemble (trained on the

data-scarce MyPersonality dataset) will achieve a *statistically significant* lower MSE (higher accuracy) in Big Five prediction than the *single* fine-tuned Mistral 7B baseline (from P1).

- **Rationale:** This directly tests the connection identified in Section 4.3, based on the success of BoostingBERT in low-data regimes.[9]
- **Methodology:**
  1. Train Model 1 (Mistral 7B) on the MyPersonality dataset.
  2. Identify high-error examples from the training set.
  3. Train Model 2 (Mistral 7B) on a re-weighted dataset that gives higher importance to these errors, as per Boosting theory.[67]
  4. Repeat for Model 3.
  5. Combine predictions via a weighted vote.
  6. Compare the final ensemble's MSE against the P1 baseline MSE for the single Mistral 7B.

7.3 Sub-Experiment 3B: Stacking for Specialization

- **Hypothesis (H3.2):** A *heterogeneous stacking ensemble* (Base 1: Mistral-7B for Personality; Base 2: Gemma-2B for Emotion; Meta-learner: XGBoost) will outperform a *single, multi-task Mistral-7B* (a model trained on both datasets simultaneously).
- **Rationale:** This tests if a team of "specialist" models is superior to one "generalist" model for our multi-faceted "understanding" vector.
- **Methodology:**
  1. **Build the Stack (Model A):**
     - Train Base 1 (Mistral-7B) *only* on the MyPersonality dataset.
     - Train Base 2 (Gemma-2B) *only* on the SemEval dataset.
     - Generate predictions from both on a shared validation set.
     - Train a Meta-Learner (XGBoost) on these predictions.
  2. **Build the Generalist (Model B):**
     - Train a *single* Mistral 7B on a *combined* (MyPersonality + SemEval) dataset, formulated as a multi-task learning problem.
  3. **Compare:** Evaluate Model A and Model B on the held-out test sets for both tasks.

7.4 Deliverable: The "Best-in-Class" Teacher Model

This protocol will identify the *most accurate* ensemble architecture (the winner of H3.1 or H3.2), irrespective of its cost. This "winner" becomes our **"Teacher"** model for the final

experiment.

Section 8. Experimental Protocol 4: The "Distillation" Capstone

8.1 Objective

To resolve the project's central conflict: to compress the "Best-in-Class" (but unhostable) **"Teacher"** from P3 into a "Hostable" **"Student"** that meets our budget. This is the final step, applying the knowledge distillation technique used by models like BoostingBERT.[11]

8.2 Hypotheses

- **H4.1:** The "Teacher" (e.g., the Stacking Ensemble from H3.2) will fail the "Hostable" budget on both Latency and VRAM, as measured in P2.
- **H4.2:** The "Student" (a single, int8-quantized Mistral 7B) trained via knowledge distillation on the Teacher's logits will achieve **>95%** of the Teacher's F1-score and MSE.
- **H4.3:** This "Student" model will *pass* all "Hostable" budget criteria, having the VRAM/Latency profile of a *single* int8 model (as measured in P1).

8.3 Methodology

1. **Select Teacher:** The winning, high-accuracy (but slow/large) ensemble from Protocol 3.
2. **Select Student:** A single, "hostable" base model (e.g., Mistral 7B).
3. **Transfer Data:** Use a large, unlabeled corpus of free text (e.g., the "One Billion Word Benchmark" [5]) as the transfer set.
4. **Train:** Feed the transfer data to the "Teacher" and record its *output probabilities* (logits). Then, train the "Student" model not on the "hard" ground-truth labels, but on *matching the Teacher's "soft" logits* (using a Kullback-Leibler divergence loss).
5. **Quantize:** Apply int8 post-training quantization (PTQ) to the final Student model.[49]

8.4 Deliverable: The Final Validation Matrix

This is the final "go/no-go" scorecard for the project. It compares our three main contenders (Baseline, Teacher, Student) against the project goals, providing the definitive answer to the user's query.

Table 5: Final Validation Matrix for "User Understanding" Ensemble

Model	Accuracy (Avg. F1/MSE)	Peak VRAM (GB)	P95 Latency (ms)	Meets "Hostable" Budget?
P1 Baseline (e.g., Single Mistral 7B int8)	0.75 (Baseline)	7.0 GB	450 ms	**Yes**
P3 "Teacher" (e.g., Stacking Ensemble)	**0.85** (+10 pts)	21.0 GB	1200 ms	**No (Fails VRAM/Latency)**
P4 "Student" (e.g., Distilled Mistral 7B int8)	**0.83** (~97% of Teacher)	7.0 GB	450 ms	**Yes**

Section 9. Synthesis and Strategic Recommendations

9.1 Answering the Core Questions

This framework and experimental plan provide the tools to definitively answer the core questions posed in Part I.

- **What is user understanding?**
  - It is a multi-dimensional vector of personality, emotion, intent, and profiling. The

experimental results from P3.2 will likely confirm this is best modeled by a *heterogeneous stacking* architecture of "specialists" rather than a single "generalist" model.

- **What is the best hostable model?**
  - The optimal model is *not* a single model off-the-shelf, nor is it a traditional ensemble at inference time. The experimental results from P4 will demonstrate that the best model is a *distilled student* (e.g., Mistral 7B int8) that has been trained to mimic a large, complex, unhostable "Teacher" ensemble.
- **What is the optimal architecture?**
  - The "Distilled-Stack" (the P4 Student) is the optimal architecture. As shown in the final validation matrix, it is the only architecture that provides the accuracy of an ensemble (P3) within the strict engineering budget of a single model (P1).

9.2 Final R&D Recommendation

The strategic path forward is clear.

1. A **single, fine-tuned model** (the P1 Baseline) should not be the final product. It will meet the "hostable" budget but will lack accuracy on data-scarce tasks (H3.1) and struggle with the complexity of the multi-faceted "understanding" vector (H3.2).
2. A **raw ensemble** (the P3 Teacher) should not be deployed. It will achieve the highest accuracy but will fail the "hostable" budget due to the "Ensemble Tax" (H2.1, H4.1).

The final recommendation is to execute the 4-stage R&D plan outlined in this report. This plan is designed to build, validate, and deploy the "Distilled Student" model. This is the only identified path to creating a system that is simultaneously rich in "user understanding" and compliant with the non-negotiable "hostable" engineering constraints. This architecture represents the optimal synthesis of the competing forces of psychological depth, ensemble accuracy, and deployment efficiency.

Works cited

1. Building a Corpus for Personality-dependent Natural Language Understanding and Generation - ACL Anthology, accessed November 14, 2025, https://aclanthology.org/L18-1183.pdf
2. Matching Theory and Data with Personal-ITY: What a Corpus of Italian YouTube Comments Reveals About Personality - ACL Anthology, accessed November 14, 2025, https://aclanthology.org/2020.peoples-1.2.pdf
3. Workshop on Computational Personality Recognition: Shared Task - ResearchGate, accessed November 14, 2025, https://www.researchgate.net/publication/365060934_Workshop_on_Computatio

nal_Personality_Recognition_Shared_Task
4. Personality Recognition For Deception Detection - CUNY Academic Works, accessed November 14, 2025, https://academicworks.cuny.edu/cgi/viewcontent.cgi?article=3940&context=gc_etds
5. Getting Personal: A Deep Learning Artifact for Text-Based Measurement of Personality | Information Systems Research - PubsOnLine, accessed November 14, 2025, https://pubsonline.informs.org/doi/10.1287/isre.2022.1111
6. Big Five Personality Trait Prediction Based on User Comments - MDPI, accessed November 14, 2025, https://www.mdpi.com/2078-2489/16/5/418
7. Big Five Personality Detection Using Deep Convolutional Neural Networks - ResearchGate, accessed November 14, 2025, https://www.researchgate.net/publication/354545267_Big_Five_Personality_Detection_Using_Deep_Convolutional_Neural_Networks
8. Big Five Personality Traits Prediction Based on User Comments - Preprints.org, accessed November 14, 2025, https://www.preprints.org/manuscript/202504.2499/v1
9. Is Big Five better than MBTI? - Accademia University Press - OpenEdition Books, accessed November 14, 2025, https://books.openedition.org/aaccademia/3147?lang=en
10. Evaluating LLM Alignment on Personality Inference from Real-World Interview Data - arXiv, accessed November 14, 2025, https://arxiv.org/html/2509.13244v1
11. BoostingBERT: Integrating Multi-Class Boosting into BERT for NLP ..., accessed November 14, 2025, https://arxiv.org/pdf/2009.05959
12. Daily Papers - Hugging Face, accessed November 14, 2025, https://huggingface.co/papers?q=emotion%20classification
13. Emotion Analysis in NLP: Trends, Gaps and Roadmap for Future Directions - arXiv, accessed November 14, 2025, https://arxiv.org/html/2403.01222v1
14. Emotion Detection from text: A Survey - ACL Anthology, accessed November 14, 2025, https://aclanthology.org/W14-6905.pdf
15. PyPlutchik: Visualising and comparing emotion-annotated corpora - PMC - PubMed Central, accessed November 14, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC8409663/
16. Plutchik's Wheel of Emotions | Download Scientific Diagram - ResearchGate, accessed November 14, 2025, https://www.researchgate.net/figure/Plutchiks-Wheel-of-Emotions_fig1_349016413
17. A review on sentiment analysis and emotion detection from text - PMC - NIH, accessed November 14, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC8402961/
18. Sentiment Analysis in SemEval: A Review of Sentiment Identification Approaches - arXiv, accessed November 14, 2025, https://arxiv.org/html/2503.10457v1
19. emotion-analysis-project/SemEval2025-Task11: SemEval2024-task 11: Bridging the Gap in Text-Based Emotion Detection - GitHub, accessed November 14, 2025, https://github.com/emotion-analysis-project/SemEval2025-Task11
20. Pixel Phantoms at SemEval-2025 Task 11: Enhancing Multilingual Emotion

Detection with a T5 and mT5-Based Approach - ACL Anthology, accessed November 14, 2025, https://aclanthology.org/2025.semeval-1.86/
21. Cross-Lingual Multi-Label Emotion Detection Using Generative Models - arXiv, accessed November 14, 2025, https://arxiv.org/pdf/2505.13244
22. JNLP at SemEval-2025 Task 11: Cross-Lingual Multi-Label Emotion Detection Using Generative Models - arXiv, accessed November 14, 2025, https://arxiv.org/html/2505.13244v1
23. HausaNLP at SemEval-2025 Task 11: Hausa Text Emotion Detection - arXiv, accessed November 14, 2025, https://arxiv.org/html/2506.16388v2
24. SemEval 2025 Task 11: Bridging the Gap in Text-Based Emotion Detection (Track A), accessed November 14, 2025, https://www.codabench.org/competitions/3863/
25. Intent detection in AI chatbots: a comprehensive review of techniques and the role of external knowledge - ResearchGate, accessed November 14, 2025, https://www.researchgate.net/publication/396570633_Intent_detection_in_AI_chatbots_a_comprehensive_review_of_techniques_and_the_role_of_external_knowledge
26. Intent detection and slot filling for Persian: Cross-lingual training for low-resource languages, accessed November 14, 2025, https://www.cambridge.org/core/product/identifier/S2977042424000177/type/journal_article
27. Deep Learning for Dialogue Systems - ACL Anthology, accessed November 14, 2025, https://aclanthology.org/C18-3006.pdf
28. Multitask learning for multilingual intent detection and slot filling in dialogue systems - SenticNet, accessed November 14, 2025, https://sentic.net/multilingual-intent-detection.pdf
29. A survey of joint intent detection and slot-filling models in natural language understanding - arXiv, accessed November 14, 2025, https://arxiv.org/pdf/2101.08091
30. TOWARDS RELIABLE HYBRID HUMAN-MACHINE CLASSIFIERS Burcu Sayin G¨unel, accessed November 14, 2025, https://iris.unitn.it/retrieve/handle/11572/349843/579821/Burcu_PhD_Thesis_September_2022.pdf
31. Nations of the Americas Chapter of the Association for Computational Linguistics (2018), accessed November 14, 2025, https://aclanthology.org/events/naacl-2018/
32. Experimenting with ensembles of pre-trained language models for classification of custom legal datasets - ACL Anthology, accessed November 14, 2025, https://aclanthology.org/2022.icnlsp-1.8.pdf
33. Author Profiling at PAN: from Age and Gender Identification to Language Variety Identification (invited talk) - ACL Anthology, accessed November 14, 2025, https://aclanthology.org/W17-1205.pdf
34. Overview of PAN 2019: Bots and Gender Profiling, Celebrity Profiling, Cross-domain Authorship Attribution and Style Change Detection? - Eva Zangerle, accessed November 14, 2025, https://evazangerle.at/publication/daelemans-clef-2019/daelemans-clef-2019.pdf

35. PAN at CLEF 2017 - Author Profiling, accessed November 14, 2025, https://pan.webis.de/clef17/pan17-web/author-profiling.html
36. PAN at CLEF 2018 - Author Profiling, accessed November 14, 2025, https://pan.webis.de/clef18/pan18-web/author-profiling.html
37. PAN at CLEF 2015 - Author Profiling - Webis Group, accessed November 14, 2025, https://pan.webis.de/clef15/pan15-web/author-profiling.html
38. Overview of PAN 2018. Author identification, author profiling, and, accessed November 14, 2025, https://scispace.com/pdf/overview-of-pan-2018-author-identification-author-profiling-2v8d8oi0wv.pdf
39. Overview of PAN 2021: Authorship Verification, Profiling Hate Speech Spreaders on Twitter, and Style Change Detection - Databases and Information Systems, accessed November 14, 2025, https://dbis.uibk.ac.at/sites/default/files/2021-09/clef21_pan_overview.pdf
40. Shared Tasks on Authorship Analysis at PAN 2020 - PMC - NIH, accessed November 14, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC7148014/
41. An Overview of Large Language Models for Statisticians - arXiv, accessed November 14, 2025, https://arxiv.org/html/2502.17814v1
42. AIoT-MLSys-Lab/Efficient-LLMs-Survey: [TMLR 2024] Efficient Large Language Models, accessed November 14, 2025, https://github.com/AIoT-MLSys-Lab/Efficient-LLMs-Survey
43. Efficient Inference of Large Language Models through Model Compression, accessed November 14, 2025, https://sciety-labs.elifesciences.org/articles/by?article_doi=10.20944/preprints202508.0192.v1
44. A Survey on Efficient Inference for Large Language Models - arXiv, accessed November 14, 2025, https://arxiv.org/html/2404.14294v1
45. A Survey on Efficient Inference for Large Language Models - arXiv, accessed November 14, 2025, https://arxiv.org/pdf/2404.14294
46. Choose a GPU for LLM serving | Anyscale Docs, accessed November 14, 2025, https://docs.anyscale.com/llm/serving/gpu-guidance
47. LLaMA 7B GPU Memory Requirement - Transformers - Hugging Face Forums, accessed November 14, 2025, https://discuss.huggingface.co/t/llama-7b-gpu-memory-requirement/34323
48. google/gemma-2b · [AUTOMATED] Model Memory Requirements, accessed November 14, 2025, https://huggingface.co/google/gemma-2b/discussions/57
49. Efficient LLMs Training and Inference: An Introduction - IEEE Xplore, accessed November 14, 2025, https://ieeexplore.ieee.org/iel8/6287639/10820123/10756602.pdf
50. arXiv:2405.11966v4 [cs.CL] 26 Jun 2024, accessed November 14, 2025, https://arxiv.org/pdf/2405.11966
51. Benchmarking Benchmark Leakage in Large Language Models - arXiv, accessed November 14, 2025, https://arxiv.org/html/2404.18824v1
52. Are Small Language Models Ready to Compete with Large Language Models for Practical Applications? - arXiv, accessed November 14, 2025,

https://arxiv.org/html/2406.11402v2
53. Evaluating Open Language Models Across Task Types, Application Domains, and Reasoning Types: An In-Depth Experimental Analysis - arXiv, accessed November 14, 2025, https://arxiv.org/html/2406.11402v1
54. Welcome Gemma 2 - Google's new open LLM - Hugging Face, accessed November 14, 2025, https://huggingface.co/blog/gemma2
55. Mistral 7B, accessed November 14, 2025, https://mistral.ai/news/announcing-mistral-7b
56. Benchmarking Open-Source LLMs: LLaMA vs Mistral vs Gemma - DZone, accessed November 14, 2025, https://dzone.com/articles/benchmarking-open-source-llama-mistral-gemma
57. Gemma vs. Llama vs. Mistral: Exploring Smaller AI Models | Towards Data Science, accessed November 14, 2025, https://towardsdatascience.com/gemma-vs-llama-vs-mistral-exploring-smaller-ai-models-672a95f4b9b7/
58. a Hugging Face Space by open-llm-leaderboard, accessed November 14, 2025, https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard
59. Open LLM Leaderboard - Hugging Face, accessed November 14, 2025, https://huggingface.co/open-llm-leaderboard
60. Welcome Gemma - Google's new open LLM - Hugging Face, accessed November 14, 2025, https://huggingface.co/blog/gemma
61. Benchmarks for Gemma 7B seem to be in the ballpark of Mistral 7B +----------+, accessed November 14, 2025, https://news.ycombinator.com/item?id=39453780
62. Llama 3 8B vs Mistral 7B: Small LLM Pricing Considerations | Vantage, accessed November 14, 2025, https://www.vantage.sh/blog/best-small-llm-llama-3-8b-vs-mistral-7b-cost
63. Google publishes open source 2B and 7B model : r/LocalLLaMA - Reddit, accessed November 14, 2025, https://www.reddit.com/r/LocalLLaMA/comments/1awbo84/google_publishes_open_source_2b_and_7b_model/
64. Survey of transformers and towards ensemble learning using transformers for natural language processing - PubMed Central, accessed November 14, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC10838835/
65. arXiv:2312.05589v2 [cs.AI] 8 Aug 2024, accessed November 14, 2025, https://arxiv.org/pdf/2312.05589
66. A Review of Hybrid and Ensemble in Deep Learning for Natural Language Processing, accessed November 14, 2025, https://arxiv.org/html/2312.05589v1
67. Ensemble Large Language Models: A Survey - MDPI, accessed November 14, 2025, https://www.mdpi.com/2078-2489/16/8/688
68. [2508.16641] Enhancing Transformer-Based Foundation Models for Time Series Forecasting via Bagging, Boosting and Statistical Ensembles - arXiv, accessed November 14, 2025, https://arxiv.org/abs/2508.16641