From Latents to Labels: Zero-Shot Named Entity Recognition using Sparse Autoencoder Features

Nakanyseth Vuth and Gilles Sérasset and Didier Schwab
Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG
38000 Grenoble
France
first.last@univ-grenoble-alpes.fr

Abstract

Zero-shot Named Entity Recognition is critical for low-resource domains, yet existing approaches rely on opaque prompting of large language models or dense representations that suffer from polysemanticity. We propose an alternative approach that leverages monosemantic features of Sparse Autoencoders. We introduce **SAE-NER**, a training-free framework that maps monosemantic SAE feature activations to entity types through direct precision estimation, requiring no supervision or prompting. Experiments across general and biomedical domains show that SAE-NER consistently outperforms trained probing classifiers, with especially a large margin in the biomedical domain (up to +20 F1). Finally, we evaluate the utility of SAE-NER predictions as silver training data for downstream NER models. Using controlled perturbations of gold annotations to simulate realistic annotation noise, we show that false negatives are the primary bottleneck for silver-data quality, outweighing the impact of boundary imprecision and false positives.

1 Introduction

Named Entity Recognition (NER) is a core task in Natural Language Processing and a prerequisite for information extraction. While supervised approaches achieve strong performance, they rely on large, high-quality annotated datasets, which are costly to obtain in specialized domains such as biomedical, law, or defense. This has driven interest in robust zero-shot methods that function without task-specific training data.

Currently, the most common paradigm for zero-shot NER is prompting-based inference, where Large Language Models (LLMs) (Brown et al., 2020; Touvron et al., 2023) are used to perform entity extraction via natural language instructions (Shen et al., 2023; Wang et al., 2025). While flexible, prompting remains computationally expensive, brittle to phrasing (Zhao et al., 2021), and often

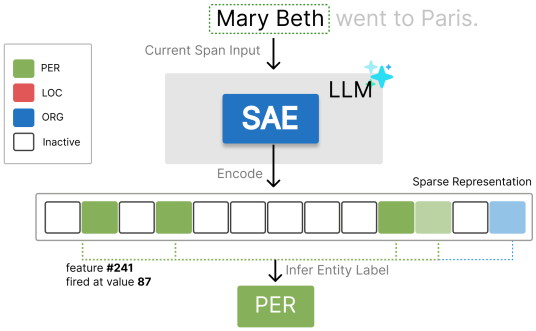


Figure 1: Visualization of the SAE-NER framework.

fails to provide the structured consistency required for reliable dataset construction. A parallel path involves extracting knowledge directly from the model's internal representations. Probing studies have confirmed that NER-relevant features are linearly decodable from hidden states (Hewitt and Manning, 2019; Tenney et al., 2019), and prototype-based methods have sought to use these representations for low-resource extraction (Snell et al., 2017; Ding et al., 2021). However, these approaches face a fundamental challenge: *polysemanticity*. Because individual neurons typically represent multiple unrelated concepts simultaneously, they cannot isolate entities without external supervision, such as labeled training data for probes or support sets of examples to define prototypes. Furthermore, these methods often rely on proxies like cosine similarity in a dense activation space, which fails to identify the specific semantic features driving the prediction.

In this work, we propose a different approach grounded in mechanistic interpretability. We hypothesize that the *monosemantic* nature of **Sparse Autoencoders (SAEs)** (Bricken et al., 2023; Cunningham et al., 2023) allows internal LLM knowledge to be leveraged directly for zero-shot NER. By decomposing dense activations into a high-dimensional space of isolated, interpretable con-

*Proceedings of the 15th Joint Conference on Lexical and Computational Semantics (*SEM 2026)*, pages 164–177

cepts, we can identify entities through specific feature activations rather than black-box generation or supervised classification. We introduce **SAE-NER** (Figure 1), a framework that reframes NER as a feature-detection problem, enabling training-free information extraction paradigm across different domains. Our contributions are:

- **SAE-NER Framework:** We propose a novel, training-free framework for zero-shot NER that leverages sparse SAE features for structured prediction, providing feature-level interpretability without the need for supervision or prompts.
- **Entity Typing Evaluation:** We compare SAE configurations across layers, types, and widths against supervised probes. We find that raw SAE features alone outperform trained classifiers in the general domain, and substantially surpass them in the biomedical domain.
- **We analyze data efficiency and noise sensitivity**, demonstrating the utility of SAE-NER for cold-start silver data generation and revealing that false negatives are the most detrimental form of noise in such datasets.

2 Preliminaries

2.1 Sparse Autoencoder

Deep neural network neurons are often *polysemantic*, they respond to a mixture of seemingly unrelated features. For instance, Bricken et al. (2023) identified neurons firing for different concepts like academic citations and Korean text. This stems from *superposition* (Elhage et al., 2022), a compression strategy where models represent more features than they have dimensions by encoding them as linear combinations in activation space. Sparse Autoencoders (SAEs) address this by decomposing these dense representations into interpretable, *monosemantic* features. By projecting activations into a higher-dimensional space with a sparsity constraint, SAEs isolate individual concepts. Formally, let $\mathbf{x} \in \mathbb{R}^d$ be an activation vector from a transformer layer. An SAE consists of an encoder and a decoder that map $\mathbf{x}$ to a sparse feature vector $\mathbf{f} \in \mathbb{R}^m$ and back to a reconstruction $\hat{\mathbf{x}} \in \mathbb{R}^d$, The process is defined as:

$$\mathbf{f}(x) := \sigma(W_{\text{enc}}\mathbf{x} + \mathbf{b}_{\text{enc}}) \quad (1)$$

$$\hat{\mathbf{x}} := W_{\text{dec}}\mathbf{f} + \mathbf{b}_{\text{dec}} \quad (2)$$

Here, $W_{\text{enc}} \in \mathbb{R}^{m \times d}$ and $W_{\text{dec}} \in \mathbb{R}^{d \times m}$ are the learned weight matrices, and $\mathbf{b}_{\text{enc}}, \mathbf{b}_{\text{dec}}$ are bias terms. The vector $\mathbf{f}$ represents the sparse feature activations, where each element f_i corresponds to the activation value of the i-th latent feature. The nonlinearity σ enforces sparsity; while the standard ReLU is common (Bricken et al., 2023; Cunningham et al., 2023), recent works have introduced TopK activation (Gao et al., 2024) and JumpReLU (Rajamanoharan et al., 2024) to improve feature interpretability and reconstruction fidelity.

2.2 Problem Formulation

Given an input document $I = [t_1, \ldots, t_n]$, the goal of NER is to identify a set of spans $X = \{x_1, \ldots, x_m\}$ and assign a corresponding label $y_k \in \mathcal{Y}$ to each span x_k, where $\mathcal{Y}$ is a pre-defined set of entity types. Each span $x_k = (i_k, j_k)$ corresponds to the token subsequence $[t_{i_k}, \ldots, t_{j_k}]$. Unlike traditional supervised approaches that train a dense classifier $P(y|x;\theta)$, we propose a zero-shot framework that exploits the learned representations of a pre-trained SAE. We denote an SAE trained on the activations of layer l with a dictionary size d as $\mathcal{M}^l_d$.[1]

3 The SAE-NER Framework

Our framework operates in two stages: (1) Candidate span generation, and (2) Zero-shot entity typing via sparse feature detection.

3.1 Candidate Span Generation

Since our method does not involve training a sequence labeling model, we require an unsupervised mechanism to propose candidate spans. We use two complementary strategies: (1) syntactic noun phrase extraction, where we parse the text with a constituency parser[2] to collect noun phrase spans as entity candidates, which we then merge with additional candidates identified via custom POS tag sequences; and (2) a class-agnostic mention detector[3] that proposes entity mentions and associated coreference clusters, capturing multi-token and referential spans that syntactic heuristics may miss.

[1] Dictionary size refers to the dimensionality of the sparse representation; different SAEs use different dictionary sizes.

[2] We use the Berkeley Neural Parser integrated with spaCy.

[3] We use the Mention Detector from https://github.com/SapienzaNLP/maverick-coref

3.2 Entity Typing

For each candidate span x_k, we determine its label y_k based on the pooled SAE activations of its constituent tokens. First, we obtain the sparse feature activations for the entire document:

$$[\mathbf{a}_1, \ldots, \mathbf{a}_n] = \mathbf{SAE}(I) \\ = [f(e_1), \ldots, f(e_n)] \quad (3)$$

where e_t is the contextual representation of token t from layer l of the LLM, and $\mathbf{a}_t \in \mathbb{R}^d$ is the resulting sparse activation vector. Each dimension of $\mathbf{a}_t$ corresponds to a specific latent feature from the SAE dictionary.

Feature Aggregation (Pooling) Since SAE activations are token-level, an aggregation strategy is necessary to obtain a fixed-size representation for a multi-token span. To prioritize tokens that are semantically "rich" (showing a strong signal from the autoencoder), we employ a norm-weighted pooling strategy. First, a scalar weight w_t is calculated for each token t in span x based on the L_2 norm of its feature vector $\mathbf{a}_t$:

$$w_t = \frac{\|\mathbf{a}_t\|_2}{\sum_{t' \in x} \|\mathbf{a}_{t'}\|_2} \quad (4)$$

The final pooled activation vector $\mathbf{v}_x$ for the span is the log weighted sum of the token vectors:

$$\mathbf{v}_x = \log\left(\sum_{t \in x} w_t \cdot \mathbf{a}_t\right) \quad (5)$$

where the $\log$ is applied element-wise to scale the value of extreme feature activations.

Feature to Label Mapping Before inferring entity labels, we must identify which SAE features serve as reliable detectors for our target entity types. We assume that an ideal SAE feature is monosemantic. Intuitively, if a feature f represents a concept like LOC, its activation should consistently activate for location entities, which act as a high-precision indicator for that class. Our objective is to discover these high-fidelity mappings by constructing an *Inference Matrix* $M \in \mathbb{R}^{|\mathcal{Y}| \times d}$, where each entry $M_{y,f}$ represents the empirical precision of feature f for label y. The mapping process consists of the following steps:

1. **Corpus Preparation:** Using a training set from a Gold dataset, we collect a set of positive spans X_P labeled with ground truth entity types. We also generate a set of negative "O" spans $X_O = \{(i,j) \mid (i,j) \notin X_P\}$ using the syntactic heuristic from Section 3.1. The size of the negative set is limited to $|X_O| = \min(\lambda|X_P|, |X_O|)$, where λ is a balancing coefficient.

2. **Span Activation Caching:** For every span $x \in X_P \cup X_O$, we compute the pooled activation vector $\mathbf{v}_x$. Let $\mathcal{F}_x$ be the set of indices corresponding to the top-k highest values in $\mathbf{v}_x$. For each feature $f \in F_x$, we store the activation-label pair $(v_x[f], l_x)$ in a cache C_f, where $l_x \in \mathcal{Y} \cup \{O\}$ is the ground-truth label of span x (set to O for non-entity spans in X_O). For instance, if Feature #42 appears in the top-k for several spans, its cache might look like: $\mathcal{C}_{42} = [(0.8, \text{ORG}), (0.85, \text{ORG}), (0.20, \text{LOC})]$. To reduce noise, we exclude common features that activate $> 2\%$ on the 10k-Pile dataset[4].

3. **Precision Estimation:** We proceed to analyze the features stored in the cache. For each feature f, we use the cached top-k values $\mathcal{C}_f$ to distinguish significant signals from noise. We set an activation threshold τ_f to the q-th percentile of the values in $\mathcal{C}_f$:

$$\tau_f = \text{Percentile}(\{v \mid (v,l) \in \mathcal{C}_f\}, q) \quad (6)$$

With this adaptive threshold, we calculate the precision $P(f,y)$ for each label $y \in \mathcal{Y}$. A prediction is considered a hit if the cached activation $v > \tau_f$. Thus, the precision is defined as:

$$P(f,y) = \frac{\sum_{(v,l) \in \mathcal{C}_f} \mathbf{1}\{v > \tau_f \wedge l = y\}}{\sum_{(v,l) \in \mathcal{C}_f} \mathbf{1}\{v > \tau_f\}}. \quad (7)$$

Note that the denominator counts all instances in the cache where the feature activation exceeded the threshold, regardless of the label.

4. **Matrix Construction:** We assign each feature f to the single label y^* that it predicts best, provided the precision exceeds a confidence threshold T_P:

$$y^* = \arg\max_{y \in \mathcal{Y} \cup \{\text{O}\}} P(f,y), \quad p^* = P(f,y^*) \quad (8)$$

If $p^* \geq T_P$, we set $M_{y^*,f} = p^*$; all other entries for feature f remain zero.

[4] https://huggingface.co/datasets/NeelNanda/pile-10k

Inference At test time, zero-shot classification is performed by aggregating the evidence from the active sparse features. Let $\mathbf{v}'_x \in \mathbb{R}^d$ denote the pooled activation vector for a test span x, filtered to keep only the top-k values. The evidence score S_y for a candidate label y is the dot product between the label's precision (row y of M) and the span's activations:

$$S_y = M_y \cdot \mathbf{v}'_x = \sum_{f=1}^{d} M_{y,f} \cdot v'_{x,f} \tag{9}$$

The scores for all labels are computed as $\mathbf{S} = M\mathbf{v}'_x$. The final prediction $\hat{y}$ follows a "confident argmax" logic. We first select the candidate label with the highest evidence score. However, to ensure reliability, we reject predictions that fail to meet a global confidence threshold T_{conf}, defaulting them to 'O'. Let $y^* = \underset{y \in \mathcal{Y} \cup \{\text{O}\}}{\text{argmax}}\ S_y$. The final prediction is:

$$\hat{y} = \begin{cases} y^* & \text{if } S_{y^*} \geq T_{conf} \\ \text{O} & \text{otherwise} \end{cases} \tag{10}$$

4 Experiments

We design our empirical evaluation to address two primary research questions: **RQ1 (Representation Quality):** Do the sparse features of an SAE provide a superior basis for zero-shot NER compared to the original dense representations? **RQ2 (Data Efficiency):** Can our framework generate silver-standard data to mitigate label scarcity?

4.1 Experimental Setup

SAE Models : We evaluate how size, architecture (Base vs. Instruction Tuned), and depth affect performance using Lieberum et al. (2024) SAEs trained on `Gemma-2-2B`, `9B`, and `9B-IT` activations.[5] For 9B models, we use layers $l \in \{9, 20, 31\}$ with dictionary widths $d \in \{16\text{k}, 131\text{k}\}$. For 2B, we use $l \in \{5, 12, 19\}$ and $d \in \{16\text{k}, 65\text{k}, 1\text{M}\}$. The layer choices correspond to early, middle, and late representations, allowing us to assess the effect of depth on entity recognition. The dictionary widths reflect the configurations released by Lieberum et al. (2024), as not all widths are available across model sizes.

Datasets : We evaluate on two datasets: Re-DocRED (Tan et al., 2022), a general-domain document-level relation extraction dataset with 6 coarse entity types, and AnEM (Ohta et al., 2012), a biomedical corpus with 11 fine-grained anatomical entity types. These datasets were selected to assess SAE-NER across two axes of variation: domain specificity (general vs. biomedical) and ontological granularity (coarse vs. fine-grained), which together probe the limits of SAE feature disentanglement.

4.2 RQ1: Representation Quality

4.2.1 Baselines and Metrics

We compare our approach against a supervised **Probing Classifier**: a 1-layer MLP (512 units, ReLU) trained on dense activations from middle (layers 10–20) and final layers. Results are averaged over 5 seeds. Performance is assessed across two levels: (1) **Entity Typing (Oracle Spans)** We classify ground-truth gold spans directly. This isolates the quality of SAE feature representations from span generation errors, providing a clean measure of whether sparse features align with entity type distinctions independently of the candidate generator. (2) **Full NER (Predicted Spans)** We classify candidate spans generated from our heuristics (Sec. 3.1). Since heuristic boundaries may not perfectly match gold annotations, we evaluate each span using different IoU thresholds.[6]

4.2.2 Results: Entity Typing

As shown in Figure 2, SAE-NER consistently outperforms probing baselines across both datasets in the majority of configurations, with a few exceptions at suboptimal layer and width combinations (layer 31 on Re-DocRED, where performance drops below the probing baseline), suggesting that layer choice is a critical factor. The most competitive configurations are concentrated in early-to-middle layers, particularly for the general domain.

Domain Robustness and Disentanglement: The performance gain is most significant in the biomedical domain (`AnEM`). Our best 9B configuration ($\mathcal{M}^{20}_{131k}$) reaches 0.623 F1, surpassing the 0.421 probing baseline. This suggests SAEs successfully disentangle specialized concepts, such as specific proteins or tissues, which otherwise remain hidden in polysemantic dense representations.

Layer and Model Analysis: Optimal layer depth varies across domains. For `Re-DocRED`, perfor-

[5] `https://huggingface.co/google/gemma-scope`

[6] IoU@θ denotes that a predicted span is considered a match if its Intersection over Union with the gold span exceeds θ, i.e., $\frac{|p \cap g|}{|p \cup g|} \geq \theta$.

mance peaks at early layers (L9), reaching 0.803 F1, as these layers retain surface-level lexical features and local syntactic patterns that are most favorable for identifying general entity types (Hewitt and Manning, 2019; Tenney et al., 2019). Conversely, the specialized biomedical entities in AnEM require more abstract semantic representations; consequently, performance in this domain benefits from the more refined features found in deeper layers and larger (9B) models. Interestingly, Instruction-Tuned (IT) models consistently outperform Base models on AnEM, particularly with larger width. We attribute this to *implicit supervision*: instruction tuning datasets often include extraction-style tasks or scientific QA, which forces the model to refine its internal representations of fine-grained entities (i.e., distinguishing "HEK293 cells" as a biological entity rather than a miscellaneous noun).

Further Analysis and Ablations: Extended analysis (Appendix A) reveals two further insights. First, correct labels often fall within the top 2-3 predictions, highlighting SAE-NER's utility for human-in-the-loop annotation. Second, targeted feature ablations support a *Distributed Representation Hypothesis*, confirming that entity knowledge is not centralized in a single "grandmother neuron" but is distributed across a set of specialized features.

4.2.3 Results: Full NER

Table 1 presents the performance of the full pipeline. The results characterize SAE-NER as a **high-precision, low-recall framework**. For instance, on Re-DocRED using the Mention Detector, we achieve a high Precision of 0.83 (IoU@0.3), confirming that when a candidate span is correctly proposed, our SAE-based typing mechanism is highly reliable. However, the overall F1 is heavily constrained by Recall ($\sim$0.56). We attribute this low recall primarily to a referential bias of the mention detector: our analysis reveals that the detector frequently fails to identify entities of type TIME and NUM, as these entity types are rarely referential and are thus systematically overlooked by the coreference-based mention detector.

Domain-Specific Challenges: Recall is most severely reduced in the biomedical domain, where overall recall is capped at $\sim$25%. This disparity highlights the limitations of domain-agnostic tools for specialized tasks; biomedical entities often exhibit complex syntactic structures that defy general-purpose POS patterns. However, the span generator is not the only bottleneck. For certain niche entity types in AnEM, the feature mapping process yielded an empty set. This suggests that general-purpose SAEs lack the specific features required to distinguish fine-grained biomedical concepts, resulting in zero recall for those classes. This highlights the need for SAEs trained on domain-relevant corpora.

Boundary Sensitivity and Semantic Mismatch: Evaluation across IoU thresholds reveals a trade-off between **Semantic Localization** and **Boundary Precision**. On AnEM, F1 drops from 0.32 (IoU@0.3) to 0.14 (IoU@0.7), indicating that while the framework correctly localizes the semantic concept, it struggles with exact gold-standard boundaries. We identify a recurring **Boundary-Semantics Mismatch**: for example, extracting "*Eminem*" instead of the gold span "*The Eminem Show*". Because the span isolates the artist's name from the full album title, the SAE types this span as PER, whereas the gold annotation labels the full span "*The Eminem Show*" as MISC. In such cases, the error stems from boundary imprecision that shifts the semantic interpretation, rather than a failure of the SAE to recognize the underlying concept. Appendix C provides more of these error examples.

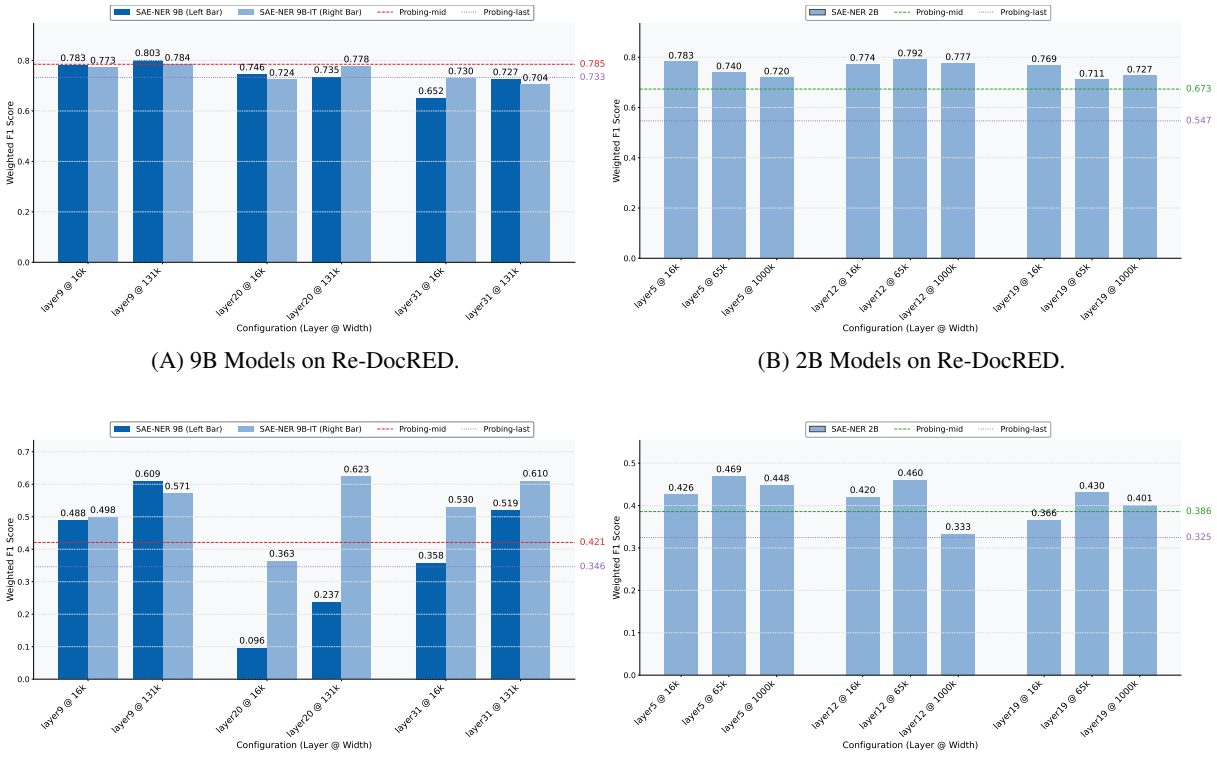
4.3 RQ2: Data Efficiency

4.3.1 Experimental Setup

Given the robust entity typing performance on the general domain in previous experiments, we focus our data efficiency analysis on the **Re-DocRED** dataset. We evaluate if SAE-NER predictions can serve as efficient silver training data for standard NER models.

Downstream Model: We fine-tune bert-base-uncased for 5 epochs (lr $= 2 \times 10^{-5}$, 5 random seeds) with a linear token classification head over the final hidden states, using the BIOES tagging scheme. We hypothesize that its explicit boundary tags (E and S) will help the student model refine the imprecise span boundaries inherent in our silver data.

Training Data Variants: To assess the quality of our generated data, we construct several training sets representing different data quality scenarios, including mixed configurations that combine silver and gold data to simulate realistic low-resource



(A) 9B Models on Re-DocRED.

(B) 2B Models on Re-DocRED.

(C) 9B Models on AnEM.

(D) 2B Models on AnEM.

Figure 2: Entity Typing F1 scores across model sizes, types, layers, and widths. Dashed lines represent supervised probing baselines.

Dataset	Span Generator	Metric	Precision	Recall	F1
Re-DocRED	Mention Detector	IoU@0.3	**0.8297**	**0.5589**	**0.6439**
		IoU@0.5	0.7720	0.5234	0.6016
		IoU@0.7	0.6454	0.4458	0.5092
		MUC	0.7078	0.5054	0.5743
	Syntactic Parser	IoU@0.3	0.7808	0.4710	0.5783
		IoU@0.5	0.6704	0.4047	0.4969
		IoU@0.7	0.5131	0.3123	0.3823
		MUC	0.6358	0.4319	0.5062
AnEM	Mention Detector	IoU@0.3	**0.5041**	**0.2492**	**0.3262**
		IoU@0.5	0.4166	0.2110	0.2742
		IoU@0.7	0.2122	0.1099	0.1414
		MUC	0.3927	0.2009	0.2609
	Syntactic Parser	IoU@0.3	0.4689	0.2548	0.3224
		IoU@0.5	0.3533	0.1959	0.2460
		IoU@0.7	0.1670	0.0995	0.1229
		MUC	0.3765	0.2154	0.2688

Table 1: Full NER performance on Re-DocRED and AnEM using IoU thresholds and MUC evaluation.

settings. (1) **Gold Standard ($\mathcal{D}_{Gold}$):** The original human-annotated set (upper bound). (2) **SAE-Generated Silver ($\mathcal{D}_{Silver}$):** Replacing gold annotations with zero-shot predictions from our best configuration (9B - $\mathcal{M}^{9}_{131k}$) on the unlabeled training corpus; spans are included if they exceed a confidence threshold. (3) **Data Scarcity Simulation ($\mathcal{D}_{Gold@d\%}$):** Subsampled documents ($d\%$) from $\mathcal{D}_{Gold}$ to simulate high-quality, low-resource scenarios. (4) **Label Noise Simulation:** To assess the impact of different noises, we artificially perturb each $\mathcal{D}_{Gold}$ sample by (i) randomly selecting $p\%$ of entities and relabeling them as O (*Low-Recall*), (ii) assigning random labels to $p\%$ of non-entity spans (*Low-Precision*). (5) **Boundary Noise ($\mathcal{D}_{Jitter}$):** We simulate our observed boundary mismatch in silver data by randomly shifting or contracting the start/end indices of $p\%$ of gold entities in each sam-

ple by one token.

For all perturbation experiments, we vary the noise rate $p \in \{10\%, \ldots, 70\%\}$ and evaluate on the Gold Test Set using strict **Exact Match Entity-level F1**.

4.3.2 Results

As shown in Table 3, silver-standard supervision (0.47 F1) lags significantly behind the gold baseline (0.82 F1). Silver data provides a clear benefit in cold-start scenarios: augmenting a minimal 10% gold split nearly doubles performance from 0.25 to 0.49 F1. However, this advantage vanishes as gold data increases; in the 30% split, silver augmentation causes a sharp decline from 0.75 to 0.60 F1.

Notably, filtering silver spans by confidence (0.45, 0.65) yields negligible improvements. While thresholding is often intended to improve dataset quality, our results suggest that any gains in label precision are offset by the resulting surge in false negatives. Beyond this, results from the perturbation experiments reveal some insights and structural flaws in using zero-shot data for training.

False Negative Noise Is Most Detrimental Our analysis reveals a clear disparity between False Positive (FP) and False Negative (FN) noise. As shown in Figure 3, the student model tolerates significant FP noise (blue line), maintaining 0.775 F1 even at 50% corruption. Conversely, FN noise causes rapid degradation once corruption exceeds 40% (red line). This is particularly critical as our silver dataset contains **45,917 false negatives** (Table 2), making FN noise the dominant source of error. In imbalanced NER settings, missing entity labels do not merely remove positive supervision; they actively reinforce the majority O class bias. We find many FNs stem from **Typing Contamination** during span generation. For example, extracting "*the cruiser Bogatyr*" instead of "*Bogatyr*" creates a PER FP and a MISC FN simultaneously. This suggests that the effectiveness of SAE-NER is tightly coupled to the quality of the candidate spans, and that errors in span generation can propagate directly into systematic FN noise during training.

Data Augmentation In the mixed-supervision setting shown in Table 3 (Silver + Gold@30%), adding silver data significantly degrades performance, with F1 dropping from 0.75 to 0.60. This stems from the non-uniform noise distribution across entity types (Figure 4): SAE-NER system-

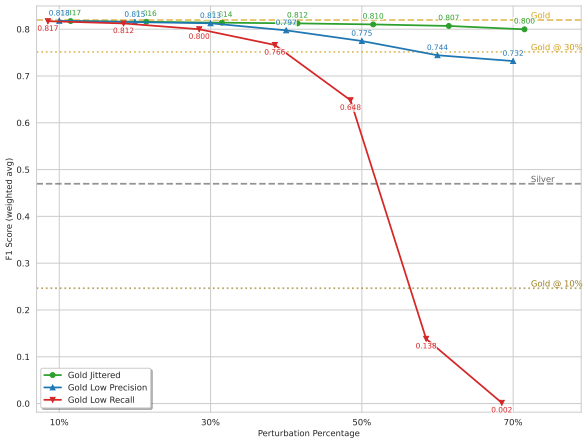


Figure 3: Effect of dataset perturbations on F1 score across increasing noise levels. Green, blue, and red curves denote $\mathcal{D}_{Jitter}$, $\mathcal{D}_{Low\text{-}Precision}$, and $\mathcal{D}_{Low\text{-}Recall}$, respectively.

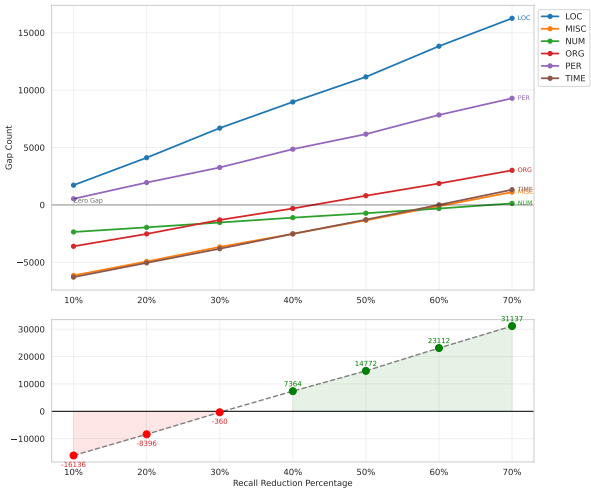


Figure 4: Entity count gap between Gold Low Recall datasets and the Silver dataset, reported at the entity-type level and in aggregate.

atically over-generates frequent categories (LOC, PER) but under-retrieves abstract or ambiguous types (TIME, MISC). By naively concatenating the full Silver dataset with Gold data, we introduce a severe **Distributional Skew**. The training signal becomes dominated by noisy, high-recall labels for entity types the model already predicts reliably, while providing little additional supervision for the weakest classes. These results suggest that effective silver-data usage requires **Stratified Augmentation**, selectively sampling silver data to target under-represented or difficult entity types rather than using uniform concatenation.

The Need for Noise-Aware Training Our results suggest that standard cross-entropy loss is subopti-

Dataset	Docs	Entities	$F1_{token}$	$F1_{entity}$	FP	FN	Density (Ents/Sent)
Gold	3049	78788	–	–	–	–	25.84
Silver	3049	53418	0.77	0.47	20547	45917	17.52
Gold Low Recall 50%	3049	38646	0.81	0.66	0	40142	12.67
Gold Low Precision 50%	3049	118930	0.88	0.81	40142	0	39.01
Gold Jittered 50%	3049	78744	0.90	0.56	34534	34578	25.83

Table 2: Dataset statistics for gold, silver, and perturbed ($p = 50\%$) datasets. Metrics (FP, FN, F1) are computed relative to original gold annotations. See Appendix D for extended statistics.

Training Configuration	Precision	Recall	F1 (weighted)
Silver-only supervision			
Silver	0.5675 ± 0.0043	0.4239 ± 0.0019	0.4697 ± 0.0019
Silver 0.65	0.5955 ± 0.0030	0.4107 ± 0.0027	0.4641 ± 0.0018
Silver 0.45	0.5733 ± 0.0032	0.4211 ± 0.0020	0.4685 ± 0.0016
Gold-only supervision			
Gold	0.8348 ± 0.0017	0.8056 ± 0.0015	0.8196 ± 0.0015
Gold @ 10%	0.5057 ± 0.1194	0.2743 ± 0.0216	0.2465 ± 0.0280
Gold @ 30%	0.8050 ± 0.0035	0.7136 ± 0.0082	0.7515 ± 0.0061
Mixed silver and gold supervision			
Silver + Gold (10%)	0.6245 ± 0.0043	0.4274 ± 0.0025	0.4865 ± 0.0014
Silver + Gold (30%)	0.7263 ± 0.0049	0.5262 ± 0.0064	0.5952 ± 0.0048
Zero-shot			
SAE-NER	0.6174 ± 0.0000	0.3999 ± 0.0000	0.4666 ± 0.0000

Table 3: NER results for various supervision settings. *Silver* denotes unfiltered predictions, while 0.45 and 0.65 indicate confidence thresholds. *Zero-shot* row reports direct inference on the test set without downstream fine-tuning, serving as a reference point for the quality ceiling of the silver data.

mal for silver-standard supervision as it treats all labels as ground truth. Because silver data contains frequent false negatives, the cross-entropy objective explicitly penalizes the model for correctly identifying these spans, eroding previously learned representations. This is particularly detrimental under mixed supervision, where noisy silver gradients can counteract gold signals. Notably, the student model's robustness to boundary "Jitter" (Figure 3 - green line) indicates that the BERT encoder can tolerate noisy spans, suggesting the performance bottleneck is not representational, but a consequence of the loss function's inability to account for label noise. These findings motivate the adoption of **noise-aware training** strategies (Mayhew et al., 2019; Li et al., 2020) that explicitly model label uncertainty rather than treating silver and gold annotations as equally reliable.

5 Related Works

The dominant paradigm for zero-shot NER relies on prompting-based inference with LLMs (Brown et al., 2020; Touvron et al., 2023). By formulating extraction tasks as natural language instructions, these methods exploit pre-trained world knowledge without task-specific tuning. While flexible and scalable, prompting suffers from inherent reliability issues: performance is notoriously sensitive to prompt phrasing (Zhao et al., 2021), models frequently hallucinate or fail to adhere to output schemas (Ji et al., 2022), and the inference costs of multi-billion parameter models are prohibitive for large-scale data generation. Furthermore, the black-box nature of generation offers no interpretability regarding why a specific entity was extracted. A parallel line of research focuses on accessing the model's internal knowledge directly. Prototype-based methods approach zero-shot NER via metric learning, mapping span embeddings to entity-type descriptions (Snell et al., 2017; Ding et al., 2021). Closely related are probing classifiers (Hewitt and Manning, 2019; Tenney et al., 2019), which train linear models on frozen activations to diagnose the presence of linguistic features, including named entities.

However, these internal methods face a fundamental bottleneck: they rely on *dense* representations. Due to superposition, these embeddings are highly polysemantic (Elhage et al., 2022), forcing methods to rely on indirect similarity metrics

or trained classifiers to untangle the signal. Thus, probing remains primarily diagnostic: it confirms that entity information is linearly decodable from dense representations, but the probe itself is an external classifier trained to decode the signal; it does not expose the underlying features responsible for the prediction, nor does it provide a direct, interpretable mechanism for entity extraction in the way SAE features do.

6 Conclusion

In this work, we introduced SAE-NER, a training-free framework that leverages the monosemantic features of Sparse Autoencoders for zero-shot NER. Our empirical evaluation supports two key conclusions. First, we demonstrate that sparse features provide a functionally superior signal for identifying entity types compared to dense representations, particularly in specialized domains like biomedical. Unlike probing classifiers, which require supervised training to disentangle polysemantic dense representations, SAE-NER directly aggregates evidence from monosemantic sparse features, providing interpretable, training-free entity detection given a pre-trained SAE. Second, we establish the practical utility of this framework for generating silver training data in cold-start scenarios. However, we also identify a critical recall bottleneck driven by heuristic span generation, which limits the utility of the silver data in high-resource settings. We conclude that while SAEs provide the semantic precision necessary for structured extraction, future success relies on coupling them with high-recall candidate generators and noise-aware training paradigms.

Limitations

While our results demonstrate the potential of SAE-NER, several limitations remain that point towards important paths for future work:

Model and Language Scope: Our study is restricted to the `Gemma-2` SAE family and English-language corpora. Future work is required to verify if these findings generalize to other architectures and multilingual settings, which is particularly critical for low-resource languages.

Inference Mapping Dependency: Our current method relies on gold annotations to map SAE features to specific entity labels. Future research should investigate automated mapping via distant supervision on target-domain text. By using external knowledge bases to label raw corpora and caching the resulting feature activations, mappings could be established without any manual gold data.

Span Generation Bottleneck: Our heuristic-based candidate generator lacks sufficient recall for diverse entity types and struggles with the syntactic complexity of specialized domains. Future work should explore more sophisticated, domain-specific generation strategies to improve coverage.

Noise Modeling Depth: Our jitter analysis was limited to single-token shifts. Silver annotation noise is more complex; a granular study considering multi-token shifts and varied degradation rates is required to fully characterize model robustness.

Evaluation Protocol: Our evaluation protocol may be overly conservative. Because we rely on exact-match F1, semantically plausible predictions that suffer from boundary imprecision, such as identifying *Eminem*" as PER instead of the gold *The Eminem Show*" (MISC) are penalized as double errors. This suggests that the framework's actual utility for NER may be higher than our metrics imply.

Acknowledgements

This work was supported by the RIPOSTE project of AID, the CHIST-ERA grant CHIST-ERA-25-SOL-02 (CLASiK project) funded by the French Agence Nationale de la Recherche (ANR) under grant ANR-25-CHR4-0004, and COST Action CA23147 GOBLIN - Global Network on Large-Scale, Cross-domain and Multilingual Open Knowledge Graphs, supported by COST (European Cooperation in Science and Technology, https://www.cost.eu).

References

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Ma teusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *ArXiv*, abs/2005.14165.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2023. Sparse autoencoders find highly interpretable features in language models. *ArXiv*, abs/2309.08600.

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. 2021. Few-NERD: A few-shot named entity recognition dataset. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3198–3213, Online. Association for Computational Linguistics.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. 2022. Toy models of superposition. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2022/toy_model/index.html.

Leo Gao, Tom Dupr'e la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. 2024. Scaling and evaluating sparse autoencoders. *ArXiv*, abs/2406.04093.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *North American Chapter of the Association for Computational Linguistics*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Delong Chen, Wenliang Dai, Andrea Madotto, and Pascale Fung. 2022. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55:1 – 38.

Yangming Li, Lemao Liu, and Shuming Shi. 2020. Empirical analysis of unlabeled entity problem in named entity recognition. *ArXiv*, abs/2012.05426.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, J'anos Kram'ar, Anca Dragan, Rohin Shah, and Neel Nanda. 2024. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. *ArXiv*, abs/2408.05147.

Stephen Mayhew, Snigdha Chaturvedi, Chen-Tse Tsai, and Dan Roth. 2019. Named entity recognition with partially annotated training data. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 645–655, Hong Kong, China. Association for Computational Linguistics.

Tomoko Ohta, Sampo Pyysalo, Jun'ichi Tsujii, and Sophia Ananiadou. 2012. Open-domain anatomical entity mention detection. In *Proceedings of the Workshop on Detecting Structure in Scholarly Discourse*, pages 27–36, Jeju Island, Korea. Association for Computational Linguistics.

Senthooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, János Kramár, and Neel Nanda. 2024. Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders. *ArXiv*, abs/2407.14435.

Yongliang Shen, Zeqi Tan, Shuhui Wu, Wenqi Zhang, Rongsheng Zhang, Yadong Xi, Weiming Lu, and Yueting Zhuang. 2023. PromptNER: Prompt locating and typing for named entity recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12492–12507, Toronto, Canada. Association for Computational Linguistics.

Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical networks for few-shot learning. In *Neural Information Processing Systems*.

Qingyu Tan, Lu Xu, Lidong Bing, Hwee Tou Ng, and Sharifah Mahani Aljunied. 2022. Revisiting docred – addressing the false negative problem in relation extraction. In *Proceedings of EMNLP*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. Bert rediscovers the classical nlp pipeline. In *Annual Meeting of the Association for Computational Linguistics*.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Niko lay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melissa Hall Melanie Kambadur, Sharan Narang, Aur'elien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, Guoyin Wang, and Chen Guo. 2025. GPT-NER: Named entity recognition via large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4257–4275, Albuquerque, New Mexico. Association for Computational Linguistics.

Tony Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*.

A Entity Typing Analysis

Top-K Accuracy Analysis To diagnose whether errors stem from a lack of knowledge or a failure in calibration, we analyze Top-k accuracy (Figure 5). We observe a significant performance-knowledge gap, most notably in the `AnEM-9B-L20` configuration. Despite a collapsed Top-1 Accuracy of 0.06, the Top-3 Accuracy recovers dramatically to 0.47. This indicates that the SAE features successfully map the span to the correct semantic neighborhood, but the model lacks the discriminative knowledge to disambiguate between closely related fine-grained types. **Practical Implication:** This finding highlights the potential of SAE-NER for **human-in-the-loop annotation**. While the zero-shot Top-1 prediction may be noisy in specialized domains, the high Top-k accuracy suggests that the correct label is frequently contained within the top few candidates. Consequently, the framework can serve as an effective annotation assistant, presenting a short list of high-probability labels to human experts, and therefore accelerating the creation of gold-standard data in new domains.

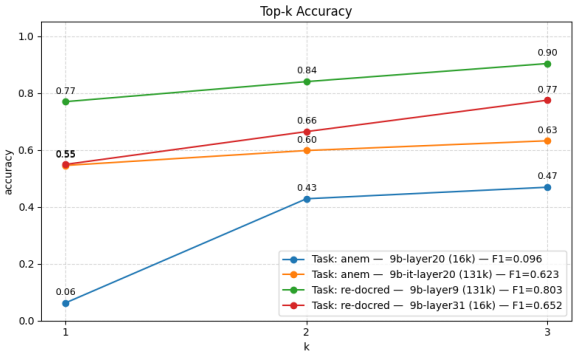


Figure 5: Top-k accuracy for different configurations on Entity Typing.

Ablation Study To verify that our method relies on causal mechanisms rather than spurious correlations, we perform a targeted ablation study on Re-DocRED ($\mathcal{M}^{9}_{131k}$). For each entity type, we zero out the top 20% and 50% of its mapped features and measure the impact. We also perform a **random control ablation**, where we randomly remove 20% of the mapped features.

As shown in Figure 6, the results reveal a diagonal pattern: removing features for a specific class causes a sharp drop for that class, i.e., -0.58 for TIME in the 50% condition, while leaving other entity types virtually unaffected. **In contrast, the random control ablation resulted in negligible performance changes (< 0.005 drop in F1).** This stark contrast confirms that SAE-NER successfully isolates the specific, causal mechanisms responsible for entity recognition. Furthermore, the fact that ablating 20% of features removes roughly half the performance of ablating 50% supports a **Distributed Representation Hypothesis**: entity knowledge is not centralized in a single "grandmother neuron" but is distributed across a set of specialized features. Interestingly, we observe minor performance gains for competing classes during targeted ablation; ablating LOC improves ORG by +3.3 points. This suggests a **competitive inference process**: by suppressing the signal for one class, we reduce confusion in ambiguous contexts, allowing the correct alternative signal to dominate.

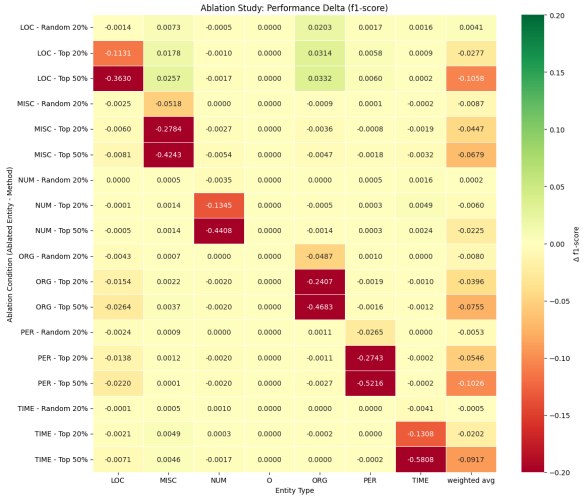


Figure 6: Feature Ablation results on Entity Typing.

B Full NER Performance

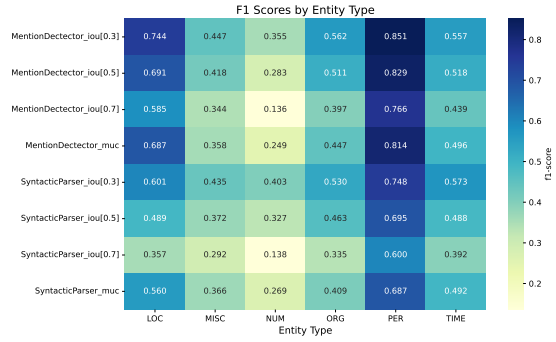


Figure 7: Full NER performance on Re-DocRED reported by class.

C Error Examples

Span	Gold	Pred	$\mathbf{p}_{\text{pred}}$	$\mathbf{p}_{\text{gold}}$	$\mathbf{p}_{\text{2nd}}$	Comment
Cy Becker	LOC	PER	0.995	0.001	0.001	High-conf. pred.
communications-based train control	MISC	O	0.989	0.002	0.002	High-conf. pred.
2016 Summer Olympics	MISC	TIME	0.871	0.102	0.102	TP in 2nd pred.
Casanova	PER	LOC	0.772	0.134	0.134	TP in 2nd pred.
La Brea Tar Pits	LOC	MISC	0.410	0.358	0.358	Low-conf. pred., TP in 2nd pred.
Atari	ORG	MISC	0.483	0.478	0.478	Low-conf. Pred., TP in 2nd pred.

Table 4: False positive cases on Entity Typing; p_{2nd} denotes the confidence of the second-ranked prediction on Re-DocRED. For many test spans, the right prediction can be found in the second prediction.

Span	Gold	Pred	$\mathbf{p}_{\text{pred}}$	$\mathbf{p}_{\text{gold}}$	$\mathbf{p}_{\text{2nd}}$	Comment
GB	Organ	O	0.083	0.083	0.083	...
NCC - derived tissues	Tissue	Cell	0.662	0.042	0.277	...
Be2C cells	Cell	O	1.000	0.000	0.000	...
upper GI	Organism Subdivision	O	0.083	0.083	0.083	...

Table 5: False positive cases on Entity Typing; p_{2nd} denotes the confidence of the second-ranked prediction on AnEM. We found that a lot of entity spans are predicted as O.

Pred. Span	Gold Span	Label_{Pred}	Label_{Pred}	p	IoU	Comment
the cruiser Bogatyr	Bogatyr	PER	MISC	0.85	0.60	Typing contamination
Rafael Correa 's PAIS Alliance	PAIS Alliance	PER	ORG	0.61	0.38	Typing contamination
Eminem	The Eminem Show	PER	MISC	1.00	0.33	Typing contamination
Kitzmiller	Kitzmiller v. Dover Area School District	PER	LOC	0.90	0.25	Typing contamination
Lucy	Jimmy White / Lucy	PER	MISC	1.00	0.25	Typing contamination
Marilyn Manson	Marilyn Manson	PER	ORG	0.42	1.00	Obvious wrong typing
Ernie Pyle Theatre	Ernie Pyle Theatre	PER	LOC	1.00	1.00	Obvious wrong typing
Mathematics	Steklov Institute of Mathematics	ORG	ORG	0.83	0.17	Not enough intersection

Table 6: Error Analysis on full NER. Results indicate that while entity typing is strong, the full NER pipeline is highly sensitive to the candidate span generator.

D Dataset Statistics

Entity Type	Gold	Gold @ 10%	Gold @ 30%
LOC	24,238	2,457	7,150
MISC	12,151	1,110	3,484
NUM	4,126	333	1,171
ORG	11,117	1,101	3,260
PER	14,507	1,454	4,321
TIME	12,649	1,250	3,639

Table 7: Entity count statistics for the full Gold dataset and its 10% and 30% subsets.

Entity Type	Silver	Silver 0.45	Silver 0.65	Silver + Gold @ 10%	Silver + Gold @ 10%
LOC	23,108	22,678	21,576	21,864	22,472
MISC	4,601	4,332	3,665	4,427	6,094
NUM	1,311	1,245	1,073	1,331	1,946
ORG	6,191	5,879	5,043	5,634	6,769
PER	13,312	13,006	12,390	12,580	12,995
TIME	4,895	4,845	4,717	5,531	7,001

Table 8: Entity count statistics for the Silver datasets and the mixed variation with Gold datasets.

Gold Jitter Perturbation							
Entity	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**
LOC	24,230	24,227	24,230	24,223	24,221	24,223	24,216
MISC	12,151	12,149	12,147	12,150	12,149	12,147	12,147
NUM	4,120	4,117	4,116	4,107	4,106	4,109	4,098
ORG	11,117	11,115	11,116	11,117	11,113	11,113	11,116
PER	14,507	14,507	14,507	14,507	14,506	14,505	14,507
TIME	12,649	12,648	12,649	12,649	12,649	12,649	12,648

Table 9: Entity counts under Gold Jitter perturbations.

Gold Low-Precision Perturbation							
Entity	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**
LOC	25,749	27,009	28,334	29,599	30,849	32,163	33,547
MISC	13,669	14,944	16,298	17,564	18,689	20,174	21,447
NUM	5,741	6,970	8,386	9,635	10,983	12,306	13,623
ORG	12,654	13,986	15,316	16,531	17,735	19,200	20,427
PER	16,008	17,344	18,634	19,929	21,112	22,603	24,005
TIME	14,201	15,509	16,830	18,264	19,562	20,824	22,246

Table 10: Entity counts under Gold Low-Precision perturbations.

Gold Low-Recall Perturbation							
Entity	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**
LOC	21,387	18,999	16,415	14,139	11,956	9,281	6,849
MISC	10,746	9,529	8,259	7,121	5,942	4,711	3,484
NUM	3,667	3,266	2,843	2,421	2,034	1,625	1,179
ORG	9,794	8,719	7,500	6,501	5,389	4,326	3,178
PER	12,777	11,364	10,051	8,454	7,145	5,476	4,021
TIME	11,183	9,941	8,710	7,418	6,180	4,887	3,570

Table 11: Entity counts under Gold Low-Recall perturbations.