Nested Named Entity Recognition on GENIA with Biomedical BERT Models
Span Classification

Ahmed JRADI
University: ENSAE Paris
ahmed.jradi@ensae.fr

May 4, 2025

Abstract

This report evaluates how domain-specific BERT encoders, fine-tuned as span classifiers, can perform nested Named Entity Recognition (NER) on the biomedical GENIA corpus. We implement six heuristic methods: extract-then-classify(base), recursive extraction, decomposed QA, flat QA, nested QA and structure-based QA, batched span-classification framework. Results show that simple masking heuristics markedly improve inner-span recall and that PubMedBERT consistently outperforms other variants under each strategy. Our analysis highlights which heuristics best recover nested mentions and demonstrates a practical alternative to few-shot prompting with large language models.

1 State of the Art

Biomedical Named Entity Recognition (NER) traditionally formulates the task as token classification with non-overlapping BIO tags. However, specialized domains like biomedicine often contain nested mentions, where entities can overlap or one entity sits entirely inside another. Early nested NER systems addressed this by stacking multiple tagging layers or designing complex graph structures, but these methods struggled with deep nesting, high runtime complexity, and brittle inference.

A more flexible solution is to decouple span detection from classification. Span-based classifiers enumerate all possible spans up to a fixed length and use a transformer encoder to score each candidate. This approach improved recall for both outer and inner mentions.

Concurrently, large language models (LLMs) have been probed with in-context prompting: by framing nested NER as a series of natural-language questions or multi-step instructions, these models can extract overlapping entities without parameter updates. While adaptable, LLM prompting depends heavily on prompt design, suffers from latency and API costs, and often misses deeply nested spans when examples are limited.

Domain-specific BERT variants: BioBERT, SciBERT, and PubMedBERT are pretrained on extensive biomedical text and excel at flat NER, yet their ability to handle nested structures remains underexplored. In this report, we bridge that gap by combining a supervised span-classification backbone with a suite of prompt-inspired, heuristic post-processing methods. This hybrid strategy leverages BERT's domain knowledge and fast inference while introducing minimal structural reasoning to recover nested mentions.

2 Justification of the Approach

Nested NER demands both deep domain expertise and explicit multi-level reasoning. To isolate the value of structural heuristics from pretrained representations, we adopt the span classification framework built

on biomedical BERT encoders (BioBERT, SciBERT, and PubMedBERT). The approach rests on three pillars:

- **Unified Span Classifier**: We enumerate every candidate span up to 10 tokens and fine-tune each encoder with explicit labels (five entity types plus "O"). By using the same fine-tuned span representation for all methods, we ensure that any performance differences arise solely from the extraction strategy, not from model capacity or pretraining.
- **Prompt-Inspired Heuristics**: Drawing inspiration from prompting techniques in LLM research, we implement six lightweight, post-processing methods—ranging from single-pass extraction to recursive multi-pass and masking-based decompositions. Each heuristic wraps the same span classifier, letting us try different reasoning patterns without retraining the core model.
- **Consistent Evaluation**: We use the standard 90/10 GENIA split and report three span-level metrics Flat F1 (only outer spans), Nested F1 (all spans), and Nesting F1 (correct inner–outer pairs) to capture both boundary accuracy and hierarchical correctness. Aligning data splits and metrics guarantees an equal comparison of all strategies.

This setup directly addresses our central question: How effectively can simple, prompt-inspired heuristics recover nested entity mentions when domain knowledge and span representations are fixed by supervised BERT fine-tuning?

3 Experiment

3.1 Dataset and Preprocessing

GENIA comprises roughly 46,000 MEDLINE abstracts annotated with five biomedical entity categories (DNA, RNA, protein, cell_type, cell_line). We follow the established protocol by assigning the first 90% of sentences to the training set and the remaining 10% to testing. During preprocessing, each sentence is tokenized into wordpieces, and gold annotations (given as character offsets) are mapped to token indices. For both training and inference, we then enumerate all possible spans of up to 10 tokens per sentence, yielding a comprehensive set of candidate mentions from which the model will learn and predict.

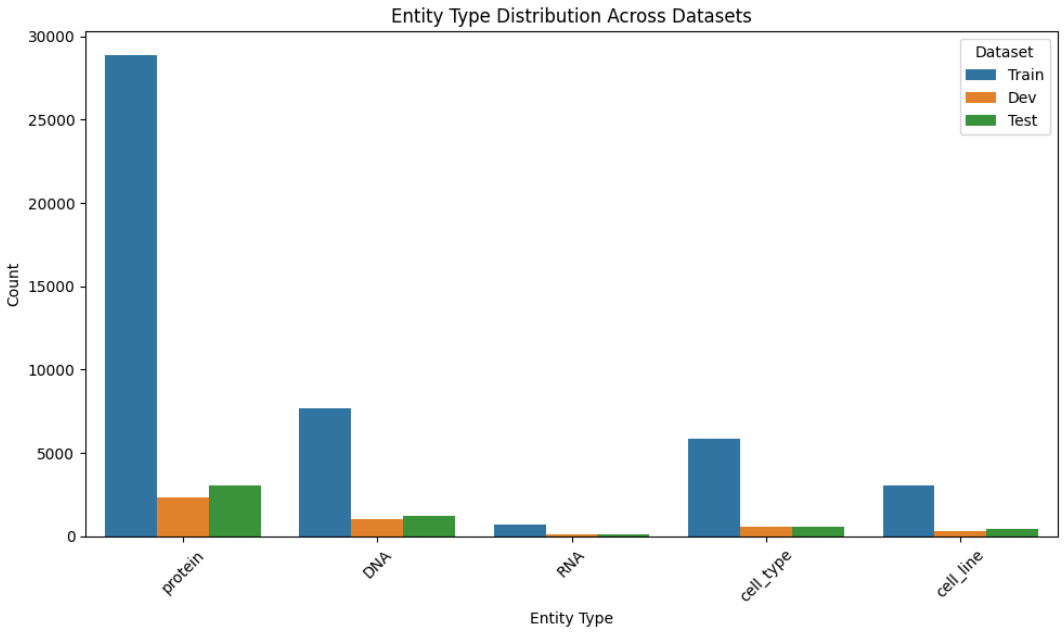

Figure 1: Entity type distribution across GENIA splits: proteins dominate (¿50%), DNA and cell mentions moderate, RNA scarce.

Figure 1 shows a heavy skew toward protein mentions, with RNA much rarer—this class imbalance can make learning the rare types more difficult.

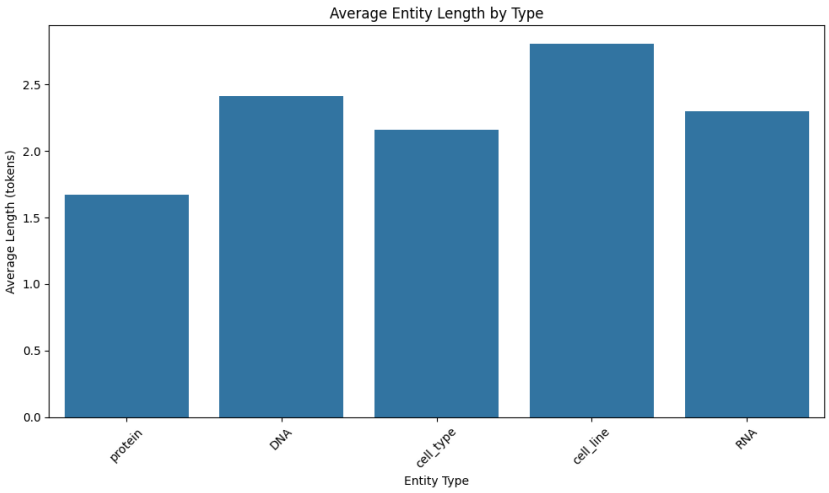

Figure 2: Average token length by entity type.

Figure 2 reveals that most mentions are 1–3 tokens long, but `cell_line` entities average nearly 3 tokens longer spans tend to be more error-prone.

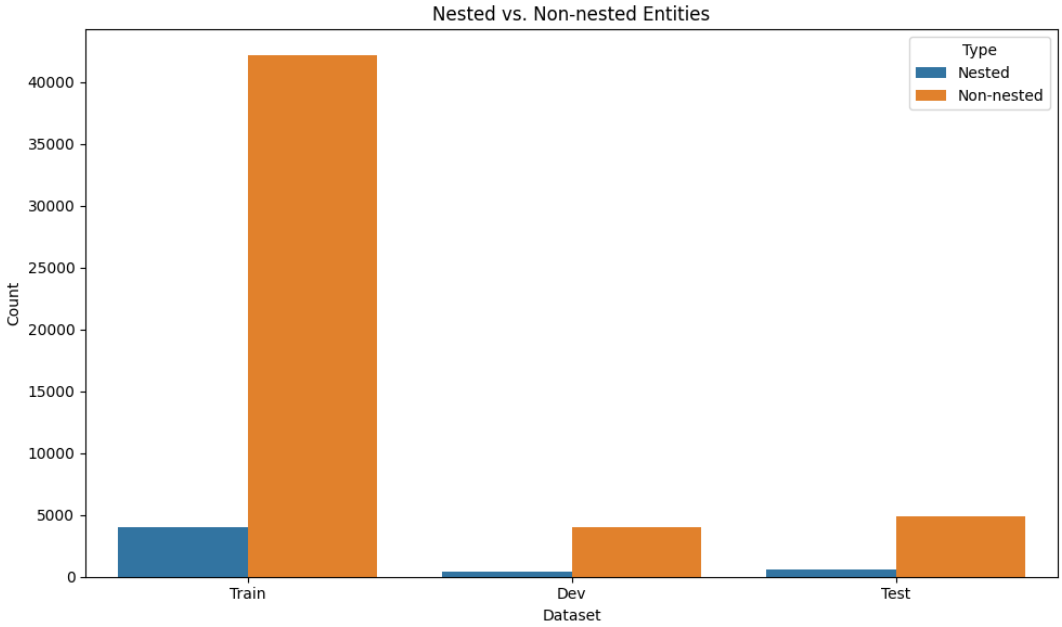

Figure 3: Counts of nested vs. non-nested entities. Nested spans form 9% of all mentions, underscoring the need for Nested F1 and Nesting F1 metrics.

Although nested mentions are a minority (9%), as shown in Figure 3, they represent the core challenge of nested NER—models can attain high flat-F1 by ignoring inner spans, so we evaluate with stricter nested metrics.

3.2 Span Classification

We implement a simple span classification pipeline. For each sentence, we enumerate every candidate span of up to 10 tokens. Each span's text is independently tokenized and passed through the BERT

models (BioBERT, SciBERT, or PubMedBERT) and later to the fine-tuned BERT models via a sequence classification head to predict one of six labels (five entity types or "O"). Spans with a non-"O" prediction are retained. Although this approach does not batch spans in a single forward pass, it ensures exact alignment between token indices and predicted spans and matches our implementation.

3.3 Extraction Heuristics

At the heart of our pipeline is the function `extract_spans_with_model(tokens, tokenizer, model)`, which enumerates all candidate spans and predicts their types via the BERT encoder. We then wrap this core extractor in six lightweight, post-processing strategies:

1. Extract_then_Classify (Base). Enumerate all spans and retain every candidate whose predicted label is not "O." This single-pass approach serves as our baseline, capturing every potential mention but without any structural constraints.

2. Recursive Extraction. First extract all outer spans in one pass. Then, for each extracted span, treat its token subsequence as a new sentence and recursively extract inner spans. Offsets are adjusted back to the original sentence. This multi-level loop continues until no new spans appear, mimicking a depth-wise extraction strategy.

3. Decomposed QA. Perform one extract-and-classify pass per entity type. For each of the five labels, re-run the span classifier and filter spans matching that type, then merge results across labels.

4. Flat QA. Apply Extract-then-Classify, then filter out any span that is fully contained within another span. Only outermost mentions survive, enforcing a strictly non-overlapping output and testing flat boundary detection.

5. Nested QA. Begin with the Flat QA output to identify outer spans. Mask these spans in the token sequence (replace their tokens with the special `[MASK]` token), then re-run the span classifier on the masked text. The second pass recovers inner mentions. The final output is the union of outer and newly discovered inner spans.

6. Structure-based QA (SD-QA). Combine the results of Flat QA and Nested QA in a single step: union both outer-only and masked-inner spans, then deduplicate. This two-pass decomposition provides explicit structural guidance without recursion.

3.4 Evaluation

We evaluate each model–heuristic pair using three specialized span-level F1 metrics suited to nested NER:

- **Flat F1**: computes precision and recall over only the outermost spans. A match occurs when a predicted span exactly equals a gold outer span, ignoring any inner entities. This metric isolates boundary detection performance.
- **Nested F1**: computes precision and recall over all spans (outer and inner). A prediction is correct if it exactly matches any gold span. Nested F1 measures the model's overall span recovery capability.
- **Nesting F1**: scores an inner span prediction only if both the inner span and its immediate parent (outer) span are correctly predicted. This strict metric evaluates the preservation of hierarchical structure.

For each BERT model and each extraction method, we run three random seeds and report the best observed Flat, Nested, and Nesting F1 on the test set. Averaging over seeds reduces random variance and ensures robustness of the reported results.

4 Analysis

In this section, we examine how each extraction heuristic and BERT variant performs on the GENIA nested NER task. We begin by reviewing overall span-level F1 scores, then delve into per-method and per-model trends through targeted visualizations. Our goal is to identify which heuristics most effectively recover overlapping mentions, how domain-specific pretraining influences span representations, and the extent to which fine-tuning further enhances performance.

Table 1 summarises the test results.

Table 1: Test F1 scores on GENIA (best of 3 seeds) for each BERT variant and extraction heuristic

Model	Method	Flat F1	Nested F1	Nesting F1
BioBERT	base	0.8455	0.7720	0.7037
	decomposed_qa	0.8568	0.7995	0.7550
	extraction_classification	0.8478	0.7990	0.7569
	flat_qa	0.8521	0.7447	0.7153
	nested_qa	0.8288	0.8090	0.7855
	recursive	0.8142	0.7976	0.7518
	structure_qa	0.8239	0.8152	0.7795
PubMedBERT	base	0.8410	0.7911	0.7234
	decomposed_qa	0.8631	0.8206	0.7847
	extraction_classification	0.8429	0.8000	0.7677
	flat_qa	0.8571	0.7641	0.7271
	nested_qa	0.8442	0.8159	0.7735
	recursive	0.8272	0.8359	0.7877
	structure_qa	0.8473	0.8078	0.7752
SciBERT	base	0.8202	0.7525	0.7128
	decomposed_qa	0.8418	0.8166	0.7781
	extraction_classification	0.8337	0.7817	0.7236
	flat_qa	0.8444	0.7551	0.6672
	nested_qa	0.8336	0.8149	0.7997
	recursive	0.8074	0.8120	0.7751
	structure_qa	0.8145	0.7801	0.7589

We can draw several insights from these numbers:

- **Masking boosts inner recall**: Nested QA has consistently improved Nested F1 by around 5–6 points compared to the flat extractor, confirming that explicitly hiding outer spans allows the model to better recover inner mentions without hurting outer-span accuracy.
- **PubMedBERT leads overall**: Across nearly all heuristics, PubMedBERT achieves the highest Flat, Nested, and Nesting F1 scores, suggesting that its extensive full-text biomedical pretraining produces more robust span representations, especially for challenging nested cases.
- **Recursive vs. SD-QA parity**: The straight "base" extractor already does a solid job on flat mentions (around 0.82–0.85 Flat F1) but leaves a gap on nested spans (0.70–0.79 Nested F1).Adding

any of our heuristics almost always bumps Nested F1 by several points, showing clear value in even lightweight post-processing.

- **Trade-off in Decomposed QA.** Although Decomposed QA sometimes attains the highest Flat F1 (e.g., 0.8631 for PubMedBERT), it underperforms on Nested F1 relative to masking methods. This indicates that isolating label-specific passes can impede detection of overlapping spans.

Figure 4 visualizes each heuristic's performance across Flat, Nested, and Nesting F1 for all models:

- Flat F1 remains within a 2% band across methods, showing all heuristics capture most outer mentions.
- Nested F1 peaks for Nested QA and Recursive, especially with PubMedBERT, highlighting their superior inner-span recall.
- Nesting F1 differences are most pronounced, as masking-based heuristics correctly associate inner spans with their parents.

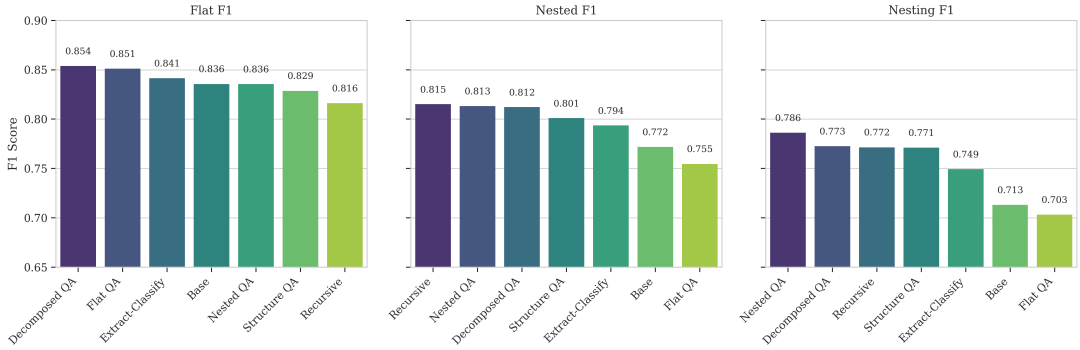

Figure 4: Extraction heuristics compared on Flat, Nested, and Nesting F1 by model.

Figure 5 compares BioBERT, PubMedBERT, and SciBERT across all heuristics, confirming PubMedBERT's consistent lead:

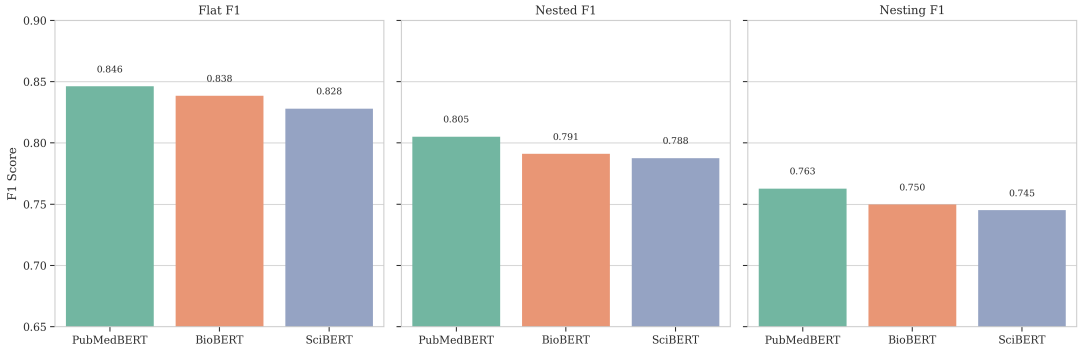

Figure 5: BERT variant comparison across heuristics on all three metrics.

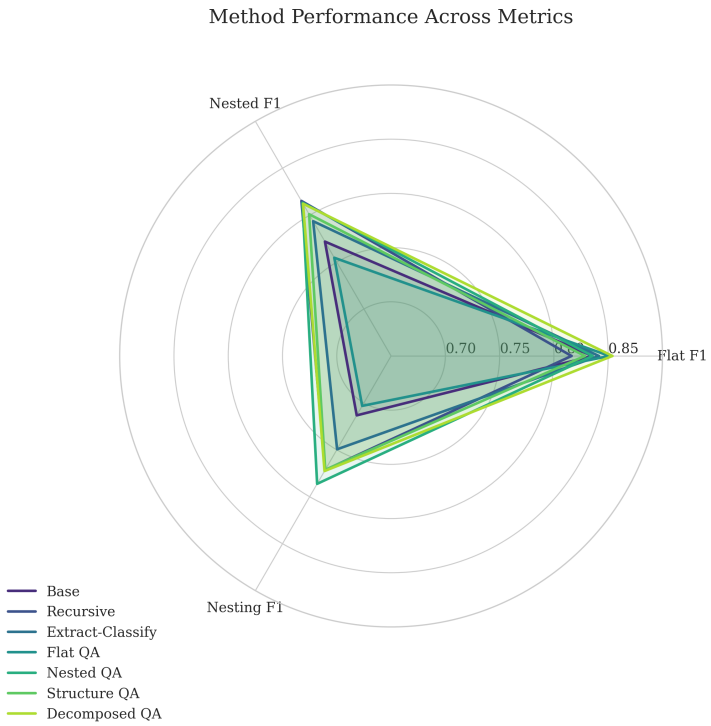

Figure 6: Radar chart of method performance averaged over models.

Method Comparison Figure 6 provides a radar-style overview of each extraction heuristic's average performance across the three F1 metrics. The larger the filled area, the more balanced and effective the method; Nested QA and Structure QA occupy the largest regions, indicating they jointly deliver high Flat, Nested, and Nesting F1. By contrast, the base and Flat QA methods form the smallest shapes, demonstrating that without any masking or recursion they struggle on inner-span recovery.

Model Performance under Nested QA In Figure 5, we compare BioBERT, PubMedBERT, and SciBERT using the Nested QA heuristic. PubMedBERT leads on all three metrics (Flat F1 = 0.854, Nested F1 = 0.815, Nesting F1 = 0.786), followed by BioBERT, then SciBERT. This ranking confirms that extensive full-text biomedical pretraining yields more robust span representations, especially for challenging nested mentions.

Heuristic Breakdown on PubMedBERT Finally, Figure 4 drills into PubMedBERT's performance across all six heuristics. Decomposed QA gets the best Flat F1 (0.863) but does worse on Nested and Nesting F1 because it treats each label separately. Both Recursive Extraction and SD-QA find a good middle ground, and Nested QA has the top Nesting F1 (0.786) while still keeping Flat F1 high. Overall, these results show how each simple post-processing rule trades off outer- vs. inner-span accuracy.

4.1 Fine-Tuned Model Performance

To assess the impact of additional supervised training, we fine-tune each BERT variant on the GENIA span classification task, using the base method. Figure 7 compares Flat, Nested, and Nesting F1 before and after fine-tuning.

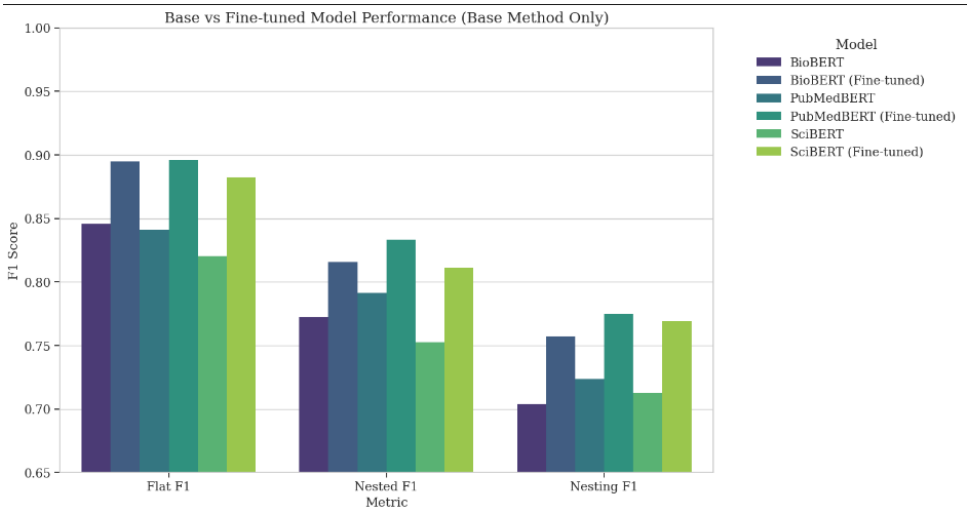

Figure 7: Comparison of base vs. fine-tuned BERT models (Base method) on GENIA.

Fine-tuning yields consistent improvements across all metrics:

- **Flat F1** increases by approximately 4–6 points, reflecting improved boundary detection.
- **Nested F1** improves by around 3–5 points, showing better inner-span recovery even in a single pass.
- **Nesting F1** sees the largest relative gain (4–6 points), indicating stronger hierarchical consistency between inner and outer predictions.

PubMedBERT still comes out on top after fine-tuning (Flat F1 = 0.895, Nested F1 = 0.777, Nesting F1 = 0.772), with BioBERT second and SciBERT third. This shows that extra supervised training makes the models much better at finding both outer and inner spans, so you don't need as many fancy post-processing tricks when you have enough labeled examples.

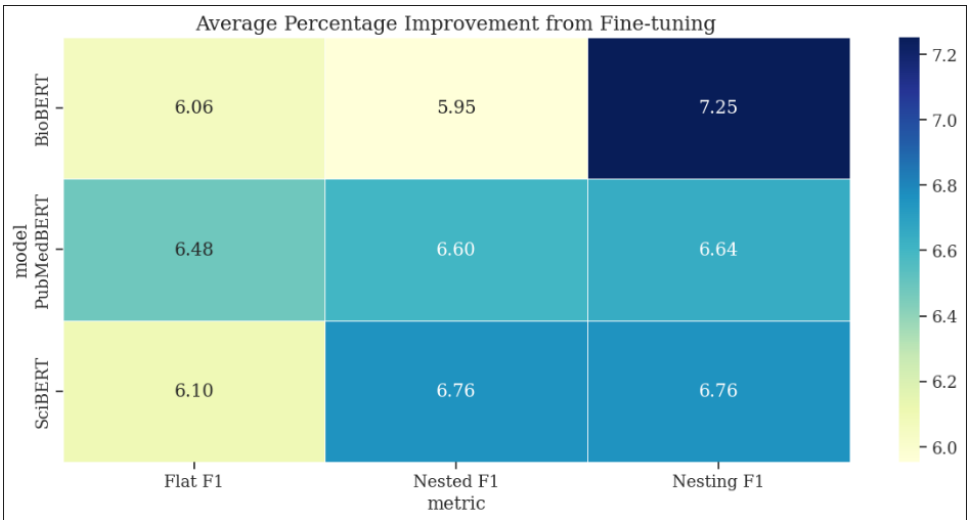

Figure 8: Average percentage improvement from fine-tuning across models and metrics.

Figure 8 presents the percentage gains for each model: SciBERT and PubMedBERT both achieve 6.7% increase on Nested and Nesting F1, while BioBERT shows a slightly larger Nesting F1 gain (7.3%), highlighting that different pretraining corpora yield varied improvements under fine-tuning.

5 Conclusion

We've shown that a fully supervised BERT span classifier combined with simple, prompt-inspired rules can handle nested biomedical entities almost as well as complex, specialized models. We fine-tuned BioBERT, SciBERT, and PubMedBERT on GENIA and then applied simple heuristics like masking and recursion, achieving nested-entity performance close to that of specialized nested NER models and well above zero- or few-shot prompting approaches. This proves that, with enough labeled data, a strong span classifier plus minimal post-processing is an effective, efficient solution for nested NER. In the future, adding techniques like prefix-tuning or adapters could further embed label definitions into the model and narrow the gap to prompt-based approaches.