ToMMeR – Efficient Entity Mention Detection from Large Language Models

Victor Morand[1] **Nadi Tomeh**[2] **Josiane Mothe**[3] **Benjamin Piwowarski**[1]

[1]Institut des Systèmes Intelligents et de Robotique (ISIR),
Sorbonne Université, CNRS, F-75005 Paris, France

[2]LIPN, Université Sorbonne Paris Nord, UMR7030 CNRS

[3]INSPE, UT2J, Univ. Toulouse, IRIT, CLLE
UMR5505 UMR5263 CNRS, F-31400 Toulouse, France

Abstract

Identifying which text spans refer to entities –mention detection– is both foundational for information extraction and a known performance bottleneck. We introduce ToMMeR, a lightweight model (<300K parameters) probing mention detection capabilities from early LLM layers. Across 13 NER benchmarks, ToMMeR achieves 93% recall zero-shot, with an estimated 90% precision under a human-calibrated LLM-judge protocol, showing that ToMMeR rarely produces spurious predictions despite high recall. Cross-model analysis reveals that diverse architectures (14M-15B parameters) converge on similar mention boundaries (DICE >75%), confirming that mention detection emerges naturally from language modeling. When extended with span classification heads, ToMMeR achieves competitive NER performance (80-87% F1 on standard benchmarks). Our work provides evidence that structured entity representations exist in early transformer layers and can be efficiently recovered with minimal parameters.

https://github.com/VictorMorand/llm2ner

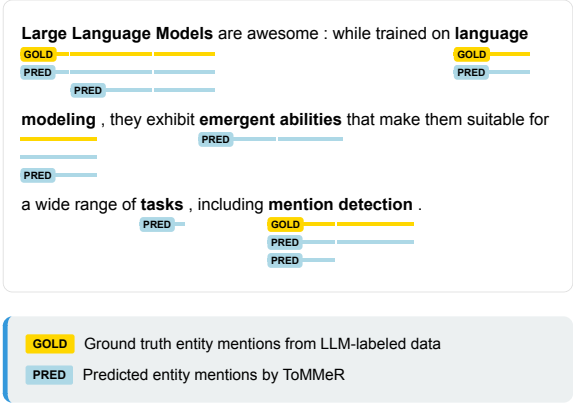

Figure 1: ToMMeR, a lightweight model probing emergent mention detection capabilities from early layers representations of any LLM backbone. Trained to generalize LLM annotated data, ToMMeR achieves high Zero-Shot recall across a wide set of NER benchmarks.

1 Introduction

Information extraction (IE) pipelines start with a fundamental task: *mention detection*—identifying text spans that refer to entities or concepts worth tracking. These mentions range from specific referential entities (*Marie Curie*, *Tesla Inc.*) to abstract concepts (*philosopher*, *oxidation process*), typically realized as noun phrases. Despite its importance, mention detection remains a recognized bottleneck in NER systems (Popovic and Färber, 2024), yet it is almost always conflated with entity typing in a single joint task. This conflation obscures a key question: *Where and how do models learn to detect span boundaries?*

In contrast with entity labels, mention boundaries are fundamentally schema-invariant, decoupling detection from typing could then be key to building IE systems that transfer better. While NER models typically employ hundreds of millions of parameters trained on task-specific annotations (Zaratiana et al., 2023), evidence from mechanistic interpretability suggests that Large Language Models (LLMs) may already encode entity spans during pretraining (Feng and Steinhardt, 2024a; Geva et al., 2023; Morand et al., 2025). If mention detection emerges from language modeling objectives, *it should be recoverable from LLM representations with minimal additional parameters*.

We propose ToMMeR (Token Matching for Mention Recognition), a lightweight architecture (under 300K parameters, trainable in hours) that scores spans from early layers of frozen LLM backbones, using only one (partial) forward pass—no prompting, no schema specification, no text generation (42× *faster than prompting methods*, cf App B). We train ToMMeR using only span *boundaries* from Pile-NER—GPT-3.5 annotations on samples from *The Pile* (Zhou et al., 2023b; Gao et al., 2020a). Because typed NER benchmarks underlabel many legitimate mentions (nested, generic,

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 27489–27509
July 2-7, 2026

etc.), "precision" against gold often penalizes coverage, we thus complement initial benchmark annotation by estimating ToMMeR's precision with human-calibrated LLM judges. Across 13 NER benchmarks, ToMMeR achieves 93% recall with precision estimated over 90%, while also showing strong multi-lingual transfer on Latin scripts (Table 1). Cross-model analysis reveals that diverse LLM architectures (14M to 15B, both autoregressive and encoder-only) converge on similar mention boundaries (DICE scores >0.75), suggesting mention detection is a shared, emergent capability rather than a dataset artifact.

While various systems perform untyped mention detection, most rely on supervision tied to specific datasets or annotation schemes. Coreference models rank spans but inherit conventions from their training data (Lee et al., 2017). Weakly supervised approaches prioritize high-recall proposals when gold annotations are incomplete (Miculicich and Henderson, 2020), and span-based event systems detect untyped triggers before clustering (Lu and Ng, 2021). In all cases, span scorers remain shaped by benchmark-specific schemas and do not transfer cleanly across domains or annotation guidelines. Closer to our work, EMBER (Popovic and Färber, 2024) trains NER models over LLM attention scores and hidden states, but requires labeled data for each schema, thus remaining tailored to specific datasets. Generalist extractors such as GLiNER broaden coverage, supporting zero/low-shot transfer, yet require an input schema at inference time, reintroducing task specification and alignment costs (Zaratiana et al., 2024).

Our contributions are threefold: **(i)** a simple, efficient probing architecture that recovers mention spans from a single forward pass of early layers; **(ii)** empirical evidence that mention boundaries are robustly encoded across layers, models, scales, and architectural families, with consistent cross-model predictions despite no shared supervision; and **(iii)** we release ToMMeR models and demonstrate a straightforward extension to full NER via span classification, achieving competitive performance (80-87% F1) on standard benchmarks and enabling modular, schema-agnostic extraction pipelines.

2 ToMMeR

Entity Mentions in Transformers. In transformer-based language models (Vaswani et al., 2017), a text is tokenized into a sequence of tokens $(t_1, \ldots, t_n) \in \mathcal{V}^n$, with $\mathcal{V}$ the vocabulary used by the tokenizer. These tokens are embedded into a sequence of initial *representations* that are sequentially processed through the transformer layers. Each layer $\ell \in \{0, \ldots, N_L - 1\}$ generates a new series of representations $(\boldsymbol{z}_1^\ell, \ldots, \boldsymbol{z}_n^\ell) \in \mathbb{R}^d$ from the representations of the preceding layer.

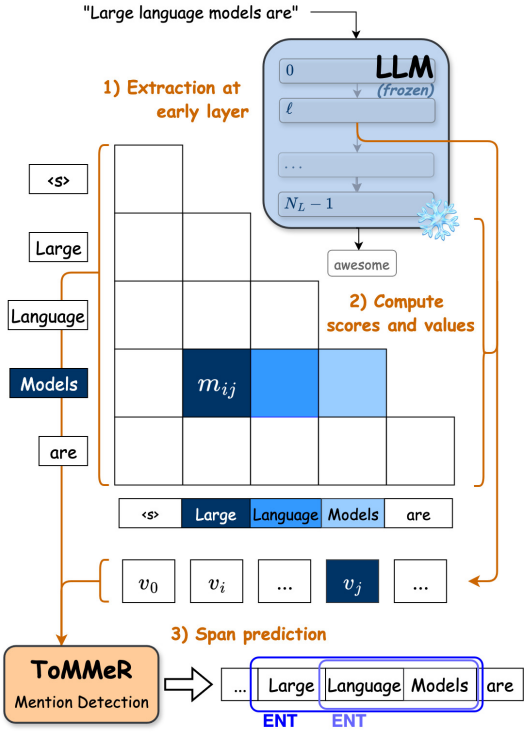

Figure 2: The ToMMeR architecture. We extract the mention detection capabilities of any *frozen* LLM backbone with less than 300K additional parameters, computationally equivalent to an additional attention head. We leverage *Matching scores* m_{ij} between tokens t_i and t_j and individual values v_i, all probed from LLM representations at layer ℓ.

For each sentence $(t_1, \ldots, t_n) \in \mathcal{V}^n$, we consider the set of N_E *entity mentions* $E = \{(s_k, e_k) \in [1, n]^{2 \times N_E}\}$ with s_k and e_k respectively the start and end *token* indexes of entity mention k, constrained in this work to contiguous spans of length ≤ 25 tokens, covering the majority of mentions[1]. This limit reduces the quadratic complexity of span enumeration while preserving most linguistically valid mentions. The task is then framed as binary classification: for each span, determine whether it constitutes a valid mention.

Architecture. We design ToMMeR as a probing classifier—a small neural network trained on frozen representations to recover latent capabilities. Unlike fine-tuning, probing preserves the backbone

[1]More details on the dataset in Section A.2

model and requires minimal parameters. More specifically, we ground our approach on the binding ID framework from mechanistic interpretability (Feng and Steinhardt, 2024b), which posits that transformers dynamically bind related tokens through learned signals, enabling later retrieval via attention. Extending this idea, we hypothesize that LLMs implicitly group entity-mention tokens using analogous binding mechanisms—effectively encoding mention boundaries within their hidden states. ToMMeR leverages these latent binding signals to extract mention spans directly from representations of a frozen LLM, *requiring no modifications to the backbone model*. For each pair $(\boldsymbol{z}_i^\ell,\ \boldsymbol{z}_j^\ell)_{1\le i\le j\le n}$ of token representations at layer ℓ, the *Matching score* $m_{ij} \in \mathbb{R}$ quantifies this association.

Matching Score. To detect entity bindings, we adapt the transformer's attention mechanism, which measures token similarity via dot products between query and key vectors. While standard attention computes a probability distribution over tokens, our goal is to capture binary token-to-token matching. Thus, we replace softmax with ℓ_2 normalization which proved to be stable across backbones, yielding cosine similarity as our matching metric. Formally, we compute pairwise scores using learned projections on a rank r query-key subspace $(W_Q, W_K) \in \mathbb{R}^{r\times d}$.

$$m_{ij} = \cos(W_Q \boldsymbol{z}_i^\ell\ ,\ \underbrace{W_K \boldsymbol{z}_j^\ell}_{\in\mathbb{R}^r}) \in [0,1] \qquad (1)$$

We also explore other normalization functions and formulations for the matching score, probing different transformer components in Appendix E.

Token Values and Span Probability. To complement pairwise matching scores, which capture inter-token bindings but lack boundary and anchor information, we incorporate token-level information with a learned linear layer (or probe) $v_i = W_V \boldsymbol{z}_i^\ell \in \mathbb{R}$. These values leverage the observation that LLMs concentrate entity information in its final token representation (Meng et al., 2022; Geva et al., 2023), providing critical cues for autoregressive models. The final span probability $\hat{p}_{ij} \approx p\left((i,j) \in E\right)$ is predicted with a logistic model, with parameters $\theta \in \mathbb{R}^5$, and inputs the matching scores and individual values around the mention's last token. The model is given by equ. 2.

$$\hat{p}_{ij} = \theta \cdot \begin{pmatrix} m_{ij} \\ \max\{m_{kj}\}_{i<k\le j} \\ \min\{m_{kj}\}_{i<k\le j} \\ v_j \\ v_{j+1} \end{pmatrix} \qquad (2)$$

where max/min pooling over intermediate matching scores m_{ik} captures the strength of internal token bindings within the span, and v_j, v_{j+1} provide information about the span's last token and its immediate context (See architecture Figure 2).

2.1 Training on Span Detection

Data. We use Pile-NER (Zhou et al., 2023c), a dataset of 45,889 samples from The Pile (Gao et al., 2020b), annotated with fine-grained entity types using GPT-3.5. While originally designed for zero-shot NER systems like UniversalNER (Zhou et al., 2023c) and GLiNER (Zaratiana et al., 2023), its broad semantic coverage and diverse mention types make it ideal for studying mention detection. Additional dataset statistics are provided in Appendix A.

Loss. ToMMeR's parameters are optimized end-to-end on binary span classification (valid/invalid). Mention detection however faces severe class imbalance, as non-entity spans (negative examples) vastly outnumber entity mention spans (positive examples), even in Pile-NER. To address this, we employ Balanced Binary Cross-Entropy (BBCE), which reweights the standard BCE loss using a dynamic factor α. This ensures equal contribution from both classes regardless of their imbalance: For each batch, the loss is computed as:

$$\begin{aligned}\texttt{BBCE}(\hat{p}, y) = \frac{-1}{\#_{\text{Tot}}} \sum_{i<j} \alpha\ y_{ij}\ \log(\hat{p}_{ij}) \\ + (1-y_{ij})\ \log(1-\hat{p}_{ij})\end{aligned} \qquad (3)$$

where $\hat{p}_{ij}$ is the predicted probability, y_{ij} is the gold binary label, α is the balanced class weight, computed for each batch as $\alpha = \frac{\#\text{Neg}}{\#\text{Pos}}$, and $\#_{\text{Tot}}$ is the total number of spans in the batch.

Distillation. Although it has been generated with an LLM, and already contains an important number of fine-grained entity types, Pile-NER also suffers from incompleteness. Nested mentions are for instance not labeled. Moreover, even when mentions are extracted by the LLM, the labels may not fully reflect the internal notion of mention detection. To mitigate these limitations, we adopt a two-stage training strategy: after a first fit on the available

annotations, we use the learned model to augment the training dataset with new mentions that were not annotated in the data, thus reducing the number of false negatives. See Appendix C for detailed hyper-parameters.

3 Related Work

Mechanistic and Algorithmic Structure. Mechanistic studies suggest specific circuits for entity-related behavior: *Binding ID* capture abstract entity via activation directions (Feng and Steinhardt, 2024a); induction heads implement copy/coreference-like mechanisms (Olsson et al., 2022); attention heads bracket NPs or track antecedents (Clark et al., 2019); and modular subgraphs compose across subtasks (Mondorf et al., 2025). Complementarily, probing/theoretical work recovers tree or chart-like structure from hidden states (e.g., Inside-Outside/CKY signals) (Zhao et al., 2023; Tenney et al., 2019), and Allen-Zhu and Li (2023) show causal autoregressive models can learn formal grammars, with hidden states linearly encoding boundaries and attention flows mimicking dynamic programming. Recent work also studies entity identification directly in language-model representations, showing that entity information is highly linearly separable and concentrated in low-dimensional subspaces of early layers (Sakata et al., 2025). These findings support the view that the allocation of probability over spans is a natural byproduct of next-token prediction; we connect this to the detection of untyped mentions extracted from early layers.

Embedded/Probing Detectors and Span Boundary Models. Low-latency extraction can be achieved with probes on frozen LLMs. Popovic and Färber (2024) predict token and span-boundary signals from hidden states and attention during generation. While EMBER targets schema-specific NER with supervised training, we extract a schema-agnostic notion of mention that generalizes zero-shot. Classic probing shows entity/span information concentrates in intermediate layers, and structural probes reveal linear syntax/span structure (Tenney et al., 2019; Hewitt and Manning, 2019); probes have also tested entity state tracking (Kim and Schuster, 2023). Orthogonally, pointer/boundary decoders focus on detecting mention as the bottleneck for Information Extraction/Entity Linking (Li et al., 2019; Shang et al., 2018; Bian et al., 2023).

Open-Schema IE and Generalist NER. A broad line of work targets ontology-agnostic IE via generalist or instruction-driven interfaces. GLiNER and GLiNER2 support schema-driven extraction and zero/low-shot transfer (Zaratiana et al., 2024, 2025); UniversalNER distills LLM capabilities into smaller models for open NER (Zhou et al., 2023d). Unified text-to-structure frameworks and instruction-tuned systems (UIE, USM, InstructUIE, RAIT, YAYI-UIE, PIVOINE, RUIE, InstructIE, TRUE-UIE) expand this paradigm with retrieval and prompting strategies (Lu et al., 2022; Lou et al., 2023; Wang et al., 2023; Xie et al., 2024; Xiao et al., 2024; Lu et al., 2023; Liao et al., 2025; Jiao et al., 2023; Wang et al., 2024). Recent LLM-based NER work also improves in-context extraction without parameter updates through label-guided demonstration retrieval and targeted error reflection (Bai et al., 2025). Retrieval-based mention retrieval further enables zero-shot typing (Shachar et al., 2025). We differ by *probing* early layers of frozen LLMs to recover a model-internal notion of entity mention with minimal additional parameters, rather than training a new generalist encoder or relying on in-context demonstrations at inference time.

Distillation, Pseudo-Labels, and Evaluation. Distillation from LLM to small models aids broad-coverage information extraction (Zhou et al., 2023c); self-training and confidence-based pseudo-labeling mitigate annotation gaps (Sohn et al., 2020), alongside prototype/contrastive approaches and pseudo-label refinement (Zhou et al., 2023a; Zhang et al., 2023). Relatedly, weakly supervised few-shot domain adaptation methods leverage small labeled support sets together with unlabeled target-domain data, for instance through joint constrained k-means and discriminative subspace selection for NER (Hammal et al., 2025). Our approach complements these methods by revealing that LLMs already encode a rich, generalizable notion of entity mentions—effectively distilling and amplifying this latent knowledge with minimal parameter overhead.

4 Mention detection Task Evaluation

To evaluate ToMMeR's ability to detect entity mentions without fine-tuning, we assess its performance across three dimensions: (1) zero-shot transfer to common english NER benchmarks, (2) precision validation using LLM-based judgment, (3) Multi-lingual transfer. Our findings demonstrate

		Threshold decoding			Greedy (flat) decoding					
	Dataset	R	P	F1	R	P	F1	#samples	#entities	Nested
Gold Benchmarks (en)	MultiNERD	98.6	21.7	35.5	94.0	30.0	45.5	154 144	23,8005	✗
	CoNLL 2003	94.8	33.6	49.6	86.4	44.7	58.9	16 493	3,4761	✗
	CrossNER politics	97.0	32.4	48.6	84.2	54.5	66.5	1 389	8,838	✗
	CrossNER AI	97.0	35.0	51.5	87.2	51.2	64.5	879	3,776	✗
	CrossNER literature	94.4	40.3	56.5	85.9	56.6	68.3	916	4,749	✗
	CrossNER science	95.7	38.2	54.6	85.9	55.1	67.1	1 193	6,318	✗
	CrossNER music	95.5	44.1	60.3	86.7	61.7	72.1	945	6,420	✗
	ncbi	91.9	12.7	22.2	66.0	17.1	27.2	3 952	6,808	✗
	FabNER	73.6	30.1	42.8	49.0	39.7	43.9	13 681	64,761	✗
	WikiNeural	97.8	20.7	34.1	90.8	28.2	43.1	92 672	149,005	✗
	Ontonotes	73.0	25.5	37.8	59.0	31.4	40.9	42 193	103,956	✗
	ACE 2005	42.0	28.7	34.1	32.5	31.9	32.2	8 230	30,778	✓
	GENIA NER	95.7	24.8	39.4	72.0	34.6	46.6	16 563	55,968	✓
	Aggregated	92.6	23.2	37.1	84.0	31.2	45.5	353 250	714,143	
	Average	88.2	29.8	42.6	75.3	41.3	52.1	353 250	714,143	
WikiANN	WikiANN - en	84.2	33.4	47.8	75.5	46.2	57.3	40 000	56 035	✗
	WikiANN - es	85.3	34.9	49.5	74.1	53.0	61.8	40 000	49 280	✗
	WikiANN - fr	86.8	38.5	53.4	75.8	56.1	64.5	40 000	52 972	✗
	WikiANN - de	91.3	33.8	49.3	83.6	48.6	61.5	40 000	55 329	✗
	WikiANN - zh	40.7	6.3	10.9	24.0	8.8	12.9	40 000	50 033	✗
LLM	MultiNERD-gpt-4.1-mini	71.1	92.8	80.5	50.7	95.9	66.3	1064	8,915	✓
	GENIA-gpt-4.1-mini	68.4	92.3	78.6	37.5	94.9	53.7	512	8,704	✓

Table 1: Zero-shot mention detection performance of ToMMeR (plugged at layer 6 of LLAMA3.2-1B, 274K parameters only), on various NER benchmarks. Precision (P), Recall (R) and F1-scores for threshold and greedy (flat) decoding. Precision and F1 in gray stress that low precision is expected when evaluating a schema-agnostic mention detection model on typed data. Top sub-table shows results on standard english benchmarks, followed by aggregated and mean values. The middle sub-table shows generalization on multi-lingual data, showing transfer to other Latin languages, while bottom rows shows results on LLM-annotated (LLM) datasets. ToMMeR yields high recall on most common NER datasets in a zero-shot setup; while real precision is controlled with LLM judged data.

that ToMMeR achieves high recall (92.6%) with minimal parameters (274K), while LLM-judged precision (92%) confirms alignment with a broad notion of entities.

4.1 Zero-shot Mention Detection

Datasets and Metrics. We evaluate ToMMeR on 13 NER corpora spanning news, Wikipedia, scientific/biomedical and industrial domains, and multi-genre resources, as well as multi-lingual data. We detail all datasets in Appendix H, and provide a comparison of schemas and annotation methods in Appendix I. We report recall, precision, and F1 scores on mention detection. We target high recall to capture all potential entity mentions, while lower precision is expected as ToMMeR detects mentions beyond standard benchmark types (Precision and Recall in gray in Table 1).

Results. In this zero-shot setup, ToMMeR consistently achieves high recall across most of the 13 tested benchmarks (See Table 1 upper part and average), demonstrating strong coverage of entity mentions and general alignment with the notion of entity captured in these datasets (up to 98.6% recall on MultiNERD). Precision is as expected lower on gold data due to non-nested and limited scope of annotated entities. One exception is on ACE 2005 (40% recall vs. 88.2% in average), due to a distinct notion of entity mention learned on Pile-NER. ACE include both determiners (e.g., "the president"), which differs from the patterns ToMMeR learned (e.g. "president") and pronouns (for coreference resolution, not annotated in Pile-NER), which ToMMeR didn't learn to detect[2]. ToMMeR also transfers well on Latin languages, achieving 86% average recall on French, Spanish, and German when evaluated on WikiANN. Performance however drops on Chinese (41% recall), likely due to significant differences in tokenization and syntax encoding in the English-centric Llama backbone.

Flat (Non-Nested nor Overlapping) Decoding By design, ToMMeR can predict any *continuous* span, which naturally produces many nested entities under our criteria. Since most applications require flat annotations, we can choose to post-

[2]A comparison of those schemas is provided in appendix, fig. 15, as well as random ACE samples with ToMMeR's predictions in App A.3. More details in App. H and I.

process the predictions to obtain a non-overlapping segmentation of the text. Additionally, by adjusting the decision threshold, ToMMeR offers a flexible trade-off between precision and recall, depending on the application's needs. To evaluate flat (non-nested nor overlapping) segmentation, we implement a *greedy decoding algorithm* that iteratively select the highest scored span that does not overlap with previously chosen ones. Constraining the model to produce non-overlapping mentions on flat NER datasets further improves precision, increasing the average from roughly 30% to 40%. (See greedy decoding columns in Table 1).

4.2 Evaluating precision – LLM as a Judge

To better evaluate ToMMeR precision, we must determine whether predicted spans that are not labeled in the dataset are false positives. We use an *LLM-as-a-judge* evaluation to create two LLM-judged datasets, and cautiously validate them with human judgments to address circularity risks and control biases.

LLM Annotation. We consider MultiNERD (commonsense) and GENIA (domain-specific) benchmarks. We first generate as many candidate mentions as possible using the ToMMeR model with highest recall on validation data, ensuring we capture the broadest possible set of entity spans. Then, given a span predicted by ToMMeR, we prompt `gpt-4.1-mini` (OpenAI, 2025) to assess whether it qualifies the predicted span as an entity according to the Wikipedia definition (Wikipedia contributors, 2025). The complete prompt is provided in the appendix Figure 13. We sample 10,000 spans, providing sufficient coverage to produce reliable precision estimates while limiting inference costs. After removing spans deemed invalid by the LLM judge, we obtain curated datasets containing a high number of nested entity mentions. This approach allows us to capture the precision of our model beyond the limitations of existing benchmark annotations, including nested and otherwise unannotated mentions and providing a more comprehensive and realistic measure of performance. On these curated datasets, ToMMeR reaches a precision of 92% (two LLM rows Table 1), demonstrating that while the model captures nearly all mentions, it rarely produces spurious predictions.

Human validation of LLM judgments. We first verify that the entities accepted by the LLM cover the gold-annotated ones: This is the case in more than 99% of the cases. We then quantify the positive bias (Thakur et al., 2025) of our judge, which potentially overestimate precision. We collected human annotations on predicted spans sampled from LLM-judged data across GENIA (600 spans) and MultiNERD (1,200 spans). Annotations were performed by 5 CS researchers following identical instructions as the LLM judge, with strict blinding to both LLM predictions and other annotators' judgments (full protocol in Appendix D). For MultiNERD, human-LLM agreement reaches 91.5% ($\kappa = 0.449$), indicating substantial consensus on common-sense entities. For GENIA, agreement drops to 78.5% ($\kappa = 0.239$), reflecting genuine ambiguity in technical biomedical entities rather than simple judge failure—both non-expert human annotators and the LLM struggle with domain-specific boundaries.

Data	MultiNERD	GENIA	Aggregated
Judged Precision	92.8	92.3	**92.55**
Human Agreement	91.5 %	78.5 %	**87.17 %**
Support	1200	600	**1800**
Cohen's κ	0.449	0.239	**0.359**
Bias Correction	-2.00%	-5.17%	**-3.06%**
Rectified Precision	90.80	87.13	**89.49**
95% Lower Bound	89.42	84.03	**88.11**

Table 2: Estimating ToMMeR real precision with the PPI framework: using an LLM-as-a-judge, controlled with human annotation. While `gpt-4.1-mini` does exhibit positive bias, human validation confirms that it can be used as a reliable evaluator. More details in App. D.

To obtain unbiased precision estimates, we apply the Prediction-Powered Inference (PPI) framework (Angelopoulos et al., 2023). It uses a small human-labeled sample to adjust predictions from a larger LLM-labeled set: providing a scalable, cost-effective method for precision estimation. This yields a corrected precision of $90.8 \pm 2\%$ for MultiNERD and $87.1 \pm 5\%$ for GENIA with 95% confidence (instead of raw LLM-judged precisions of 92.8 and 90.3 respectively), substantially exceeding the low precision using the gold-annotation from standard benchmarks (23.2%). This confirms that most ToMMeR predictions rejected by benchmarks are genuine entities outside the annotation scope rather than false positives. Details, including Cohen's κ scores, are reported in table 2. More details are reported in Appendix D.

This approach ultimately captures ToMMeR's performance beyond incomplete benchmark annotations. Even if the GENIA results show higher uncertainty due to domain complexity and anno-

tator expertise limitations, results are consistent across both datasets: high agreement between human and LLM judge, modest positive bias, and strong PPI-corrected precision.

4.3 Deeper Analysis

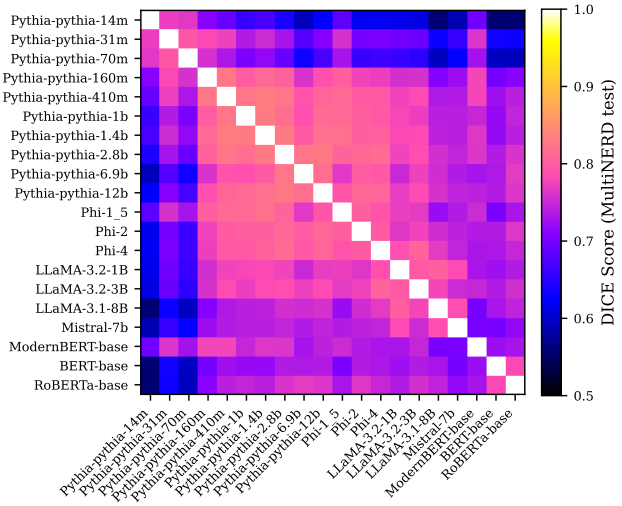

Figure 3: DICE score between inference of ToMMeR trained on various LLMs on MultiNERD test (Results for GENIA are similar, see Appendix A, fig. 8).

LLMs share a common Notion of Entity Mention. If two models capture the same underlying notion of entity, their predicted entity mention sets should exhibit a high degree of overlap. We measure this similarity by computing the Sørensen–Dice coefficient (Dice, 1945) for mention predictions across all tested LLMs (See Figure 3). We consider LLMs ranging from 14M to 15B parameters, providing a broad basis to study the effect of scale while keeping resource use manageable. For decoder-only models, we use the Pythia family, which offers a controlled size sweep for systematic scaling studies. We also include LLaMA 3 models (1B, 3B, and 8B) and MISTRAL-7B, representing recent state-of-the-art open LLMs. We further include the PHI family –trained with textbook-style synthetic data– (see in appendix, Table 6 for details about all the models considered). To test transferability beyond decoder-only settings, we additionally evaluate encoder-only architectures, including BERT and RoBERTa as established NER baselines, and ModernBERT as a more recent encoder. We observe that BERT family offers the best model size–F1 trade-off when considering the aggregated on the 13 benchmarks.

Auto-regressive backbones (excluding small Pythia models with <160M p.) reach Dice >75%, indicating strong agreement in their learned notion of mentions. This suggests that diverse LLMs—despite differences in scale or training data—develop a convergent and architecture-agnostic representation of entity boundaries. Encoder-only models predictions slightly differ from decoder-based architectures (Dice drop of $\sim$ 10%). From the results, we hypothesize that they predict less nested entities due to their bidirectional attention mechanism, which may suppress overlapping spans in favor of flat, non-hierarchical segmentation. More results in Appendix Figure 12.

Mention detection capabilities through layers. To localize where mention detection capability emerges within the network, we train a ToMMeR model on the hidden representations extracted from each layer of the backbone LLM (here LLAMA-3.2-1B). Maximum performance is nearly reached after the first layer of the transformer (Figure 5). This suggests that mention-detection signals are established very early in the model's computation and remain largely stable across intermediate layers.

We nonetheless observe a noticeable loss at the final layer, where the model likely discards these signals to prioritize next-token prediction. We also show the Dice similarity scores across layers in appendix (fig. 6), showing that entity-related signals are not only learned early in the network but also remain largely consistent across layers.

5 Full NER via Span Classification

To further assess the utility of ToMMeR as a generalist mention detection framework, we conduct an extrinsic evaluation by applying it within a complete NER pipeline that includes entity typing.

Approach. For each NER benchmark, we first finetune ToMMeR to align with its notion of entity mention, ensuring high recall on the benchmark. Then, for each predicted mention, we compute a d-dimensional span embedding used for classification: Following Zaratiana et al. (2023), we compute this embedding by passing the concatenation of the spans first and last tokens representations through a two-layer perceptron (MLP) of hidden dimension 1024. We train this representation layer on the benchmark schema using standard Cross-Entropy loss, an additional `0` (not-entity) label is added to filter mentions detected by ToMMeR but not in the benchmarks schema. This lightweight architecture (7.4M params) enables us to adapt our ToMMeR model (using LLAMA-3.2-1B representations) to

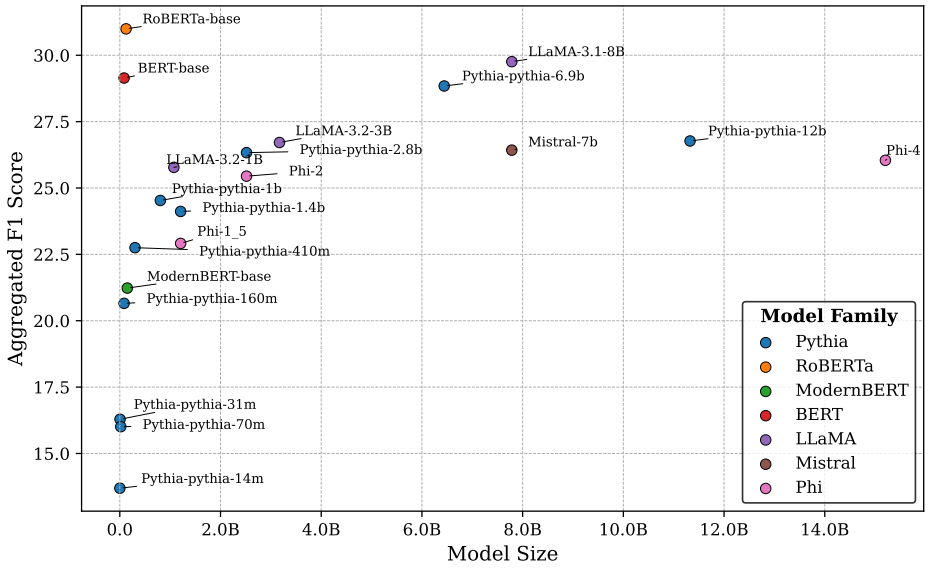

Figure 4: F1 Score of ToMMeR Models —aggregated on the 13 benchmarks considered in this work- versus number of parameters of LLM backbone. We also plot the precision versus recall for all those models in appendix Figure 12.

Model	Backbone	#Trained Params	CoNLL 2003	GENIA	MultiNERD	OntoNotes	ncbi
GLiNER (2023)	deBERTa-v3	209M	**88.7**	**78.9**	**93.8**	**89.0**	**87.8**
EMBER (2024)	GPT2-xl (0–47)	11.5M	85.1	–	–	79.3	–
ToMMeR (ours) + span embed (ℓ11)	LLaMA-3.2-1B (ℓ6)	7.6M	84.8	66.5	92.2	80.4	78.1
	LLaMA-3.2-3B (ℓ5)	7.7M	86.8	69.3	**93.3**	81.7	**82.1**
	LLaMA-3.1-8B (ℓ5)	7.9M	85.0	**70.1**	92.4	80.0	80.8
	RoBERTa-base (ℓ5)	7.4M	**87.3**	67.8	92.6	**85.4**	74.8
	BERT-base (ℓ3)	7.4M	85.0	66.5	90.4	82.1	77.3

Table 3: Micro-F1 scores when training a classification head on top of a ToMMeR model to perform full NER. Baselines are reported from Zaratiana et al. (2023) and Popovic and Färber (2024). We also report in parenthesis the transformer layers probed to perform mention detection.

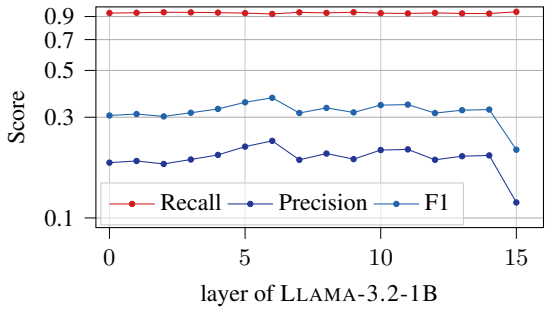

Figure 5: Layer-wise Performance on LLAMA3.2-1B. Recall, precision and F1 Score of ToMMeR probing representations across the 16 layers (0-15) of LLAMA3.2-1B. Performance is nearly optimal from layer 0 onward.

CoNLL in less than 20 minutes on a V100-32Gb.

Results. We benchmark ToMMeR using a range of backbone LLMs, and report the results in Table 3. Despite being attached to autoregressive models that cannot exploit right-side context, ToMMeR achieves near SOTA performance on multiple datasets. On OntoNotes, for instance, the LLAMA-3.2-1B–based ToMMeR reaches an F1 score of 80.4%, even though its initial zero-shot mention detection recall was only 73% (Table 1). This shows ToMMeR can effectively adapt to a dataset-specific notion of entity mention during fine-tuning. In comparison to EMBER (Popovic and Färber, 2024) –trained end-to-end on the NER task and identifying mention detection as the main performance bottleneck– ToMMeR achieves comparable performance. This further supports our approach of probing existing mention detection capabilities in LLMs and demonstrates their utility for downstream NER tasks. We extend our analysis to encoder-only architectures, which are both more parameter-efficient and well-suited for NER due to their bidirectional attention mechanisms. Interestingly, these models do not consistently outperform their autoregressive counterparts under the ToMMeR + typing setup, suggesting that entity-level representations may emerge differently across architectural families. Moreover, larger LLMs may also encode a broader spectrum of entity types.

6 Conclusion

We showed that mention detection capabilities —crucial for information extraction— can be efficiently probed from early LLM layers using less than 300K parameters. Our work provides both practical and conceptual contributions: practically, it offers a lightweight, transferable method for high-coverage mention detection that can be plugged into any LLM; conceptually, it provides evidence that LLMs develop structured mention representations in their early layers that can be recovered through simple probing mechanisms. ToMMeR achieves 93% recall across 13 diverse NER benchmarks, transfers well zero-shot to other latin languages, while maintaining an estimated 90% precision using a single partial forward pass, enabling real-time streaming deployment with minimal overhead (Efficiency analysis in Appendix B). Across models from 14M to 15B parameters, we find that diverse architectures converge on a shared notion of entity mention, producing highly consistent spans. This suggests mention tracking emerges as a byproduct of language modeling rather than as an artifact of a particular architecture. When extended with a typing head, ToMMeR achieves competitive full NER performance despite using auto-regressive models lacking right-side context. Unlike costly prompt-based extraction methods, ToMMeR reduces costs by orders of magnitude (42× faster than prompting) while maintaining flexibility to adapt to any downstream schema or knowledge base. ToMMeR's schema-agnostic design integrates naturally into modern RAG/IE pipelines as a first-stage mention filter, enabling entity-aware document chunking, or providing spans for downstream typing and linking.

Limitations

Lack Ground Truth for Untyped Mention Detection. The fundamental challenge in evaluating our approach is the lack of established ground truth for untyped mention detection. While typed NER benchmarks provide gold annotations, they only label entities from specific ontologies (e.g., person, location, organization), making it unclear whether unlabeled spans are true negatives or simply out-of-scope entities. We address this by using LLM-based judgment aligned with Wikipedia's entity definition, but this introduces its own limitations: (1) `gpt-4.1-mini` is slightly biased towards positive judgements (see Section D) (2) the Wikipedia definition is broad and potentially ambiguous in edge cases. While human validation shows reasonable agreement with the LLM judge (87.7% agreement on 1800 annotated mentions), comprehensive human annotation would be needed to definitively establish ToMMeR precision. Our PPI analysis however bounds the true underlying precision to above 89%.

Architectural Constraints. Our focus on continuous spans excludes discontinuous entities (e.g., "New York" and "City" separated by other tokens), which appear in some linguistic phenomena and specialized domains. Additionally, our auto-regressive models lack access to right-side context, potentially missing boundary information that bidirectional encoders naturally capture. While Section 5 shows that competitive full NER performance is still achievable, this architectural limitation may impact mention detection quality compared to encoder-only approaches.

Dataset and Training Limitations. Training on Pile-NER, despite its broad coverage, inherits the biases and gaps of GPT-3.5's annotations from 2023d. The distillation strategy, while mitigating some incompleteness, risks amplifying systematic biases present in the initial annotations.

Schema Generalization. While we evaluate across 13 benchmarks (spanning news, biomedical and general domains) and obtain strong average recall. The ACE 2005 results (42% recall) demonstrate that our models struggle with annotation conventions that differ substantially from Pile-NER, such as including determiners in entity spans. This suggests that while the learned notion of mentions generalizes well across most domains, it may not align with all possible annotation schemes, although we also show that ToMMeR can easily further be tuned to fit to such schemes.

Acknowledgements

The authors acknowledge the ANR – FRANCE (French National Research Agency) for its financial support of the GUIDANCE project n°ANR-23-IAS1-0003 as well as the Chaire Multi-Modal/LLM ANR Cluster IA ANR-23-IACL-0007. This work was granted access to the HPC resources of IDRIS under the allocation 2024-AD011015440R1 made by GENCI. The authors also gratefully acknowledge the support of the Centre National de la Recherche Scientifique (CNRS)

through a research delegation awarded to J. Mothe.

References

Zeyuan Allen-Zhu and Yuanzhi Li. 2023. Physics of language models: Part 1, learning hierarchical language structures. Available at SSRN: https://ssrn.com/abstract=5250639. SSRN working paper.

Anastasios N. Angelopoulos, Stephen Bates, Clara Fannjiang, Michael I. Jordan, and Tijana Zrnic. 2023. Prediction-Powered Inference. *Preprint*, arXiv:2301.09633.

Fan Bai, Hamid Hassanzadeh, Ardavan Saeedi, and Mark Dredze. 2025. LLMs are better than you think: Label-guided in-context learning for named entity recognition. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 28372–28392, Suzhou, China. Association for Computational Linguistics.

Junyi Bian, Rongze Jiang, Weiqi Zhai, Tianyang Huang, Hong Zhou, and Shanfeng Zhu. 2023. Dmner: Biomedical entity recognition by detection and matching. *Preprint*, arXiv:2306.15736.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Lee R. Dice. 1945. Measures of the Amount of Ecologic Association Between Species. *Ecology*, 26(3):297–302.

Jiahai Feng and Jacob Steinhardt. 2024a. How do language models bind entities in context? arXiv preprint arXiv:2310.17191. Revised May 2024.

Jiahai Feng and Jacob Steinhardt. 2024b. How do Language Models Bind Entities in Context? *Preprint*, arXiv:2310.17191.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020a. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020b. The pile: An 800gb dataset of diverse text for language modeling. *Preprint*, arXiv:2101.00027.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting Recall of Factual Associations in Auto-Regressive Language Models. *Preprint*, arXiv:2304.14767.

Ralph Grishman. 2005. Ace semantic structures. Online documentation. Describes the seven entity types used in ACE: person, organization, GPE, location, facility, vehicle and weapon.

Ayoub Hammal, Benno Uthayasooriyar, and Caio Corro. 2025. Few-shot domain adaptation for named-entity recognition via joint constrained k-means and subspace selection. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 9902–9916, Abu Dhabi, UAE. Association for Computational Linguistics.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. 2014. Ncbi disease corpus: A resource for disease name recognition and concept normalization. *Journal of Biomedical Informatics*, 47:1–10.

Yizhu Jiao, Jinah Kim, Zhengyu Liu, Fei Tian, Michael Zhou, Vineet Reddi, Wenlin Yu, Ronan Fagin, Pengcheng Chen, Peter West, Alexander Koller, and Peter Clark. 2023. Instruct and extract: Instruction tuning for on-demand information extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10030–10051, Singapore, Singapore. Association for Computational Linguistics.

Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Nigel Collier. 2004. Introduction to the bio-entity recognition task at jnlpba. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA)*, pages 70–75, Geneva, Switzerland. Association for Computational Linguistics.

Najoung Kim and Sebastian Schuster. 2023. Entity tracking in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3835–3855, Toronto, Canada. Association for Computational Linguistics.

Aman Kumar and Binil Starly. 2021. Fabner: Dataset_ner_manufacturing. Figshare dataset. Accessed 2025-10-06.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197. Association for Computational Linguistics.

Jing Li, Deheng Ye, and Shuo Shang. 2019. Adversarial transfer for named entity boundary detection

with pointer networks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, IJCAI-19, pages 5053–5059, Macao, China. International Joint Conferences on Artificial Intelligence Organization.

Xincheng Liao, Junwen Duan, Yixi Huang, and Jianxin Wang. 2025. Ruie: Retrieval-based unified information extraction using large language model. arXiv preprint arXiv:2409.11673. To appear at COLING 2025.

Zihan Liu, Yan Xu, Tiezhen Yu, Wenliang Dai, Ziwei Ji, Samuel Cahyawijaya, Andrea Madotto, and Pascale Fung. 2020. Crossner: Evaluating cross-domain named entity recognition. *Preprint*, arXiv:2012.04373.

Jie Lou, Yaojie Lu, Dai Dai, Wei Jia, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2023. Universal information extraction as unified semantic matching. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press.

Jing Lu and Vincent Ng. 2021. Span-based event coreference resolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 14131–14138.

Keming Lu, Xiaoman Pan, Kaiqiang Song, Hongming Zhang, Dong Yu, and Jianshu Chen. 2023. Pivoine: Instruction tuning for open-world information extraction. arXiv preprint arXiv:2305.14898.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5755–5772, Dublin, Ireland. Association for Computational Linguistics.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and Editing Factual Associations in GPT. *Advances in Neural Information Processing Systems*, 35:17359–17372.

Lesly Miculicich and James Henderson. 2020. Partially-supervised mention detection. In *Proceedings of the Third Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 91–98, Barcelona, Spain (online). Association for Computational Linguistics.

Philipp Mondorf, Sondre Wold, and Barbara Plank. 2025. Circuit compositions: Exploring modular structures in transformer-based language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14934–14955, Vienna, Austria. Association for Computational Linguistics.

Victor Morand, Josiane Mothe, and Benjamin Piwowarski. 2025. On the representations of entities in auto-regressive large language models. In *Proceedings of the 8th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, and 7 others. 2022. In-context learning and induction heads. arXiv preprint arXiv:2209.11895.

OpenAI. 2025. Openai api. `https://platform.openai.com/`. Accessed: 2025-09-28.

Nicholas Popovic and Michael Färber. 2024. Embedded named entity recognition using probing classifiers. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17830–17850, Miami, Florida, USA. Association for Computational Linguistics.

Nicholas Popovic and Michael Färber. 2024. Embedded named entity recognition using probing classifiers. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17830–17850, Miami, Florida, USA. Association for Computational Linguistics.

Masaki Sakata, Benjamin Heinzerling, Sho Yokoi, Takumi Ito, and Kentaro Inui. 2025. On entity identification in language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16717–16741, Vienna, Austria. Association for Computational Linguistics.

Or Shachar, Uri Katz, Yoav Goldberg, and Oren Glickman. 2025. Ner retriever: Zero-shot named entity retrieval with type-aware embeddings. arXiv preprint arXiv:2509.04011. Findings of EMNLP 2025.

Jingbo Shang, Liyuan Liu, Xiaotao Gu, Xiang Ren, Teng Ren, and Jiawei Han. 2018. Learning named entity tagger using domain-specific dictionary. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2054–2064, Brussels, Belgium. Association for Computational Linguistics.

Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. arXiv preprint arXiv:2001.07685. Published at NeurIPS 2020.

Simone Tedeschi, Valentino Maiorca, Niccolò Campolungo, Francesco Cecconi, and Roberto Navigli. 2021. Wikineural: Combined neural and knowledge-based silver data creation for multilingual ner. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2521–2533, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Simone Tedeschi and Roberto Navigli. 2022. Multinerd: A multilingual, multi-genre and fine-grained dataset for named entity recognition (and disambiguation). In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 783–795, Seattle, USA. Association for Computational Linguistics.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Aman Singh Thakur, Kartik Choudhary, Venkat Srinik Ramayapally, Sankaran Vaidyanathan, and Dieuwke Hupkes. 2025. Judging the judges: Evaluating alignment and vulnerabilities in LLMs-as-judges. In *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM²)*, pages 404–430, Vienna, Austria and virtual meeting. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *Preprint*, arXiv:cs/0306050.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. Linguistic Data Consortium. Contains about 1 800 files across English, Chinese and Arabic annotated for entities, relations and events.

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, Jihua Kang, Jingsheng Yang, Siyuan Li, and Chunsai Du. 2023. Instructuie: Multi-task instruction tuning for unified information extraction. arXiv preprint arXiv:2304.08085.

Yucheng Wang, Dinghan Jiang, Yizeng Li, Zhiting Yi, Jinah Kim, Chen Li, Qianyuan Qiu, Wenlin Yu, Ronan Fagin, Pengcheng Chen, Peter West, Alexander Koller, and Peter Clark. 2024. True-uie: Two universal relations unify information extraction tasks. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1863–1876, Mexico City, Mexico. Association for Computational Linguistics.

Ralph Weischedel, Sameer Pradhan, Lance Ramshaw, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Nianwen Xue, Martha Palmer, Jena Hwang, Claire Bonial, Jinho Choi, and 1 others. 2013. Ontonotes release 5.0. Technical Report LDC2013T19, Linguistic Data Consortium, Philadelphia, USA.

Wikipedia contributors. 2025. Entity — Wikipedia. [Online; accessed 28-September-2025].

Xinglin Xiao, Yijie Wang, Nan Xu, Yuqi Wang, Hanxuan Yang, Minzheng Wang, Yin Luo, Lei Wang, Wenji Mao, and Daniel Zeng. 2024. Yayi-uie: A chat-enhanced instruction tuning framework for universal information extraction. arXiv preprint arXiv:2312.15548.

Tingyu Xie, Jian Zhang, Yan Zhang, Yuanyuan Liang, Qi Li, and Hongwei Wang. 2024. Retrieval augmented instruction tuning for open ner with large language models. arXiv preprint arXiv:2406.17305. To appear at COLING 2025.

Urchade Zaratiana, Gil Pasternak, Oliver Boyd, George Hurn-Maloney, and Ash Lewis. 2025. GLiNER2: An efficient multi-task information extraction system with schema-driven interface. arXiv preprint arXiv:2507.18546. Submitted 24 Jul 2025.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2023. GLiNER: Generalist Model for Named Entity Recognition using Bidirectional Transformer. *Preprint*, arXiv:2311.08526.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2024. GLiNER: Generalist model for named entity recognition using bidirectional transformer. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5364–5376, Mexico City, Mexico. Association for Computational Linguistics.

Duzhen Zhang, Hongliu Li, Wei Cong, Rongtao Xu, Jiahua Dong, and Xiuyi Chen. 2023. Task relation distillation and prototypical pseudo label for incremental named entity recognition. arXiv preprint arXiv:2308.08793. Accepted as a long paper (oral) at CIKM 2023.

Haoyu Zhao, Abhishek Panigrahi, Rong Ge, and Sanjeev Arora. 2023. Do transformers parse while predicting the masked word? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16513–16542, Singapore, Singapore. Association for Computational Linguistics.

Ran Zhou, Mengjie Hu, Yuying Guo, Yang Wang, Yipeng Cao, Erhao Xie, Yue Zhang, and Qi Li. 2023a. Improving self-training for cross-lingual named entity recognition with contrastive and prototype learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4018–4031, Toronto, Canada. Association for Computational Linguistics.

Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. 2023b. Pile-NER: Gpt-annotated named entity recognition dataset from The Pile. `https://huggingface.co/datasets/Universal-NER/Pile-NER-type`. Dataset; accessed 2025-10-07.

Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. 2023c. UniversalNER: Targeted Distillation from Large Language Models for Open Named Entity Recognition. In *The Twelfth International Conference on Learning Representations*.

Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. 2023d. UniversalNER: Targeted distillation from large language models for open named entity recognition. arXiv preprint arXiv:2308.03279. Version v2, January 2024.

A Additional Examples and Figures

A.1 Figures

Figure 6 presents the DICE score between the sets of entities inferred on the MultiNERD (test split) for ToMMeR using all possible layers of LLAMA3.2-1B. Figure 7 presents the results for GENIA. Figure 8 is also the twin figure of Figure 3 in the same Sub-section.

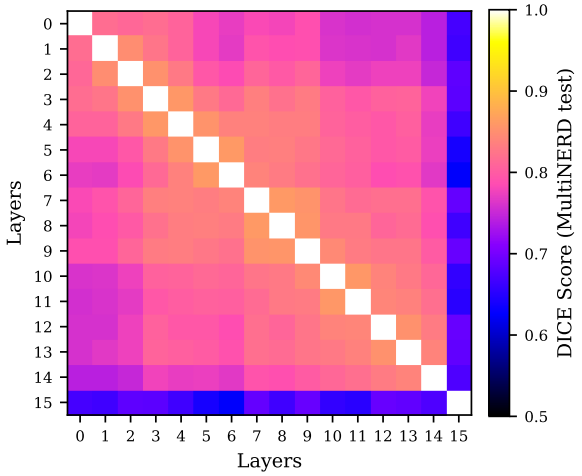

Figure 6: DICE score between the sets of entities inferred on the MultiNERD (test), for ToMMeR models probing each layer of LLAMA3.2-1B. Results for GENIA are similar, and available in appendix, Figure 7.

A.2 Pile-NER statistics

Figure 9 reports the distribution of texts token-length of in Pile-NER used in ToMMeR entity mention boundary training. While Figure 10 reports the distribution of entity mention lengths.

A.3 Qualitative examples

We provide in Section A.3 some qualitative examples of our models prediction on the ACE benchmark, where ToMMeR has lower zero-shot recall as discussed in Section 4.1, suggesting that ToMMeR learned a slightly different notion of entity mention from Pile-NER.

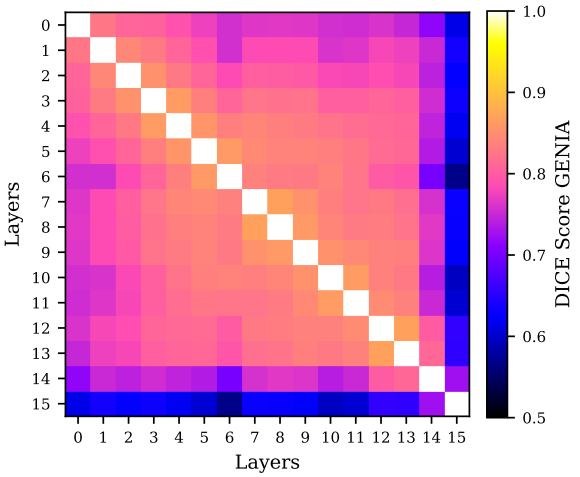

Figure 7: DICE score between inference for models using all possible layers of LLAMA3.2-1B, on the full GENIA dataset, results are similar to those in Figure 6.

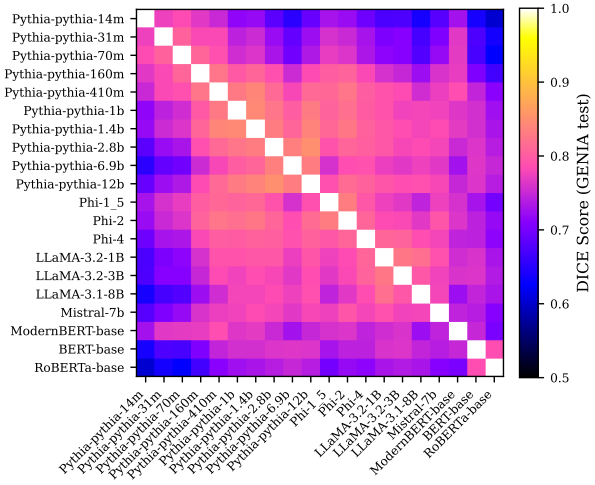

Figure 8: DICE score between inference of ToMMeR trained on various existing LLMs on GENIA test.

B Complexity and efficiency analysis

Complexity Unlike prompting methods that require auto-regressive generation ($O(N \cdot M)$ complexity where M is output length), ToMMeR classifies spans in a single parallelizable pass ($O(N)$). It is computationally equivalent to one additional attention head. On a standalone usage, ToMMeR is faster than one backbone forward pass. As we can cut the computation to at a lower layer (e.g layer 6 / 16 for Llama3-1B gives a $\approx 2,6\times$ speedup). And even faster when compared with prompting methods using several forward passes for generation. It furthermore allows for "streaming extraction"—identifying entities as the LLM generates text with almost zero latency penalty (as shown

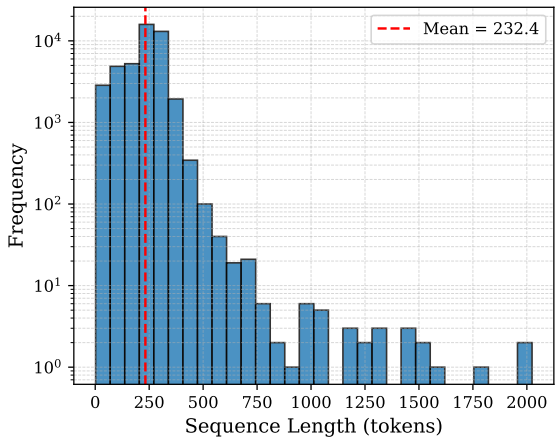

Figure 9: Sample token length distribution in Pile-NER (Zhou et al., 2023c) using LLAMA tokenizer.

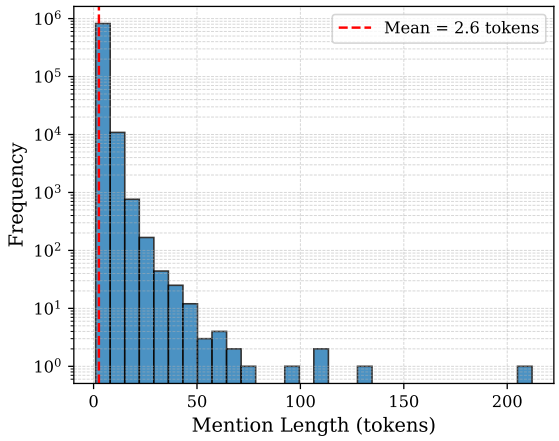

Figure 10: Entity mention length distribution in Pile-NER (Zhou et al., 2023c) using LLAMA tokenizer. We use a sliding window of 25 tokens, which includes 99.8% of mentions annotated in Pile-NER.

by Popovic and Färber). This is can be useful for real-world RAG pipelines where re-prompting the LLM for extraction doubles the cost and latency.

Quantitative Runtime comparison To compare with a prompting baseline, we applied the prompt from UniversalNER (Zhou et al., 2023d) : `"Given a passage, your task is to extract all entities and identify their entity types. The output should be in a list of tuples of the following format: [("entity 1", "type of entity 1"), ... ]. Passage: passage"` with llama3.2-1B. Running it on the passage `"Large language models are awesome. While trained on language modelling, they exhibit emergent abilities that make them suitable for a wide range of tasks, including Named Entity Recognition (NER)."` took **3.21 seconds** (24Bg GPU, using HF transformers, 100max tokens generation), While ToMMeR inference took **only 0.077 seconds, which makes a 42x improvement.**

C Reproducibility Statement

For complete reproducibility, we publish both code at `https://github.com/VictorMorand/llm2ner`, containing detailed hyperparameters and experimental pipeline, and trained ToMMeR models on huggingface. We also list in this section the most important hyper-parameters used in our main experiments.

C.1 Layer experiment

Table 4 details the hyperparameters used in the layer experiment described in Section 4.3, where we train ToMMeR models at each layer of LLAMA-3.2-1B, showing that mention detection signals are computed as early as layer 0 in the transformer.

C.2 Hardware settings

Training the rank 64 ToMMeR model using representations from layer 6 of LLAMA-3.2-1B takes 4 hours on a NVIDIA-H100-80Gb GPU (though it could be run on smaller GPUs, peak GPU memory being only 6GBs) using hyperparameters described in Section C.1. We also leveraged V100-32Gb for evaluations and inference.

C.3 Model experiment

The goal of the model experiment is to compare ToMMeR performance across LLM backbones. To

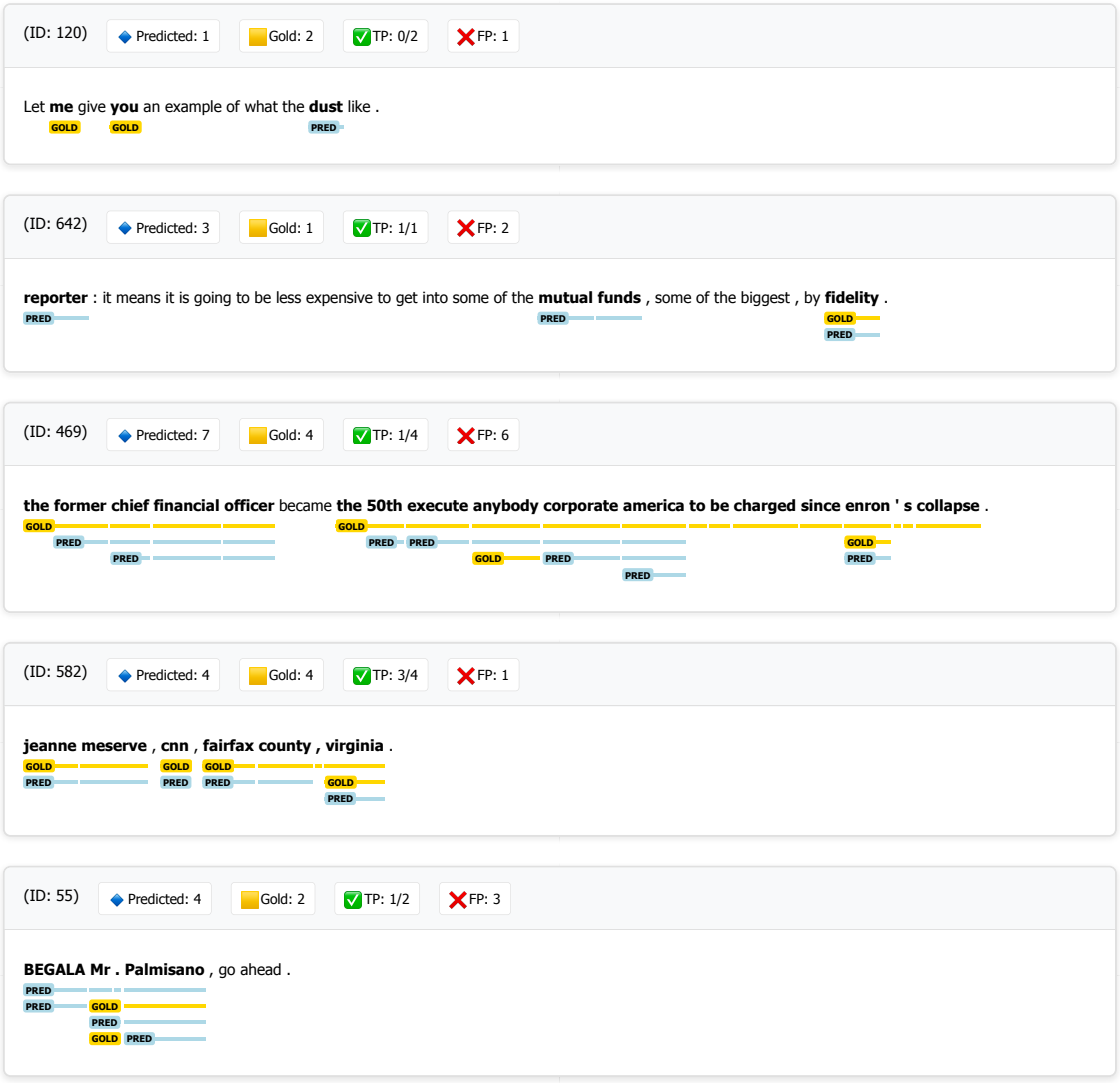

Figure 11: Qualitative examples comparing model predictions and gold annotations, randomly sampled from ACE 2005 Dataset where our ToMMeR model has a surprisingly low Zero Shot Recall (42%, Cf Table 1).

moderate variance, we trained several models, using representations from different layers, we keep the best performing ToMMeR model for each backbone. Hyper-parameters are detailed in Table 5

Models considered in this work We detail all models architecture parameters in Table 6. Please also find in Figure 12 the precision versus recall curve for those models.

D LLM as a Judge

D.1 LLM-as-a-judge

As described in Section 4.2, the goal of this experiment is to have an idea of the real precision of ToMMeR. Given a span predicted by ToMMeR, we prompt `gpt-4.1-mini` (via the OpenAI API (OpenAI, 2025)) to assess whether it qualifies the predicted span as an entity according to the Wikipedia definition (Wikipedia contributors, 2025). While it is hard for the model to predict directly the correct true/false tokens when given the context, we find that letting the model generate a small explanation before answering greatly improves accuracy. The complete prompt is provided Figure 13 along with an example answer.

The prompt used to judge the inference of our ToMMeR models can be found Figure 13. We now give more details on the human validation study.

D.2 Human validation

Human Annotation Protocol. We recruited five annotators from among the authors and colleagues—all informatics researchers, though not all NLP specialists. To ensure consistency, anno-

Table 4: Hyperparameter configuration for the layers experiment.

Parameter	Value	description
model_name	Llama-3.2-1B	LLM backbone
rank	64	rank r of the ToMMeR model
optimizer	AdamW	optimizer to use
epochs	8	number of epochs
batch_size	16	batch size
sliding_window	25	Sliding window
lr	1e-2	learning rate
patience	5000	patience for lr scheduler
accumulation_steps	1	1 for no accumulation
grad_clip	2.0	0 for no clipping
val_metric	"f1"	metric to use for validation
self_distillation_phases	1	number of self-distillation phases
reset_student_weights	true	whether to reset student weights
sparse_distill_loss	true	whether to use sparse distillation loss
teacher_thr_prob	0.90	teacher threshold probability

tators followed identical instructions to the LLM judge, augmented with clarified examples from author-annotated data. The annotation process followed strict blinding protocols: annotators had no access to (1) other annotators' judgments or (2) the LLM's predictions, preventing potential biases. Complete inter-annotator agreement metrics and Cohen's κ statistics are presented in table 7, including per-dataset disagreements breakdowns.

E Architecture variants

Along with the ToMMeR architecture for computing our matching scores m_{ij}, we explored other variants.

Normalization function In Equation 1, we normalize the dot product between query and key projections using cosine similarity (via ℓ_2 normalization), which naturally bounds matching scores to $[0, 1]$. We also experimented with alternative activation functions to normalize the raw dot products: $\arctan(\cdot)$ and $\log\sigma(\cdot)$ (log-sigmoid). However, both alternatives yielded inferior performance in preliminary experiments. Cosine similarity proved most effective, likely because: (1) it provides scale-invariant matching that is robust to variations in representation magnitudes across layers and models, and (2) its geometric interpretation as angular similarity aligns well with the binding hypothesis—tokens that belong to the same mention should have aligned representations in the learned query-key subspace. All reported results use cosine similarity unless otherwise specified.

Linear Transformation of Queries and Keys (LTQK) Since the transformer model already computes query and key vectors $\{q_i^h, k_i^h\}_{h,i} \in \mathbb{R}^{d_h}$ for each attention head $h \in [1, N_h]$ with dimension d_h, we can use them directly instead of the representations for the residual stream to compute new queries and keys for our model. The matching score m_{ij} is then computed as:

$$m_{ij} = \sum_{h=1}^{N_h} \cos\left(W_Q^h\, q_i^h \mid \underbrace{W_K^h\, k_j^h}_{\in\mathbb{R}^r}\right) \in [0,1] \quad (4)$$

With $W_Q^h, W_{K_h}^h \in \mathbb{R}^{r\times d_h}$ as the query and key matrices, the model's queries and keys are already in a lower-dimensional space, making computations lighter and keeping the number of trainable parameters low. For example, using rank $r = 16$ results in only 68K parameters.

Using existing Attention scores (LCAttn) Even further, we can directly leverage the model's attention scores $a_{ij}^h = \langle q_i^h \mid k_j^h \rangle$.

$$m_{ij} = \log\sigma\left(\sum_{l=0,h=1}^{N_L,N_h} w_h^l\, a_{ij}^h\right) \in \mathbb{R} \quad (5)$$

This approach treats attention as a natural proxy for token binding, and have closely been explored

Table 5: Hyperparameter configuration for the model experiment.

Parameter	Value	description
model_name	Llama-3.2-1B	LLM backbone
layer	[1, 3, 5].	layers l of the LLM to extract
rank	64	rank r of the ToM model
optimizer	AdamW	optimizer to use
epochs	8	number of epochs
batch_size	16	batch size
sliding_window	25	Sliding window
lr	1e-2	learning rate
accumulation_steps	1	1 for no accumulation
grad_clip	1.0	Gradient clipping
val_metric	"f1"	metric to use for validation
self_distillation_phases	1	number of self-distillation phases
reset_student_weights	true	whether to reset student weights
sparse_distill_loss	true	whether to use sparse distillation loss
teacher_thr_prob	0.90	teacher threshold probability

in (Popovic and Färber, 2024) assuming that some heads are already specialized for mention detection and that a linear combination of their scores can recover this capability from the model.

Comparing Architectures Aggregated results for LLAMA-3.2-1B (Table 8) show that while LCAttn use far fewer parameters, it fail to match ToMMer's overall performance. LTQK maintains high recall but loses precision, while LCAttn's precision drops dramatically. This highlights a clear tradeoff: extreme parameter reduction can preserve recall, but strong precision requires more capacity.

F Supplementary Ablation Studies

F.1 Impact of Rank

We show Figure 14 the precision-recall balance when varying rank r as defined in Section 2, we chose a final rank of 64 in our other experiments, as it is a good balance between recall and precision, while maintaining a small number of parameters in ToMMeR.

G Multi-Head Self Attention Model

In transformer-based language modeling, Multi Head Self Attention (MHSA) is the (Vaswani et al., 2017) main mechanism used to transfer information between token representations. We also tried to use a key and query model using an MHSA layer rather than raw projections, enabling to model contexual cues into queries and keys. We compute the entity span logit probability m_{ij} as :

$$m_{ij} = \text{MHSA}_Q(\boldsymbol{z}_i) \cdot \text{MHSA}_K(\boldsymbol{z}_j) \quad (6)$$

However, training was much longer and performance metrics were lower than using the probability in Section 2 of this paper.

H Dataset descriptions

H.1 MultiNERD

Languages and domains. MultiNERD (Tedeschi and Navigli, 2022) is a language–agnostic dataset that automatically annotates texts from Wikipedia and WikiNews in ten languages (Chinese, Dutch, English, French, German, Italian, Polish, Portuguese, Russian and Spanish). **Entity types.** It provides fine–grained annotation for 15 entity categories (*person*, *location*, *organization*, *animal*, *biological entity*, *celestial body*, *disease*, *event*, *food*, *instrument*, *media*, *plant*, *mythical entity*, *time* and *vehicle*) and adds disambiguation links to the corresponding Wikipedia pages (Tedeschi and Navigli, 2022). The annotations are created by combining WikiNEuRal silver-data creation and NER4EL fine–grained labeling, resulting in a high-quality multi-genre resource.

H.2 CoNLL-2003

The CoNLL-2003 shared task (Tjong Kim Sang and De Meulder, 2003) provides a benchmark for language–independent named entity recognition.

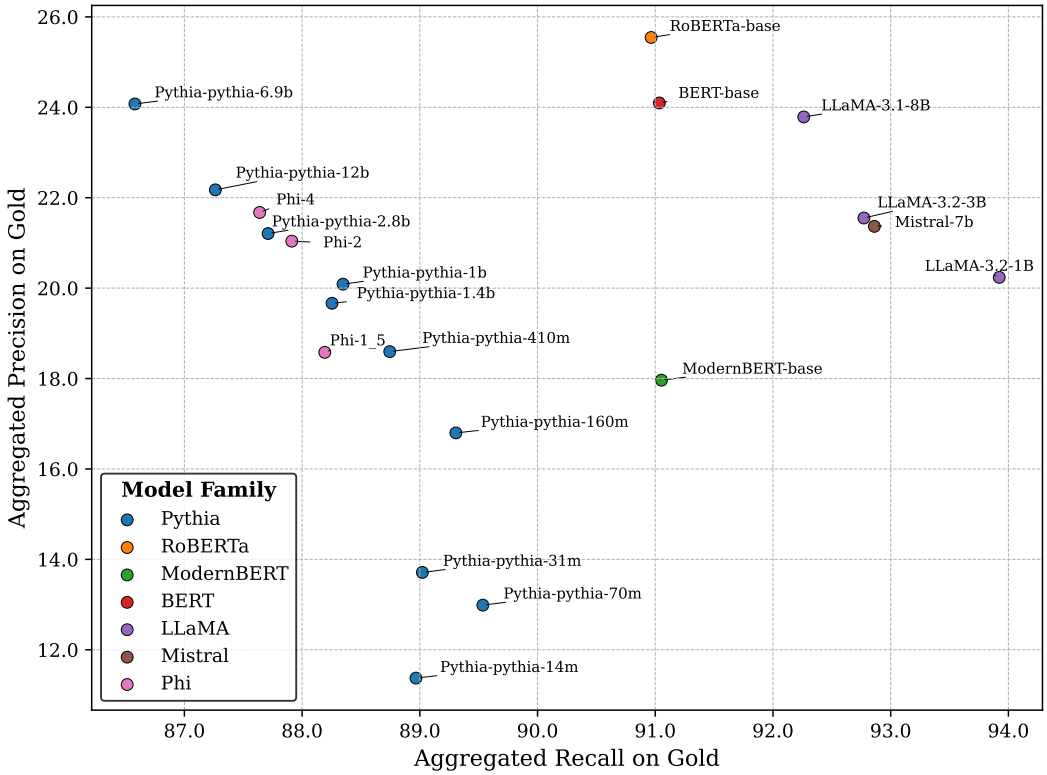

Figure 12: Aggregated precision vs recall of ToMMeR Models versus number of parameters of LLM backbone.

It contains English and German newswire articles taken from Reuters and Frankfurter Rundschau. Four entity classes (*person*, *location*, *organization* and *miscellaneous*) are annotated. The English portion comprises 946 training articles, 216 development articles and 231 test articles, while the German portion contains 518 training, 52 development and 342 test articles (Tjong Kim Sang and De Meulder, 2003).

H.3 CrossNER

CrossNER (Liu et al., 2020) is a cross-domain NER dataset covering five specialist domains: *politics*, *artificial intelligence*, *music*, *literature* and *science*. Each domain contains a small labelled training set (100–200 documents) and roughly 1,000 development and test sentences. Entity types are tailored to each domain (e.g., *politician*, *election*, *software*, *research field*), and unlabeled domain-specific corpora are provided for domain adaptation. This resource is used to evaluate whether models generalise across domains with distinct entity inventories (Liu et al., 2020).

H.4 NCBI Disease corpus

The NCBI disease corpus (Islamaj Doğan et al., 2014) contains 793 PubMed abstracts that are fully annotated at both the mention and concept levels for disease names. Manual annotation produced 6,892 disease mentions mapped to 790 unique disease concepts. Approximately 88 % of concepts link to a MeSH entry and 91 % of mentions correspond to a single concept (Islamaj Doğan et al., 2014). The corpus is split into training, development and test sets and serves as a benchmark for biomedical NER and concept normalisation.

H.5 FabNER

FabNER (Kumar and Starly, 2021) is a manufacturing domain corpus containing over 350,000 words of scientific abstracts from the Web of Science. Each word is labelled with one of 12 categories covering materials (*MATE*), manufacturing processes (*MANP*), machines/equipment (*MACEQ*), applications (*APPL*), features (*FEAT*), mechanical properties (*PRO*), characterisation techniques (*CHAR*), parameters (*PARA*), enabling technology (*ENAT*), concepts or principles (*CONPRI*), manufacturing standards (*MANS*) and biomedical concepts (*BIOP*) (Kumar and Starly, 2021). Annotations follow the BIOES tag scheme.

Model name	# Params	N layers	Model dim	Context length	N Vocab
EleutherAI/pythia-14m	1.2M	6	128	2048	50304
EleutherAI/pythia-31m	4.7M	6	256	2048	50304
EleutherAI/pythia-70m	19M	6	512	2048	50304
EleutherAI/pythia-160m	85M	12	768	2048	50304
EleutherAI/pythia-410m	302M	24	1024	2048	50304
EleutherAI/pythia-1b	805M	16	2048	2048	50304
EleutherAI/pythia-1.4b	1.2B	24	2048	2048	50304
EleutherAI/pythia-2.8b	2.5B	32	2560	2048	50304
EleutherAI/pythia-6.9b	6.4B	32	4096	2048	50432
EleutherAI/pythia-12b	11B	36	5120	2048	50688
meta-llama/Llama-3.1-8B	7.8B	32	4096	2048	128256
meta-llama/Llama-3.2-1B	1.1B	16	2048	2048	128256
meta-llama/Llama-3.2-3B	3.2B	28	3072	2048	128256
mistralai/Mistral-7B-v0.1	7.8B	32	4096	2048	32000
microsoft/phi-1_5	1.2B	24	2048	2048	51200
microsoft/phi-2	2.5B	32	2560	2048	51200
answerdotai/ModernBERT-base	149M	22	768	8192	50368
FacebookAI/roberta-base	125M	12	768	512	
google-bert/bert-base-uncased	85M	12	768	512	30522

Table 6: Architecture characteristics of the LLMs considered in this work.

Dataset	Agreement Rate (%)	Correct / Total	Cohen's κ	Human=True LLM=False	Human=False LLM=True
GENIA	78.50	471 / 600	0.239	49	80
MultiNERD	91.50	1098 / 1200	0.449	39	63
Aggregated	**87.17**	**1569 / 1800**	**0.359**	**88**	**143**

Table 7: Agreement rates and disagreement analysis between human annotators and the LLM across datasets. While the model exhibits a tendency toward slight over-prediction, it can be regarded as a reasonably reliable evaluator.

type	layer	Agg R	Agg P	Agg F1	LLM P	#params
ToM	6	92.6	**23.2**	**37.1**	**92.5**	264 198
LTQK	0	**94.4**	17.6	29.7	78.8	165 894
LCAttn	0-10	**94.3**	9.8	17.8	44.2	2 279

Table 8: Comparison of all tested architectures (all models are trained on LLAMA-3.2-1B). We report the Precision (P) Recall (R) and F1 (F1) aggregated on all 13 tested benchmarks. LLM P is the mean precision on LLM annotated data splits.

H.6 WikiNEuRal

WikiNEuRal (Tedeschi et al., 2021) generates silver-standard NER training data by combining neural models and the BabelNet knowledge base. It produces automatically labelled corpora for nine languages (Dutch, English, French, German, Italian, Polish, Portuguese, Russian and Spanish) using Wikipedia articles. The method improves span-based F1 scores by up to six points over previous approaches for multilingual silver data creation (Tedeschi et al., 2021).

H.7 OntoNotes 5.0

OntoNotes 5.0 (Weischedel et al., 2013) is a large multi-layer corpus containing annotations for syntax, predicate–argument structure, word sense, coreference and named entities across English, Chinese and Arabic. The NER layer defines 18 categories, including *PERSON*, *NORP*, *FACILITY*, *ORGANIZATION*, *GPE*, *LOCATION*, *PRODUCT*, *EVENT*, *WORK OF ART*, *LAW*, *LANGUAGE*, *DATE*, *TIME*, *PERCENT*, *MONEY*, *QUANTITY*, *ORDINAL* and *CARDINAL* (Weischedel et al., 2013). The English portion comprises approximately 300 k words of newswire, 200 k words each of broadcast news and broadcast conversation, 200 k words of web text and 100 k words of telephone conversations (Weischedel et al., 2013). Similar corpora are provided for Chinese and Arabic, making OntoNotes one of the largest resources for multi-genre NER.

```
[System:]
You are an expert in entity mention annotation.
A mention is defined as : "something that exists as itself. It does not need  to be
    of material existence."
In particular, abstractions and legal fictions are usually regarded as entities.  In
     general, there is also no presumption that an entity is animate, or present. It
     may  refer to animals; natural features such as mountains; inanimate objects
    such as tables;  numbers or sets as symbols written on a paper; human
    contrivances such as laws, corporations and academic disciplines; or
    supernatural beings such as gods and spirits."

## Instructions
- For each text span provided in [[...]], quickly determine if it is a valid mention
     as  defined above, regardless of its type, length, or style, but ensuring it is
     not a fragment.
- Briefly explain in one concise sentence whether the span fits the definition. Then
     answer with a clear "yes" or "no".

[User:]
"...here that she met her future [[second husband]] , Gottfried Lessing ..."
----------------------------------------
[Answer: ]
"The span "second husband" refers to a specific person as a distinct entity, fitting
     the definition of a mention. Yes"
Parsed answer: TRUE
```

Figure 13: Prompt used to annotate our NER inference data using OpenAI API (`gpt-4.1-mini`, accessed September 2025) (OpenAI, 2025). we use the definition of entity from Wikipedia (Wikipedia contributors, 2025), more details in Section D.

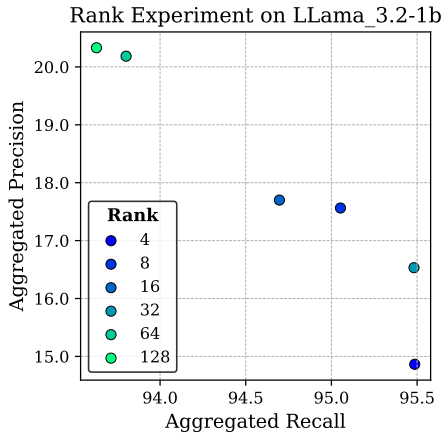

Figure 14: Aggregated precision vs recall of ToMMeR Models accross ranks r as defined in Section 2. We choose 64 for other experiments, yielding a good balance between recall and precision, while maintaining a small number of parameters in ToMMeR.

H.8 ACE 2005

The Automatic Content Extraction (ACE) 2005 corpus (Walker et al., 2006) contains around 1,800 documents in English, Chinese and Arabic drawn from newswire, broadcast news, broadcast conversation, weblogs, discussion forums and conversational telephone speech (Walker et al., 2006). Entities are annotated with seven types—*person*, *organization*, *geo-political entity (GPE)*, *location*, *facility*, *vehicle* and *weapon*—and each entity may have multiple mentions (names, nominals or pronouns) in a document (Grishman, 2005). The corpus also includes annotations for relations and events, but we only use the entity annotations in our experiments.

H.9 GENIA

The GENIA corpus version 3.02 (Kim et al., 2004) consists of 2,000 MEDLINE abstracts selected using the MeSH keywords 'human', 'blood cells' and 'transcription factors' and annotated with a fine-grained taxonomy of 36 entity classes (Kim et al., 2004). For the JNLPBA shared task the 36 classes were mapped to five coarse categories: *protein*, *DNA*, *RNA*, *cell line* and *cell type*. An additional 404 abstracts were annotated for testing, giving a total of 2,404 abstracts with over 100,000 tokens (Kim et al., 2004).

I Schema comparison

ACE2005, MultiNERD, and PileNER differ mainly in what they treat as an entity mention. ACE2005 uses an entity-centric annotation scheme with a small closed ontology, but a broad notion of mention: it annotates seven core entity types, distinguishes named, nominal, and pronominal mentions,

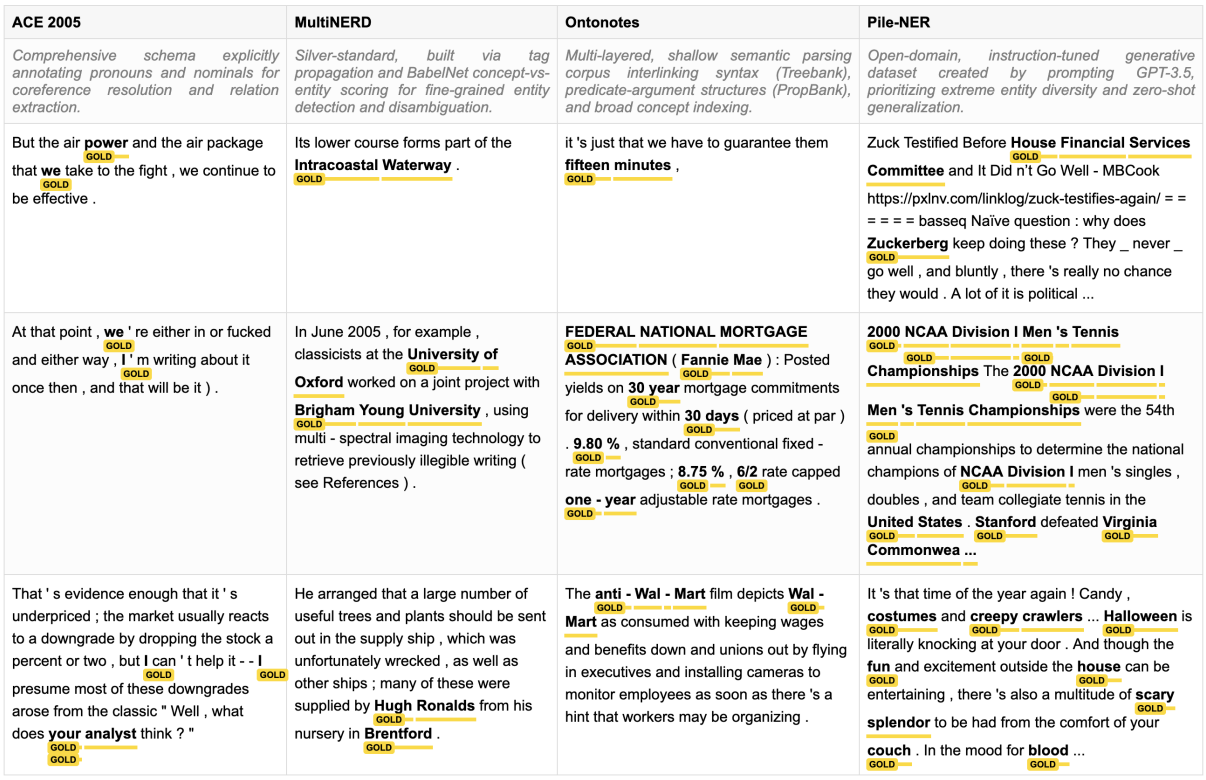

Figure 15: A comparative visualization of gold-standard annotations across four distinct benchmarks. The examples illustrate shifting definitions of what constitutes an "entity": ACE 2005 emphasizes structural tracking by tagging pronouns (e.g., "his opponent") and broad nominals (e.g., "nearly everyone") for coreference. MultiNERD focuses on traditional, well-defined proper nouns mapped to fine-grained taxonomy. OntoNotes extends beyond standard names to heavily index nested entities and precise numerical or temporal values (e.g., "fifteen minutes", "9.80 %"). Finally, Pile-NER demonstrates its generative, open-domain capabilities by extracting highly contextual, non-traditional concepts (e.g., "creepy crawlers", "fun", "blood") alongside complex overlapping spans.

marks the full noun-phrase extent together with a head, and includes mention classes such as specific, generic, negative, and underspecified. As a result, ACE2005 contains many common-noun and pronoun mentions that would not be annotated in standard flat NER. MultiNERD is much closer to conventional named-entity tagging: it uses flat BIO spans and a fixed inventory of 15 fine-grained classes, and its annotations are derived from Wikipedia/Wikinews links plus exact-match/synonym propagation within the document. Compared with ACE2005, it is narrower in mention inclusion because it mainly targets linked surface mentions rather than nominal/pronominal references, but broader in semantic coverage because it adds categories such as ANIM, BIO, CEL, DIS, FOOD, MEDIA, MYTH, TIME, and VEHI. It also explicitly relaxes strict "entity" status for some classes, keeping concept-like items such as animals, plants, foods, and diseases when needed. PileNER differs most sharply from both: it does not use a fixed ontology at all, but an open-world setup in which GPT-3.5 was prompted to "*extract all entities and identify their entity types,*" producing 45,889 examples with about 240k spans and 13,020 distinct type names. We provide an overview of schemas as well a some sampled examples in Figure 15.