David vs. Goliath: When Base Models Outperform a Large Transformer

Misra Aditya[1], **Kale Aprup**[1], **Sachan Vangmay**[1], **Haridos Sreelakshmi**[1], **Liu Yinze**[1]
[1]National University of Singapore

aditya.misra@u.nus.edu, aprup.kale@u.nus.edu,
sachan.vangmay@u.nus.edu, sreelakshmiharidos@u.nus.edu, e1249423@u.nus.edu

Abstract

Large transformer models such as DeBERTa-v3-Large achieve strong accuracy on extractive QA benchmarks, such as SQuAD v2.0, but remain expensive to deploy at scale. This work demonstrates that comparable performance can be achieved through an ensemble of smaller and mid-sized transformer models.

We systematically fine-tune a diverse set of architectures on SQuAD v1.1, and combine their predictions using ensembling techniques. We demonstrate that a carefully constructed ensemble of base-sized models can match, and in some cases surpass, the performance of a single large model, while also improving robustness on ambiguous questions.

1 Introduction

Extractive Question Answering (QA) requires selecting a contiguous answer span from a given context passage, that best answers a given question. Transformer architectures such as BERT (Devlin et al., 2019) and its successors have driven rapid progress in this task, achieving near human-level performance on datasets like SQuAD (Rajpurkar et al., 2016, 2018). These gains arise from deep stacks, large hidden dimensions, and extensive pre-training, which enable models to capture fine-grained contextual dependencies necessary for accurate span prediction.

However, this accuracy comes with substantial computational cost: large models incur high GPU memory usage, long inference latency, and limited deployability in resource-constrained settings. An important question, therefore, is whether comparable performance can be achieved without relying on a single, very large model.

A complementary direction explored in other NLP tasks is *ensembling*, where multiple smaller models are combined to exploit diverse inductive biases and reduce prediction variance. While ensemble methods have been applied to classification and sequence generation, their use in extractive QA is comparatively limited. Existing work largely focuses on homogeneous ensembles, e.g., ALBERT variants (Li et al., 2021), leaving open the question of whether heterogeneous ensembles of architecturally distinct models (e.g., ALBERT, ELECTRA, RoBERTa, DeBERTa) can provide meaningful improvements for span selection.

1.1 Literature Review

Extractive question answering has progressed rapidly with the SQuAD datasets, which established span extraction with SQuAD 1.1 (Rajpurkar et al., 2016) and answerability prediction in SQuAD 2.0 (Rajpurkar et al., 2018). Transformer-based pre-trained models such as BERT, RoBERTa, SpanBERT, XLNet, ELECTRA, ALBERT, and DeBERTa have successively improved performance through refined pre-training objectives and architectural innovations (Minaee et al., 2025). Although ensemble methods are well-established in QA systems, the majority of research has focused on either single large models or ensembles of similarly sized architectures. The potential of small, diverse models, particularly base-sized Transformers remains underexplored (Upadhyay et al., 2024). Our work contributes to this space by evaluating whether heterogeneous ensembles of smaller sized models can close the performance gap with larger models while maintaining efficiency.

1.2 Our Contributions

Reducing computational cost is a central challenge in deploying extractive QA systems, as large transformer models are often too expensive for real-world use. Yet few studies explore whether heterogeneous ensembles of smaller models combining fundamentally different architectures such as ELECTRA, and DeBERTa can offer a more efficient alternative. Motivated by this gap, we systematically construct and evaluate such ensembles to

exploit complementary strengths and reduce errors, asking whether small, diverse models can approach or even surpass the performance of a single large model.

2 Methodology

This section outlines the overall experimental pipeline, which consists of two stages: (*i*) fine-tuning several transformer-based models individually for the extractive QA task, and (*ii*) combining their predictions through both rule-based and trainable ensemble methods.

2.1 Individual Model Training

We fine-tuned several transformer architectures with complementary pre-training objectives, including DeBERTa-v3, ELECTRA, RoBERTa, SpanBERT, XLNet, and ALBERT-v2. Their masked-LM, replaced-token, span-corruption, permutation, and parameter-sharing objectives yield distinct span-prediction error profiles, providing the architectural diversity that motivates our ensemble.

All models were initialized from publicly available Hugging Face checkpoints and fine-tuned on the SQuAD 1.1 dataset. Fine-tuning follows a standard supervised setup: given a context paragraph and a question, the model predicts start and end logits corresponding to the answer span within the context. The objective is the sum of cross-entropy losses for the start and end indices. Training used the `AdamW` optimizer with a linear learning-rate schedule and warmup, a batch size of 16, a maximum sequence length of 384, and a stride of 128 for sliding-window tokenization.

To ensure comparability, all models were trained under identical hyperparameters for 3 epochs with mixed-precision enabled. The best checkpoint on the validation set was retained for ensembling.

2.2 Ensemble Design

We explored two ensemble strategies: a *voting-based* ensemble, and a lightweight *trainable* ensemble. Both operate on the prediction outputs of individually fine-tuned models.

Voting-based ensemble. Each model produces a top-ranked answer span. We normalize spans (lowercasing, punctuation and article removal) and assign each model a scalar weight. The final answer is obtained by weighted majority voting over normalized spans, returning the original form of the highest-scoring candidate. This method exploits complementary error patterns across architectures without additional training.

Trainable ensemble. To learn combination behaviour beyond fixed voting rules, we train a lightweight **multilayer perceptron (MLP)** on span-level features. For each question, we pool the top-k spans from all models and construct a feature vector that includes per-model span scores, presence indicators, margins against each model's null score, and simple global statistics. A two-layer MLP is trained with a binary cross-entropy objective to classify whether each candidate matches the gold answer. At inference, the MLP scores all candidates and selects the highest-scoring span, allowing the ensemble to learn complementary confidence patterns across models.

2.3 Evaluation and Implementation Details

All models and ensembles were evaluated on the SQuAD 1.1 development set using the official evaluation script, reporting **Exact Match (EM)** and **F1**. Ensemble performance was compared against the best individual model to quantify gains from combining heterogeneous architectures.

Experiments were conducted on a single NVIDIA A100 (40 GB). Fine-tuning each model required approximately 0.5–1 hours depending on size, while the MLP ensemble trained in under 30 minutes. Preprocessing used Hugging Face tokenizers with custom scripts for offset mapping, sliding-window construction, and extraction of span candidates for ensembling.

2.4 Summary

Our methodology integrates architecturally diverse transformer models and combines them using both rule-based and learned ensembling. This enables a systematic study of how architectural diversity and ensemble structure contribute to improved span accuracy, robustness, and calibration in extractive question answering.

3 Experiments and Results

3.1 Experimental Setup

All experiments use the fine-tuned checkpoints from Section 2. Ensembles operate purely at inference time using saved logits and span candidates. We evaluate three practical questions:

- whether architectural diversity among strong transformer backbones yields complementary span predictions,

- whether probability-level aggregation improves robustness beyond hard voting,
- whether a learned MLP can exploit model-specific confidence patterns.

3.2 Individual Model Performance

Table 1 reports results for all individually fine-tuned models. Multiple base-sized models exceed 91 F1, confirming that they are reliable standalone extractors. Their differing pre-training paradigms yield distinct, non-overlapping error modes, motivating ensembling.

Model	Params	F1	EM
Reference			
DeBERTa-v3-large	304M	**95.08**	**89.57**
Base Models			
DeBERTa-v3-base	86M	93.24	87.16
ELECTRA-base	110M	92.38	86.05
RoBERTa-base	125M	92.14	85.54
SpanBERT-base	110M	91.40	84.80
XLNet-base	110M	91.25	84.25
ALBERT-base-v2	12M	90.57	83.46
BERT-base	110M	88.40	80.53
DistilBERT-base	66M	84.69	75.85

Table 1: Individual fine-tuned model performance on SQuAD 1.1.

Observation. Low-capacity models (ALBERT, DistilBERT) produce unstable span boundaries and noisier confidence scores. Ablations confirm that these errors propagate into ensembles and consistently degrade performance.

3.3 Voting-Based Ensembles

We first evaluate non-learned aggregation strategies.

Hard voting. Table 2 shows results for increasingly diverse model pools. We present experiments in two stages: (i) a mid-capacity pool, and (ii) a strong pool that includes DeBERTa-v3-base.

Config	Models	F1	EM
Stage 1: Mid-capacity pool			
3-model	EL., RoB., SB.	93.45	87.72
4-model	+ ALBERT	93.69	88.16
5-model (weak)	+ BERT	93.43	87.85
Stage 2: Strong pool (with DeBERTa-v3-base)			
5-model (strong)	DeBv3, EL., RoB., SB., XL.	**94.10**	**88.81**
6-model	+ ALBERT	94.10	88.81

Table 2: Hard voting ensembles across weak and strong architectural subsets. Abbreviations: EL. = ELECTRA, RoB. = RoBERTa, SB. = SpanBERT, XL. = XLNet, DeBv3 = DeBERTa-v3-base.

Insight. Architectural diversity improves accuracy only when all included models are sufficiently strong. Performance saturates at five strong models; weaker additions (e.g., ALBERT, BERT) introduce noise.

Soft voting (probability aggregation). We compute final answer scores as:

$$s(a) = \sum_{i=1}^{N} w_i \cdot P_i(a)$$

We use the following heuristic weights based on validation-set F1:

$$w_{\text{DeBv3}} = 1.8, \quad w_{\text{ELECTRA}} = 1.6,$$
$$w_{\text{RoBERTa}} = 1.6, \quad w_{\text{SpanBERT}} = 1.4,$$
$$w_{\text{XLNet}} = 1.4.$$

Method	F1	EM
Hard voting (5 models)	94.10	88.81
Soft voting (heuristic weights)	**94.22**	**89.05**

Table 3: Soft voting using heuristic weights.

Insight. Soft voting improves robustness, particularly for examples containing multiple plausible mentions where distributional confidence, not just the top span, matters.

Weight optimisation (Optuna). We further optimise the weights using Optuna (300 trials, TPE sampler):

Method	F1	EM
Soft voting (heuristic)	94.22	89.05
Soft voting + Optuna	**94.41**	**89.51**

Table 4: Optimised ensemble performance using Optuna.

Final optimised weights. Optuna assigns the following weights:

$$w_{\text{DeBv3}} = 2.46, \quad w_{\text{ELECTRA}} = 1.51,$$
$$w_{\text{RoBERTa}} = 1.06, \quad w_{\text{SpanBERT}} = 1.53,$$
$$w_{\text{XLNet}} = 1.22.$$

Insight. Optimised weights place higher emphasis on models whose errors diverge from DeBERTa-v3-base, indicating that *cross-model error complementarity*, not individual F1, drives improvements.

3.4 Trainable MLP Ensemble

Finally, we evaluate the trainable ensemble introduced in Section 2, where a two-layer MLP classifies candidate spans.

Method	F1	EM
MLP (4 models)	91.97	85.45
MLP (6 models)	90.75	83.82

Table 5: Performance of the trainable MLP ensemble.

Insight. Despite access to span-level features, the MLP underperforms both hard and soft voting. The learned layer disrupts well-calibrated logits from the transformer QA heads and overfits to rare span patterns.

3.5 Summary

Across all experiments we find that:

- Architectural diversity improves span robustness when all included models are individually strong.
- Probability aggregation (soft voting) consistently outperforms hard voting.
- Learned stacking via an MLP fails due to calibration mismatch.
- Weight optimisation yields the strongest non-learned ensemble: **94.41 F1**.

The final ensemble closes much of the gap to DeBERTa-v3-large while using only base-sized backbones.

4 Limitations

Limitations

- **Model Quality Sensitivity:** The ensemble benefits only from models above a certain performance threshold; adding weaker or truly lightweight architectures consistently harms accuracy, limiting usable diversity.
- **Dataset Scope:** All experiments were conducted on SQuAD 1.1, a clean extractive QA benchmark. It is unclear whether the same improvements would generalize to tasks requiring long-context reasoning, multi-hop inference, or handling noisy real-world text.

5 Conclusion

Our work investigated whether ensembles of heterogeneous, base-sized transformer models can serve as an efficient and competitive alternative to deploying a single large model for extractive question answering. By fine-tuning a diverse set of architectures and evaluating multiple ensemble strategies, we showed that architectural diversity yields complementary span predictions that can be effectively exploited through ensembling. Our strongest system - a soft-voting ensemble with optimized weights - achieved an F1 score of 94.41 on SQuAD 1.1, matching much of the performance of DeBERTa-v3-Large while relying solely on base-sized models. Our findings highlight that carefully constructed ensembles can strike a compelling balance between accuracy and complexity..

However, our results also reveal important caveats. More complex meta-learning approaches such as MLP stacking did not provide additional gains, and in some cases degraded performance due to calibration mismatches across models. Likewise, including weaker models consistently introduced noise, underscoring that architectural diversity is beneficial only when each model meets a minimal performance threshold. These observations suggest that the design of effective QA ensembles depends not on the number of models, but on principled selection, weighting, and aggregation of complementary predictors.

Overall, we demonstrate that heterogeneous ensembles offer a practical path toward high-quality extractive QA without relying on prohibitively large models. Extending these methods to more challenging QA setups - including long-context tasks, multi-hop reasoning, and noisy or adversarial datasets - presents a promising direction for future work, where complementary error modes across architectures may provide even greater gains.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *NAACL-HLT*.

Shilun Li, Renee Li, and Veronica Peng. 2021. Ensemble ALBERT on squad 2.0. *CoRR*, abs/2110.09665.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2025. Large language models: A survey. *Preprint*, arXiv:2402.06196.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *ACL*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *EMNLP*.

Prashant Upadhyay, Rishabh Agarwal, Sumeet Dhiman, Abhinav Sarkar, and Saumya Chaturvedi. 2024. A comprehensive survey on answer generation methods using nlp. *Natural Language Processing Journal*, 8:100088.