# OWNER - Towards Unsupervised Open-World Named Entity Recognition

Pierre-Yves Genest, Pierre-Edouard Portier, Előd Egyed-Zsigmond, Martino Lovisetto

# OWNER — Towards Unsupervised Open-World Named Entity Recognition

Pierre-Yves Genest[1, 2], Pierre-Edouard Portier[3], Előd Egyed-Zsigmond[2], and Martino Lovisetto[1]

[1]Alteca, 69100 Villeurbanne, France
[2]INSA Lyon, CNRS, Universite Claude Bernard Lyon 1, LIRIS, UMR5205, 69621 Villeurbanne, France
[3]Caisse d'Epargne Rhône Alpes, 69003 Lyon, France

October 14, 2024

## Abstract

Unsupervised and zero-shot Named Entity Recognition (NER) aims to extract and classify entities in documents from a target domain without annotated data. This setting is particularly relevant for specific domains (biomedical, legal, scientific, ...) where labeled documents are scarce and expensive to create. While zero-shot NER approaches yield impressive outcomes, they operate under the assumption that all entity types are predefined and known. This limitation makes their application impossible in novelty detection, exploration, or knowledge graph construction scenarios.

To address this shortcoming, we introduce OWNER, our unsupervised and open-world NER model, which does not need annotations in the target domain (similar to zero-shot) and does not require knowledge of the target entity types or their number. We propose a novel triangular architecture to type and structure entities automatically. It comprises a prompt-based entity type encoder, an unsupervised clustering model, and embedding refinement with contrastive learning to refine entity embeddings and elicit entity types more precisely. Results on 13 domain-specific datasets show that OWNER outperforms open-world large language model prompting (4 % – 18 % in AMI) and performs competitively with state-of-the-art zero-shot models. Qualitative analysis shows that OWNER effectively groups entities into semantically meaningful clusters that closely resemble actual entity types (without knowing them beforehand). The source code of OWNER is publicly available at `https://github.com/alteca/OWNER`.

***Keywords*—** Named entity recognition, open information extraction, open-world named entity recognition, unsupervised named entity recognition.

## 1 Introduction

Named Entity Recognition (NER) is a fundamental NLP task that aims to identify entities in text and classify them into entity types. Historically, NER has primarily been approached as a supervised task, [74, 91], which presents challenges in specific domains (e.g., scientific, biomedical) where large labeled corpora may not be readily available. As a consequence, interest in low-resource and few-shot NER has risen [45, 78, 23], especially since the emergence of Encoder-only Language Models (EncLM) such as BERT [16]. However, these approaches still require annotated documents.

Zero-shot NER aims to alleviate this constraint. Recent models typically transfer knowledge from a source domain $\mathcal{D}_S$ to a target domain $\mathcal{D}_T$ where no annotated data is available [57, 92]. Although they do not require labels in $\mathcal{D}_T$, they assume a closed-world hypothesis, where entity types are known in advance. This is particularly problematic for tasks such as knowledge graph construction that require novelty detection. On a specific domain, an expert user can define the structure of information (entity types) he is interested in extracting. However, being exhaustive in this scheme identification and structuration process is difficult, if not impossible. This incompleteness would result in missing meaningful and valuable entities not initially envisioned.

The solution to these issues — 1) lack of annotations in specific domains, and 2) closed-world — is unsupervised and open-world[1] NER. Our literature review shows, however, that this setting has been little studied compared to closed-world NER. To the best of our knowledge, the last research work dates back to 2020 [48] and is not reproducible (lack of source code and implementation details). Therefore, this article aims to re-explore this setting in light of the recent advancements in NLP. We present **OWNER**, our "Unsupervised **O**pen-**W**orld **N**amed **E**ntity **R**ecognition" model. OWNER is unsupervised and open-world: it infers and structures entities into non-predefined entity types. OWNER uses annotated data from $\mathcal{D}_S$, which can be manually or automatically annotated documents, to

[1]Unsupervision implies an open-world setting as no prior knowledge (including entity types) is given to the model. In this article, when we employ the adjective "unsupervised," we implicate "open-world" as well.

learn named entity recognition and transfer it to $\mathcal{D}_T$, where no annotated document is available. We split NER into two subtasks: mention detection (locating entities) and entity typing (typing the extracted entities). For mention detection, we implement a simple BIO sequence labeling NER (see Sect. 2), expecting it to better generalize on specific domains than more complex architectures [23][2]. For entity typing, we employ EncLM (e.g., BERT [16]) prompting and clustering, which allow us to organize unseen entity types. In particular, we propose a heuristic to fasten the estimation of the number of clusters. We also implement an embedding refinement approach based on contrastive learning to isolate entity types more effectively in $\mathcal{D}_T$. This simple yet innovative architecture empirically outperforms LLM-based (Large Language Model) open-world NER and is competitive with closed-world zero-shot models. We expect it to be a strong benchmark for future unsupervised and open-world NER research. To summarize our main contributions:

- We propose OWNER, an unsupervised and open-world NER model that extracts and classifies entities from a target domain $\mathcal{D}_T$ 1) without annotations in $\mathcal{D}_T$, 2) without knowing the target entity types $\mathcal{T}_T$, nor 3) their number $|\mathcal{T}_T|$.
- To type entities, we propose a novel architecture with 1) prompt-based entity encoding, 2) unsupervised clustering to classify entities into entity types, and 3) contrastive learning to elicit entity types more precisely.
- Experimental results on 13 domain-specific datasets show that OWNER surpasses LLM-based open-world NER and performs comparably to state-of-the-art closed-world zero-shot models.
- Qualitative analysis highlights that OWNER structures entities in semantically coherent clusters close to true entity types.

# 2 Related Work

Before starting this section, we clarify the mathematical notations. NER analyses documents $\boldsymbol{X} = [x_0, x_1, ...x_{|\boldsymbol{X}|-1}]$. $x_i \in \boldsymbol{X}$ is a token[3]. The objective is to extract entities $\boldsymbol{e} = [x_i, ..., x_j]$ and classify their entity type $t$. The set of entity types is denoted $\mathcal{T}$. We assume access to labeled documents from a source domain $\mathcal{D}_S$ (with its set of entity types $\mathcal{T}_S$) and try to generalize to a target domain $\mathcal{D}_T$ (associated with the entity types $\mathcal{T}_T$) where annotated data is absent. Closed-world models need to know $\mathcal{T}_T$ (number, names, and sometimes descriptions), whereas open-world methods such as OWNER cannot access it.

## 2.1 Few-Shot & Zero-Shot NER (Closed-World)

As a reminder, few-shot and zero-shot models suppose knowing the target entity types list $\mathcal{T}_T$ beforehand. Most approaches assume the availability of labeled data in a source domain $\mathcal{D}_S$ and try to learn from $\mathcal{D}_S$ and transfer to $\mathcal{D}_T$. $\mathcal{D}_S$ and $\mathcal{D}_T$ differ stylistically (type of text), semantically (topic), or from the entity-type perspective ($\mathcal{T}_S \neq \mathcal{T}_T$). $\mathcal{D}_S$ can be a manually annotated dataset [23, 64], a distantly labeled dataset [68], or a synthetically generated dataset [55, 92, 86]. Recent approaches are divided into two families: 1) two-stage NER, and 2) one-stage or integrated NER.

Two-stage approaches split NER into Mention Detection (MD) and Entity Typing (ET) [88, 23, 49]. MD aims to identify spans of $\boldsymbol{X}$ that are entities, and ET classifies the type of each extracted entity. Integrated models combine MD and ET in one step, the motivation being to reduce cascading errors [14, 68, 31, 92, 64]. In practice, both paradigms attain state-of-the-art results [23, 14]. Until recently, most approaches relied on Encoder-only Language Models (EncLM), such as BERT [16]. We see now the rising use of Large Language Models (LLM) in these two low-resource settings [92, 64], where LLMs are shown to shine [3].

**Mention Detection (MD)** Few-shot and zero-shot approaches follow architectures similar to supervised models for MD. They usually implement span-based extractors [78, 20, 86], although BIO sequence labeling is still used [23, 49]. These extractors are trained in a supervised fashion on $\mathcal{D}_S$ entities. The challenge involves transferring the learned patterns from $\mathcal{D}_S$ to $\mathcal{D}_T$ entities. BIO sequence labeling classifies each token $x$ in $\boldsymbol{X}$ as either $B$ (first token of an entity), $I$ (second or following token of an entity), or $O$ (not an entity). A decoding algorithm then reconstructs the entity's boundaries using the predicted classes. Greedy algorithms are used, especially with recent language models [23]. Conditional random fields [25] are extensively employed to improve decoding. The main weakness of BIO is that it cannot predict nested entities. This is the main motivation for span-based extractors.

In general, span-based extractors score each possible span in $\boldsymbol{X}$ and determine the true entities [91, 78]. To do that, they compute *start of span* and *end of span* vector representations (usually embeddings of the first and last tokens of the candidate span). Zhong et al. [91] concatenate the *start* and *end* embeddings and use them in a perceptron that scores the candidate span. Wang et al. [78] use bilinear layers to replace the perceptron, allowing more efficient computations compared to Zhong et al. [91]. Span-based approaches suffer from the quadratic number of possible spans, making scoring the candidate spans expensive for long documents. Dobrovolskii [19] tries to overcome this problem with a hybrid approach. First, each word in $\boldsymbol{X}$ is classified as an entity head or not. An entity head is the main word of an entity; Dobrovolskii considers the head to be the root of the syntactic subtree of the entity. This ingenuity allows him to lower the quadratic span complexity to a linear (word) complexity. Once the entity heads are identified, the boundaries of each entity are determined using a convolutional neural network. Finally, Zaratiana et al. [85] propose to adapt conditional random fields for span-based extractors to enforce non-overlapping spans.

**Entity Typing (ET)** The general principle is to compute a vector representation of the extracted entities (entity embeddings) and compare them to those of the exemplars (few-shot) or the target entity types (zero-shot and few-shot). Zhang et al. [88] propose to use the k-nearest neighbors with the few-shot exemplars to identify the type. Prototypical networks [70]

[2] Experimental results confirm this hypothesis (see Sect. 5.3).

[3] Word, part of a word, or punctuation as defined by SentencePiece [39].

are generally preferred to classify entities [78, 23, 20]. The entity-type prototypes are computed using the exemplars.

Entity embeddings are computed by aggregating the EncLM embeddings of the individual tokens composing the entity in the case of a BIO extractor [23] or by using the span representation constructed by the span extractor [78]. Shen et al. [68] and Ding et al. [17] explore prompting techniques with EncLM (using the `[MASK]` token) to generate entity embeddings.

Meta-learning [24] is employed to enhance the efficacy of transfer learning [49]. The idea is to generate large amounts of few-shot episodes using the annotated data of $\mathcal{D}_S$; each episode contains a subset of $\mathcal{T}_S$, randomly selected few-shot exemplars associated to $\mathcal{T}_S$, and test documents to compute the performance. Then, the model is trained on the episodes to achieve the best transfer in the smallest fine-tuning steps possible (hence the meta-learning term). This allows fine-tuning even on the limited few-shot exemplars, as the model is adapted to converge quickly and reliably.

Finally, Liu et al.[45] and Mahapatra et al. [51] explore the effectiveness of adapting EncLM embeddings to the target domain. They employ large amounts of unannotated documents of $\mathcal{D}_T$ and fine-tune BERT weights using a masked language modeling task. Empirically, they observe a link between a decrease in perplexity and an increase in NER performances. Mahapatra et al. [51] decrease the training time required for domain adaptation by filtering the unannotated documents of $\mathcal{D}_T$ to keep those more aligned to the actual documents where entities are to be extracted.

**Large Language Models** Recently, LLMs [81, 75] have been successfully applied to few-shot and zero-shot NER and have state-of-the-art results on the zero-shot setting.

First, "raw" prompting obtains impressive results compared to previous works [79, 84, 82]. Wang et al. [79] and Ye et al. [84] require few-shot exemplars to specify the output format. Wei et al. [82] (ChatIE) propose a multi-turn framework that works in a zero-shot setting (without the need for exemplars). Surprisingly, they reverse the usual MD and ET steps order. Indeed, they first ask the LLM which entity types are present in the document (given a predefined list of entity types). In subsequent turns, they ask the LLM about the entities associated with each entity type. Xie et al. [83] propose to generate few-shot instances using GPT-3.5 [54] automatically and refine them with an ensemble method (multiple generations with temperature and a voting system to gather entity predictions). They empirically observe that these automatically generated few-shot instances significantly improve zero-shot performance. The weakness of their works [83, 82] is the multiple turns required to analyze a document, which are expensive and slow when using the APIs of the largest LLMs.

Sainz et al. [64], Zhou et al. [92], and Wang et al. [80] explore the idea of fine-tuning small LLMs [75, 6, 35] on manually or synthetically labeled datasets. In doing so, they create NER-specialized LLMs with better performances than generalist LLMs while being much smaller. Zhou et al. [92] propose to annotate documents from the Pile corpus [26] using GPT-3.5 (they call this dataset Pile-NER) and fine-tune Vicuna [6] on it. Their UniNER model achieves better performances than GPT-3.5 in a zero-shot context. Additionally, fine-tuning using a large amount of synthetic data allows them to specify a custom JSON format that UniNER follows reliably. GoLLIE [64] uses Code-Llama [63] as its backbone and is fine-tuned on manually labeled datasets from the news and biomedical domains. Sainz et al. [64] and Li et al. [41] follow a "Python class" scheme, where each entity type is specified as a Python class with a name, a description, and a few examples. They find empirically that description and exemplars metadata positively impact GoLLIE and KnowCoder performances.

Regarding the prediction format, most of the approaches follow a surface form extraction scheme [84, 82, 92, 64], except GPT-NER [79]. The models output only the entity text, and a subsequent algorithm is required to localize the entity in the document. The output format is generally JSON, but Sainz et al. [64] use Python code, allowing them to add metadata elegantly in comments (description and exemplars). GPT-NER [79] proposes a sequence labeling scheme. It asks the LLM to repeat the input document, with a special markup delimiting the boundaries of entities: `@@` as the opening tag and `##` as the closing tag. This format removes the dependency on a decoding algorithm, as the detected entities are localized in the document by design. However, it is incompatible with a zero-shot setting, as in-context exemplars are required to describe the output format.

Finally, Zaratiana et al. [86] (GliNER), and Ding et al. [18] (GNER) fine-tune EncLM embeddings (DeBERTa v3 [30]) or full transformers (Flan-T5 [8]) on the GPT-3.5 generated annotations of Pile-NER [92]. In particular, GliNER implements a span-based extractor for MD coupled with a method similar to prototypical networks for ET. They obtain very competitive results compared to the much larger fine-tuned LLMs UniNER [92] and GoLLIE [64]. Today, this model represents an interesting balance between the flexibility of LLM-based zero-shot NER and the relatively small number of parameters of EncLM embeddings.

## 2.2 Unsupervised and Open-World NER

**Most Unsupervised Models are not Open-World** In theory, unsupervised models are open-world, given that the absence of annotated data implies auto-structuration and type discovery techniques (e.g., clustering). However, this is not always the case. Historically, unsupervised NER implemented rules and patterns-based models [11, 52]. They were specific to a small set of entity types, hindering the discovery of unspecified types. In fact, the most recent unsupervised NERs suffer from the same problem and require prior knowledge of the target entity types [34, 44, 57, 33]. Formally, they are zero-shot models (as they require the specification of entity types) and not unsupervised approaches.

Jia et al. [34], Liu et al. [44], and Peng et al. [57] propose to generalize the transfer-learning from $\mathcal{D}_S$ to $\mathcal{D}_T$ setting used in the zero-shot setting. They train entity-type-specific models based on BERT embeddings, which are merged together in a mixture of experts. They must know the target entity types beforehand and need access to labels for each entity type (from a different domain, but still annotations). CycleNER [33] proposes a seq-to-seq model with a double translation mechanism between text and entities. It comprises two models: S2E translating the document into its list of entities and E2S generating the text from a list of entities. The two models are trained jointly, and S2E is kept for predictions. CycleNER also

must know the target entity types in advance and requires lists of entities from $\mathcal{D}_T$.

**"True" Unsupervised and Open-World NER** To the best of our knowledge, only UNER [48] is compatible with true unsupervised and open-world scenarios. UNER uses clustering for MD and employs self-learning with auto-encoders for ET. However, UNER is subject to drifting (as it relies on self-learning) and requires careful hyperparameter tuning (number of training steps, learning rate, etc.) to prevent catastrophic performance drops. Unfortunately, UNER lacks source code and a detailed explanation of how these hyperparameters are adjusted unsupervised. It makes their results unreproducible.

As an aside, it is interesting to notice that the related domains of unsupervised and few-shot relation extraction also suffer from the hyperparameter tuning critique [58, 27]. They rely on training procedures (e.g., self-learning) sensitive to hyperparameter values that cannot be adjusted without accessing labeled data.

**Can Zero-Shot Models Be Directly Translated to an Open-World Setting?** The zero-shot and unsupervised settings are very similar, not needing annotated data in $\mathcal{D}_T$; the only difference is specifying entity types beforehand (zero-shot) or automatically discovering them (unsupervised, open-world). At first glance, the reader may think that zero-shot approaches can be easily translated to an open-world setting. But the truth is more complex. As presented in Sect. 2.1, we can divide zero-shot approaches into fine-tuned models (EncLM or LLMs) and frozen LLM prompting.

Fine-tuned approaches (based on EncLMs [86], full transformers [18] or LLMs [92]) all require the specification of an entity types schema beforehand, which is heavily employed during their training procedure. For instance, Ding et al. [18] or Lou et al. [47] experimentally observe that negative sampling (i.e., specifying entity types not mentioned in the current document) is a key to attaining state-of-the-art performances. However, if entity types are not specified (open-world), it is impossible to replicate such a training procedure, and the main contribution of these methods is lost. Similarly, Zaratiana et al. [86] require the list of predefined entity types in input as they are using the embeddings of the entity type names for their prediction. Older prototype-based or nearest-neighbor-based models are also not translatable, as they require labels to construct the prototypes or propagate the classes step by step. This category of models is not easily generalizable to an open-world setting, as removing the dependency on predefined entity types implies the definition of new input formats or training procedures.

Prompting of frozen LLMs [82, 83] is easier to adapt, as it necessitates adjusting the prompt to remove the dependency on pre-specified entity types (see Sect. 4.1). However, the impact on performances of un-specifying entity types from the prompt remains unevaluated, and we expect a performance drop compared to zero-shot prompting.

# 3 Description of OWNER

OWNER aims to extract and type entities from documents $\boldsymbol{X}$ of $\mathcal{D}_T$ in an unsupervised and open-world setting. Given $\boldsymbol{X}$, the objective is to identify the spans $\boldsymbol{e} = [x_i, ..., x_j] \in \boldsymbol{X}$ that are entities, and classify the type $t$ for each $\boldsymbol{e}$. OWNER assumes no prior knowledge of $\mathcal{D}_T$. It does not have access to:

- annotated documents of $\mathcal{D}_T$,
- the set of entity types $\mathcal{T}_T$,
- the number of entity types $|\mathcal{T}_T|$.

Similarly to recent zero-shot and few-shot models [86, 92, 68], OWNER is built upon a cross-domain transfer-learning scheme. The general idea is to learn the NER task on a source domain $\mathcal{D}_S$, where annotated data is available, and transfer it to $\mathcal{D}_T$. $\mathcal{D}_S$ differs from $\mathcal{D}_T$ stylistically, semantically, and/or from the entity type perspective ($\mathcal{T}_S \neq \mathcal{T}_T$). We go beyond zero-shot and few-shot approaches by not predefining $\mathcal{T}_T$.

As shown in Fig. 1, OWNER follows a two-step process, with:

1. Mention Detection (MD). It identifies the spans $\boldsymbol{e}$ of $\boldsymbol{X}$ that are entities.
2. Entity Typing (ET). It classifies the type $t$ for each extracted entity. In practice, OWNER finds clusters of entities with the same type $t$.

## 3.1 Mention Detection (MD)

MD identifies entities $\boldsymbol{e}$ for a given document $\boldsymbol{X}$.

As we have presented in the previous section, two main prediction paradigms exist for MD: BIO sequence labeling extractors [78, 20, 86], and span-based extractors [23, 49]. In general, span-based extractors achieve slightly better results than BIO models in supervised and low-resource settings [91, 86]. We choose to formulate MD as a BIO sequence labeling, classifying each $x_i \in \boldsymbol{X}$ as $B$ (first token of an entity), $I$ (second or following token of an entity), or $O$ (not an entity). We employ a BIO extractor due to its lower expressivity and complexity than span-based models, expecting it to lead to better generalizability on unseen domains and new entity types [23].

We employ EncLM embeddings, coming from pre-trained language models such as BERT [16], combined with a linear classifier:

$$\mathrm{f}_{\mathrm{MD}}(x_i, \boldsymbol{X}) = \sigma(\mathrm{EncLM}(x_i, \boldsymbol{X})\boldsymbol{W} + \boldsymbol{b}), \tag{1}$$

where $\boldsymbol{W}$ and $\boldsymbol{b}$ are learned weights, $\mathrm{EncLM}(x_i, \boldsymbol{X})$ is the EncLM embedding of $x_i$ in the context of $\boldsymbol{X}$, and $\sigma$ is the softmax function. $\mathrm{f}_{\mathrm{MD}}$ is fine-tuned (EncLM weights, $\boldsymbol{W}$ and $\boldsymbol{b}$) on annotated documents from $\mathcal{D}_S$.

In fact, MD is the primary motivation for annotated data. Indeed, the only MD model that works without labels relies on self-learning [48]. Yet, self-learning is known to be subject to drifting when overtrained. Preventing drifting requires careful hyperparameter tuning (especially the number of training steps and the learning rate). Luo et al. [48] do not explain how to adjust them without external annotated $\boldsymbol{X}$ from $\mathcal{D}_T$. As a result, we propose to use annotated documents from $\mathcal{D}_S$ to train MD in a supervised fashion (but cross-domain) to diminish the risk of unstable results. As a side note, annotations for $\mathcal{D}_S$ may come from manually labeled datasets, distantly annotated datasets [68], or synthetically generated data [92]. In this article, we train OWNER on manually labeled and synthetically generated datasets (see Sect. 5.2).

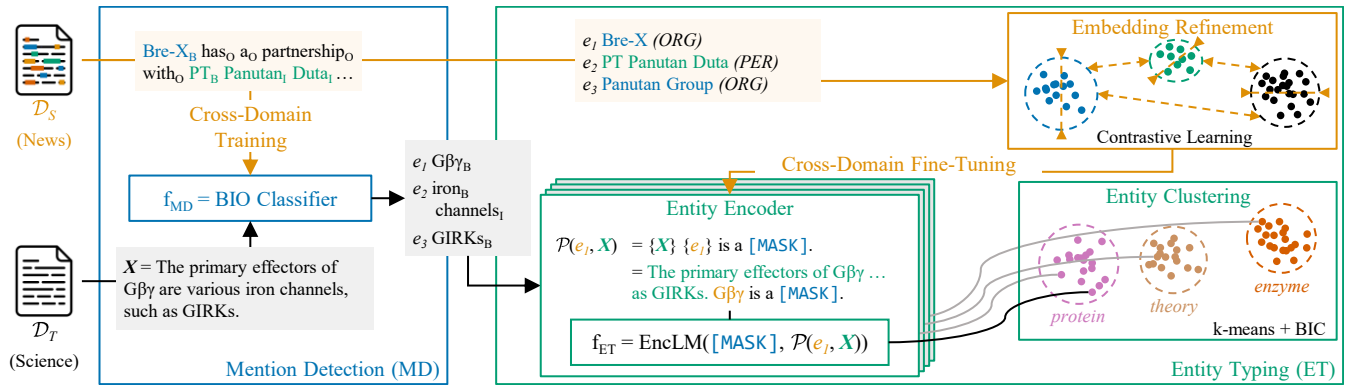



Figure 1: Overall architecture of OWNER.

## 3.2 Entity Typing (ET)

ET classifies the entities previously extracted with MD. In an unsupervised setting, the objective is to group entities with the same entity type $t \in \mathcal{T}_T$. As shown in Fig. 1, ET comprises three modules. They employ well-established technologies that have proven useful for NER, such as EncLM prompting, clustering, or contrastive learning [7]. To our knowledge, they have never been combined together[4], and this is their combination that enables open-world and unsupervised entity typing.

### 3.2.1 Entity Encoder

The first module of ET is the entity encoder, which computes a vector representation (or entity embedding) of the current entity. We want this embedding to represent the entity type: two entities $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ with close embeddings should have the same type $t$. Conversely, two entities with different $t_1$ and $t_2$ should have remote entity embeddings. To encode entities, we propose to use EncLM prompting [12, 27]. A prompt $\mathcal{P}$ is a text containing one `[MASK]` token, which the EncLM encodes. `[MASK]` indicates that the token is unknown, and the EncLM will compute an embedding representative of the missing word. By carefully designing $\mathcal{P}$, we can "ask" the EncLM the type $t$ of the current entity and use the `[MASK]` embedding as our entity embedding. The formulation of $\mathcal{P}$ is important for prompting performance and is usually adjusted using labels of $\mathcal{D}_T$ [69]. In our unsupervised setting, we decide not to tune it and choose the simplest template possible:

$$\mathcal{P}(\boldsymbol{e}, \boldsymbol{X}) = \text{"}\{\boldsymbol{X}\}\,\{\boldsymbol{e}\}\texttt{ is a [MASK].}\text{"}. \quad (2)$$

For instance (also shown in Fig. 1):

$\boldsymbol{X}$ = "The primary effectors of Gβγ are various iron channels, such as GIRKs.",
$\boldsymbol{e}$ = "Gβγ",
$\mathcal{P}(\boldsymbol{e}, \boldsymbol{X})$ = "The primary effectors of Gβγ are various iron channels, such as GIRKs. Gβγ is a [MASK]."

The entity representation is then computed as the embedding of `[MASK]` in the context of the prompt $\mathcal{P}(\boldsymbol{e}, \boldsymbol{X})$:

$$\text{f}_{\text{ET}}(\boldsymbol{e}, \boldsymbol{X}) = \text{EncLM}([\texttt{MASK}], \mathcal{P}(\boldsymbol{e}, \boldsymbol{X})). \quad (3)$$

[4]UNER [48] employs a very different auto-encoder approach for ET.

The choice of EncLM embeddings instead of LLM or full-transformers embeddings is motivated by two reasons. First, it allows us to define a fill-in-the-blank task that forces the model to predict precisely one word, most probably describing the entity type. Computing the entity embedding is then straightforward and does not require aggregation techniques (e.g., mean pooling, attention [19]). Second, EncLMs are much smaller than LLMs (10 – 100 times smaller), making them more applicable in resource-constrained environments.

### 3.2.2 Entity Clustering

Once all entities extracted in $\mathcal{D}_T$ are encoded using the previous module, we cluster the embeddings to identify groups of entities that are close and thus expected to have the same type $t \in \mathcal{T}_T$. We use the simple k-means algorithm [46, 50][5]. Since the number of entity types is unknown, the number of entity types (clusters) $k$ must be estimated.

As a side note, the only unsupervised prior work, UNER [48], required $k$ to be known in advance. This constraint is counter-intuitive and unrealistic: if $\mathcal{T}_T$ is unknown, we cannot determine $|\mathcal{T}_T|$ (and thus $k$). That is why we want to estimate $k$ automatically.

**Brute-Force Estimation** Interestingly, k-means can be seen as a simplification and approximation of a spherical Gaussian Mixture Model (GMM) [22]. The main difference resides in cluster membership: with k-means, each point belongs only to one cluster (Dirac probability distribution), whereas GMM produces soft-clustering assignments. One approach to estimate the number of clusters of a GMM is to fix an upper bound $K$, compute a clustering for each $k, 2 \leq k \leq K$, compute the Bayesian Information Criteria (BIC) [65] for each clustering and select $\hat{k}$ that minimizes BIC. BIC measures the clustering quality and adjusts it relative to the complexity of the model. Indeed, when looking at the right-hand side of Eq. (4), the left term measures the quality of the fit, while the right part estimates the complexity of the model. BIC finds a good tradeoff between the clustering quality and its complexity (number of

[5]Genest et al. [27] observed empirically that the best-performing clustering algorithm for unsupervised relation extraction was k-means. More complex approaches attained lower results, probably due to their increased expressivity that modeled noise instead of valuable information.

clusters). We propose to apply this same procedure to estimate the number of clusters with k-means, using the k-means BIC formula of Onumanyi et al. [53]:

$$BIC = n \ln(\frac{RSS}{n}) + k \ln(n), \quad (4)$$

$$RSS = \sum_{0 \leq i < n} (\mathrm{f_{ET}}(\boldsymbol{e}_i, \boldsymbol{X}_i) - \boldsymbol{c}_i)^2, \quad (5)$$

with $n$ the number of entities $\boldsymbol{e}_i$ extracted by MD, $\boldsymbol{X}_i$ the document containing $\boldsymbol{e}_i$ and $\boldsymbol{c}_i$ the centroid of the cluster containing $\boldsymbol{e}_i$. We call this procedure *brute force cluster estimation*. This is the main approach we employ during OWNER's evaluation.

**Ternary Search** One constraint of the previous approach is that it requires to compute a clustering for each $2 \leq k \leq K$, which is computationally expensive. Empirically, we find the BIC curve for ET to be smooth, globally convex, and with a single minimum (see Fig. 8 (a)). This was observed for the 13 $\mathcal{D}_T$ datasets used during evaluation (see Sect. 4.2), different EncLM embeddings, and every variation of OWNER. With this experimental observation, finding the global minimum BIC without testing every possible $k$ is possible. One such method is the ternary search. We propose implementing it and call it *ternary search cluster estimation*. The ternary search follows an iterative approach, with each cycle being:

1. In input, we have a lower bound $k_{min}$ and an upper bound $k_{max}$ for the number of clusters.
2. Select $k_1$ and $k_2$ such as they divide the search space between $k_{min}$ and $k_{max}$ in thirds.
3. For $k_1$ and $k_2$, compute the clustering and calculate the BIC.
4. If $k_1$ has a lower BIC than $k_2$, then $k_{max} = k_2$, else $k_{min} = k_1$.

The cycle is repeated until $k_{max} = k_{min}$. At each cycle, the search space is reduced by a third, giving a logarithmic complexity of $O(\log_3(K) \cdot \text{k-means})$, compared to $O(K \cdot \text{k-means})$ for the brute force method.

In practice, three improvements can be made. First, if the lowest BIC is at $k_{min}$, we set $k_{max} = k_1$; and conversely, if the lowest BIC is $k_{max}$, we set $k_{min} = k_2$. It allows the elimination of two-thirds of the search space in one cycle.

Secondly, we propose to remove the need to fix an upper bound $K$. We provide a first estimate $k_{max} = \sqrt{n}$ and allow the ternary search to increase $k_{max}$ if the minimum BIC is located after it. During the first cycle, if the lowest BIC is located at $k_{max}$, instead of updating $k_{min}$, we set $k_{max} = k_{max} + \frac{k_{max} - k_{min}}{3}$. This move is possible for the following cycles until the lowest BIC is not at $k_{max}$.

Finally, the BIC curve is not completely smooth locally. To improve the minimum estimation accuracy, when $k_{min}$ and $k_{max}$ are close (e.g., $k_{max} - k_{min} \leq 5$), we compute every clustering for $k_{min} \leq k \leq k_{max}$ and select $\hat{k}$ with the lowest BIC.

The pseudocode of the ternary search cluster estimation is displayed in Fig. 2. The function call from the user should be *Ternary-Search*$(1, \sqrt{n}, true)$. In practice, memoization is implemented to avoid recomputing the BIC multiple times for the same $k$, but it is skipped in the figure for clarity.

**function** *Ternary-Search*($k_{min}$, $k_{max}$, $firstcycle$)
**begin**
  **Data:** $k_{min}$ the lower bound for $k$, $k_{max}$ the upper bound for $k$, $firstcycle$ if the upper bound can be increased.
  **Result:** $\hat{k}$ estimation of the number of clusters.
  **if** $|k_{max} - k_{min}| < 5$ **then**
    **return** $\arg\min_{k_{min} \leq k \leq k_{max}}(\mathrm{BIC}(k))$
  $k_1 = k_{min} + \mathrm{floor}\left(\frac{k_{max} - k_{min}}{3}\right)$
  $k_2 = k_{max} - \mathrm{floor}\left(\frac{k_{max} - k_{min}}{3}\right)$
  $k_{best} = \arg\min_{k \in \{k_{min}, k_1, k_2, k_{max}\}}(\mathrm{BIC}(k))$
  **if** $k_{best} = k_{min}$ **then**
    **return** *Ternary-Search*$(k_{min}, k_1, false)$
  **else if** $k_{best} = k_1$ **then**
    **return** *Ternary-Search*$(k_{min}, k_2, false)$
  **else if** $k_{best} = k_2$ **then**
    **return** *Ternary-Search*$(k_1, k_{max}, false)$
  **else if** $k_{best} = k_{max}$ **and** $firstcycle$ **then**
    **return** *Ternary-Search*$(k_{min}, k_{max} + \frac{k_{max} - k_{min}}{3}, true)$
  **else**
    **return** *Ternary-Search*$(k_2, k_{max}, false)$
**end**

Figure 2: Pseudocode of the ternary search algorithm to estimate the number of clusters.

### 3.2.3 Embedding Refinement (ER)

ET is not trained using labeled documents. However, since MD uses labeled data in $\mathcal{D}_T$, we can also employ them for ET to isolate entity types more clearly during the clustering. Contrastive learning has been applied for this purpose in the context of low-resource NER [31, 14]. The objective is to bring entities of the same type closer and move away entities of different types by optimizing EncLM representations. Existing models apply contrastive learning on the annotated data of $\mathcal{D}_T$, which we do not have. As a result, we propose optimizing the contrastive loss on entities of $\mathcal{D}_S$, anticipating that the reorganized embedding space will also benefit entities in $\mathcal{D}_T$.

We implement the widely used triplet margin loss $\mathcal{L}_{\mathrm{TM}}$ [4]. $\mathcal{L}_{\mathrm{TM}}$ considers entity triplets $(\boldsymbol{e}^a, \boldsymbol{e}^+, \boldsymbol{e}^-)$. $\boldsymbol{e}^a$ is called the anchor. The positive entity $\boldsymbol{e}^+$ has the same type as the anchor $\boldsymbol{e}^a$, and the negative entity $\boldsymbol{e}^-$ has a different type than $\boldsymbol{e}^a$. The objective of $\mathcal{L}_{\mathrm{TM}}$ is to ensure that $\boldsymbol{e}^+$ is closer to $\boldsymbol{e}^a$ than $\boldsymbol{e}^-$ up to a certain margin. We have:

$$\mathcal{L}_{\mathrm{TM}}(\boldsymbol{e}^a, \boldsymbol{e}^+, \boldsymbol{e}^-) = \max[0, \mathrm{d}(\boldsymbol{e}^a, \boldsymbol{e}^+) - \mathrm{d}(\boldsymbol{e}^a, \boldsymbol{e}^-) + 1] \quad (6)$$

with $\mathrm{d}(\boldsymbol{e}^a, \boldsymbol{e}^+)$ the Euclidian distance between $\mathrm{f_{ET}}(\boldsymbol{e}^a, \boldsymbol{X})$ and $\mathrm{f_{ET}}(\boldsymbol{e}^+, \boldsymbol{X})$. $\mathrm{f_{ET}}$ weights are fine-tuned on entities of $\mathcal{D}_S$ using $\mathcal{L}_{\mathrm{TM}}$. We fix the $\mathcal{L}_{\mathrm{TM}}$ margin at 1. Empirically, we have not found that the margin significantly impacted the performances.

Contrary to usual EncLM fine-tuning, a larger batch size is beneficial with contrastive learning [5], as it helps regularize the embedding space reorganization. The limiting factor to increase the batch size with ET is entity encoding. For each triplet $(\boldsymbol{e}^a, \boldsymbol{e}^+, \boldsymbol{e}^-)$, three prompts $\mathcal{P}$ need to be encoded. This comes with a substantial GPU footprint, hindering large batch sizes.

**System Message:** You are a helpful information extraction system.

**Prompt:** 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: {$\boldsymbol{X}$}

Figure 3: Unsupervised prompt used by Zhou et al. [92] to annotate Pile-NER. It also corresponds to the prompt of UniNER Uns (GPT-3.5).

To mitigate this issue, we change the perspective and consider batches of entities instead of batches of triplets. Each entity $\boldsymbol{e}$ is associated with the document $\boldsymbol{X_e} \in \mathcal{D}_S$ in which it appears and its type $t_{\boldsymbol{e}} \in \mathcal{T}_S$. We encode one prompt for each entity. Then, we find all valid triplets inside the batch, respecting the condition $(t_{\boldsymbol{e}^+} = t_{\boldsymbol{e}^a}) \wedge (t_{\boldsymbol{e}^-} \neq t_{\boldsymbol{e}^a})$. We can encode 128 entities per batch in our experimental setup. Without this optimization, one batch comprises 42 triplets, and $\mathcal{L}_{\text{TM}}$ does not converge. With this optimization, one batch contains, on average, more than 100,000 valid triplets.

# 4 Experimental Setup

## 4.1 Baselines

Luo et al. [48] did not release the source code of UNER, the only comparable unsupervised and open-world baseline, and we could not reproduce their results. To solve this shortcoming, we propose an evaluation focusing on two directions.

**Zero-Shot Baselines (Closed-World)** First, we compare OWNER with state-of-the-art zero-shot NER models. These models are more supervised than OWNER (as they have access to the list of entity types) and are thus expected to achieve better results than ours. However, they allow us to contextualize the performance of unsupervised and open-world NER with more usual and standard low-resource approaches.

We include *UniNER* [92], *GoLLIE* [64], and *ChatIE (GPT-3.5)* [82]. We also evaluate ChatIE with the open-weight Llama 3 8B: *ChatIE (Llama 3)*. UniNER and GoLLIE are LLMs fine-tuned on synthetically or manually labeled datasets, whereas ChatIE implements "raw" prompting. We also test *GliNER L* [86] and *GNER* [18], which respectively use an EncLM (DeBERTav3 [30]) and a full transformer (Flan-T5 [60, 8]), both fine-tuned on the same dataset as UniNER.

We report the baselines' backbones and the number of parameters in Table 2.

**Unsupervised Baselines Creation** As no unsupervised and open-world baseline is currently evaluable, we propose creating two baselines based on zero-shot NERs. As we have seen in Sect. 2.2, not all zero-shot models can be translated to work in an open-world setting. LLM prompting is the only family of methods that can be directly adapted to work unsupervised and open-world.

1) Type Elicitation Prompt

**System Message:** A virtual assistant answers questions from a user based on the provided text.

**Prompt:** Given document: {$\boldsymbol{X}$}. Please answer: What types of entities are included in this sentence? Answer with a JSON list like: ["entity type 1", "entity type 2", ...].

2) Entity Extraction Prompt

**System Message:** A virtual assistant answers questions from a user based on the provided text.

**Prompt:** According to the document above, please output the entities of type "{$t$}" in the form of a JSON list like: ["entity 1", "entity 2", ...].

Figure 4: Unsupervised adaptation of the prompting method of ChatIE [82]. ChatIE follows a multi-turn question-answering setup, with the first prompt employed to identify the entity types mentioned in the current document and subsequent questions to identify entities for each elicitated entity type. This prompt is employed by ChatIE Uns (GPT 3.5) and ChatIE Uns (Llama 3 8B).

First, Zhou et al. [92] annotated the Pile-NER dataset by prompting GPT-3.5 without specifying entity types (see section 3.1 of their paper), thus in an unsupervised setting. They never evaluated this approach, and we include it to provide reference values of unsupervised GPT-3.5 prompting. We call this baseline *UniNER Uns (GPT-3.5)*. The prompt they used is displayed in Fig. 3. Additionally, we tried to replace GPT-3.5 with Llama 3 8B, but this smaller model could not respect the format specified in the prompt, resulting in null scores.

Second, the dual-stage method that ChatIE [82] implements, with type elicitation and entity extraction, can be translated to work under an unsupervised setting. Initially, type elicitation necessitates the list of entity types $\mathcal{T}_T$, but we can reformulate it to remove this dependency. The prompts employed are displayed in Fig. 4. We call this baseline *ChatIE Uns (GPT-3.5)*. We could successfully replace GPT-3.5 with Llama 3 8B, and we call this approach *ChatIE Uns (Llama 3)*. Finally, this baseline allows us to compare the performance between very similar zero-shot (ChatIE) and unsupervised (ChatIE Uns) models and observe the impact of not specifying entity types beforehand.

## 4.2 Datasets

### 4.2.1 Target Domain $\mathcal{D}_T$

Specific domains where annotated data is scarce or absent are the primary use cases of an unsupervised and open-world NER. We focus on datasets that differ from $\mathcal{D}_S$ stylistically (types of text), semantically (topics), and/or from the entity type perspective (unseen entity types). As a result, we evaluate OWNER on 13 domain-specific datasets:

- five CrossNER datasets [45] (*AI*, *Literature*, *Music*, *Politics*, and *Science*). They cover specific topics (scientific and literary) and unseen entity types.

- two MIT datasets [42] (*Movie* and *Restaurant*). They cover new styles of text (reviews and search engine queries), specific topics, and unseen entity types.
- *FabNER* [40] with physics and chemistry articles labeled with scientific entity types.
- *GENIA* [37] and *i2b2* [72] contain biomedical articles (taken from PubMed) annotated with biomedical entities.
- *GENTLE* [1] and *GUM* [87] cover unusual styles of text: e.g., dictionary entries, travel guides, legal notes, or poetry.
- *WNUT 17* [15] comprises social network posts.

These datasets cover a wide spectrum of types of text (encyclopedic, scientific, biomedical, social networks, customer reviews, dictionary entries, ...); domains (computer science, physics, chemistry, natural science, biomedical, literature, music, ...); and entity types (*algorithm*, *protein*, *cell type*, *poem*, *mechanical property*, *animal*, or *political party* among many others). It allows us to have a detailed picture of the quality and generalizability of OWNER.

#### 4.2.2 Source Domain $\mathcal{D}_S$

We propose to train OWNER with two datasets: *CoNLL-2003* [74], and *Pile-NER* [92]. They represent two different ways to envision the unsupervised setting.

CoNLL-2003 (named CoNLL thereafter) represents the cross-domain perspective. It contains general-domain newspaper articles manually annotated with four entity types (*person*, *location*, *organization*, and *misc*). CoNLL is chosen to be distant from $\mathcal{D}_T$ datasets stylistically, semantically, and from the entity type point of view. It allows us to evaluate the cross-domain capabilities of OWNER.

Pile-NER represents the synthetic data perspective. It comprises 50,000 documents gathered from the Pile corpus [26] automatically annotated by GPT-3.5[6], resulting in 13,000 fine-grained entity types. The idea is that large and diverse $\mathcal{D}_T$ datasets benefit the generalizability and partially close the stylistic, semantic, or entity type gap between $\mathcal{D}_S$ and $\mathcal{D}_T$. Nonetheless, as the annotation process is automatic and does not involve human actions, it is not time-consuming or expensive. In fact, the latest few-shot and zero-shot models use large amounts of automatically annotated $\mathcal{D}_S$ data (e.g., UniNER, GliNER L, and GNER train on Pile-NER), and the results show the benefits of these automatically labeled corpora.

### 4.3 Metrics

We divide evaluation metrics into two parts:

1. Mention Detection. They check whether the model extracts entities correctly without considering entity types.
2. Entity Typing. They check whether the model correctly classifies the entity type.

[6] As an aside, it is interesting to notice that Gao et al. [26] employed "real" documents from the Pile corpus instead of generating them with GPT-3.5. They argue having diverse documents and wide coverage of domains with LLM-generated documents is difficult, resulting in lower performance.

Entity typing metrics are also employed to evaluate end-to-end NER, combining mention detection and entity typing.

#### 4.3.1 Mention Detection

Similarly to previous works (Zong et al. [91] among others), we consider a predicted entity to be correct if its boundaries are the same as the ones of a ground truth entity. Thus, we define true positives (TP), false positives (FP), and false negatives (FN) as follows:

$$\hat{\boldsymbol{e}} = \boldsymbol{e} \iff \text{start}(\hat{\boldsymbol{e}}) = \text{start}(\boldsymbol{e}) \wedge \text{end}(\hat{\boldsymbol{e}}) = \text{end}(\boldsymbol{e}), \quad (7)$$

$$\text{TP}_{MD} = \sum_{\hat{\boldsymbol{e}}} \sum_{\boldsymbol{e}} \mathbb{1}_{\hat{\boldsymbol{e}}=\boldsymbol{e}}, \quad (8)$$

$$\text{FP}_{MD} = \sum_{\hat{\boldsymbol{e}}} \mathbb{1}_{\neg\exists \boldsymbol{e} \text{ s.t. } \hat{\boldsymbol{e}}=\boldsymbol{e}}, \quad (9)$$

$$\text{FN}_{MD} = \sum_{\boldsymbol{e}} \mathbb{1}_{\neg\exists \hat{\boldsymbol{e}} \text{ s.t. } \hat{\boldsymbol{e}}=\boldsymbol{e}}, \quad (10)$$

with $\text{start}(\boldsymbol{e})$ (resp. $\text{end}(\boldsymbol{e})$) the index in $\boldsymbol{X}$ of the first (resp. last) token of $\boldsymbol{e}$. By convention, this formulation has no true negatives (TN)[7]. We use the micro aggregation to compute the F1 score, precision (P), and recall (R). As a side note, F1 micro equals the accuracy because we are in a single-label prediction setting. We have:

$$\text{P}_{MD} = \frac{\text{TP}_{MD}}{\text{TP}_{MD} + \text{FP}_{MD}}, \quad (11)$$

$$\text{R}_{MD} = \frac{\text{TP}_{MD}}{\text{TP}_{MD} + \text{FN}_{MD}}, \quad (12)$$

$$\text{F1}_{MD} = \frac{2\,\text{P}_{MD}\,\text{R}_{MD}}{\text{P}_{MD} + \text{R}_{MD}}. \quad (13)$$

We notice that some recent LLM-based approaches [64, 92, 82] have changed the boundary check by a surface form check (i.e., checking that a predicted entity has the same text as a true entity)[8]. This modification is less precise than an exact boundary check and can be problematic when multiple entities with the same surface form in the same document have different types (e.g., "French persons speak French", the first French refers to a nationality, and the second to a language). In our evaluation, we evaluate all baselines and OWNER using the same boundary check metrics to ensure maximal fairness.

[7] A TN is a span that is not a true entity nor a predicted one. Given that the number of spans evolves quadratically depending on the size of the document and entities are relatively scarce, TNs would crush TPs, FPs, and FNs, leading to undiscriminative scores. Therefore, the consensus (e.g., [91, 86]) is to remove true negatives.

[8] In fact, this change is not documented in their respective papers, but it is present in their source code.

### 4.3.2 Entity Typing & End-to-End NER

As open-world methods determine entity types, the set of predicted clusters (predicted entity types) is not guaranteed to be equal to the set of true entity types. The predicted clusters cannot be directly mapped to the true entity types (no direct link between cluster IDs and class IDs). As a result, traditional classification metrics such as precision, recall, and F1 score cannot be used to evaluate open-world NER models.

Multiple metrics have been proposed to compare clustering to true labels. Compared to classification metrics, they are robust to permutations (meaning the cluster IDs will not impact the final score) and to partial matches (e.g., two or more clusters that correspond to a single class or the opposite). Two widely used metrics to compare clusterings with labels are the Adjusted Rand Index *ARI* [32, 71], and the Adjusted Mutual Information *AMI* [66][9]. In our experimental setup with unbalanced datasets, Romano et al. [61] recommend using AMI over ARI. AMI is adjusted for chance: a random clustering will reliably produce a score close to 0. Additionally, it is defined over $[-1, 1]$: scores below zero mean methods that are less effective than random clustering.

We recall the definition of AMI. First, the correspondences between the true and predicted entities are calculated with the equality defined in Eq. (7). A "predicted" placeholder is created with a specific error entity type for the true entities that were not predicted (FN). Conversely, for the predicted entities that do not exist in the ground truth (FP), a "true" placeholder with a specific error entity type is created.

The Mutual Information (MI) score measures the mutual dependence between the true entity type and predicted entity type random variables. It is defined as:

$$\begin{aligned} \mathrm{MI}(t, \hat{t}) &= \mathrm{H}(t) - \mathrm{H}(t|\hat{t}), \\ &= \mathrm{H}(\hat{t}) - \mathrm{H}(\hat{t}|t), \end{aligned} \tag{14}$$

with H the Shannon entropy [66]. $\boldsymbol{e}$, $\hat{\boldsymbol{e}}$, $t$, or $\hat{t}$ are considered to be random variables in the definitions[10]. To derive the actual values, the reader has to enumerate all $\boldsymbol{e} \in \mathcal{D}$ and $\hat{\boldsymbol{e}} \in \mathcal{D}$.

Adjusted Mutual Information (AMI) is the adjustment for the chance of MI, such that a random clustering will produce scores close to or equal to zero. It is defined as:

$$\mathrm{AMI}(t, \hat{t}) = \frac{\mathrm{MI}(t, \hat{t}) - \mathrm{E}_{t', \hat{t}'}\{\mathrm{MI}(t', \hat{t}')\}}{\max\{\mathrm{H}(t), \mathrm{H}(\hat{t})\} - \mathrm{E}_{t', \hat{t}'}\{\mathrm{MI}(t', \hat{t}')\}}. \tag{15}$$

$\mathrm{E}_{t', \hat{t}'}\{\mathrm{MI}(t', \hat{t}')\}$ is the expected MI between two random clusterings and is estimated using a hypergeometric model of randomness [77].

## 4.4 Implementation Details

OWNER follows a "train once, test anywhere" [59] methodology: it needs to be trained once on $\mathcal{D}_S$ and can be applied to multiple $\mathcal{D}_T$ datasets without further effort. Regarding hyperparameters, as OWNER is unsupervised, we cannot use validation data to adjust them. We opt for standard hyperparameter values defined by Devlin et al. [16].

**Entity Extraction** We use DeBERTa v3 embeddings [29, 30][11], train the model for 4 epochs, using the Adam optimizer [38], a decreasing linear schedule without warmup, a learning rate of $2 \times 10^{-5}$, a batch size of 32, and dropout ($p = 0.1$) between the EncLM and the linear classifier.

**Entity Typing** We use BERT embeddings[11]. We employ the simplest prompt possible, defined in Eq. (2), and train the model for 4 epochs, using the Adam optimizer [38], a decreasing linear schedule without warmup, a learning rate of $2 \times 10^{-5}$, a batch size of 128 as discussed in Sect. 3.2.3, and dropout ($p = 0.1$). For the brute force cluster estimation, we fix the upper bound $K$ to 50 and increase it if $\hat{k}$ is close to $K$: $K = 100$ for GUM, $K = 100$ for OWNER trained on CoNLL and tested on i2b2 and $K = 500$ for Pile-NER and i2b2.

**Computational Resources** Experiments were run on a single machine with 12 cores, 128 GB of RAM, and a GPU with 48 GB of VRAM. The required computational time is equivalent to BERT fine-tuning and depends on the size of the training dataset. With CoNLL, training usually last 50 min, and with Pile-NER, 5 h.

# 5 Results & Analysis

For OWNER, each experiment is repeated with five random seeds, and we report the average value and the standard deviation.

## 5.1 Comparison With the Baselines

The performances of OWNER and the zero-shot and unsupervised baselines on the 13 $\mathcal{D}_T$ datasets are reported in Fig. 5 and Table 1.

**Unsupervised Open-World Baselines** ● OWNER (Pile-NER) significantly outperforms all open-world baselines with an average AMI gap of 4.3 % with ◆ UniNER Uns (GPT-3.5), 18 % with ● ChatIE Uns (GPT-3.5), and 19 % with ● ChatIE Uns (Llama 3). ◆ UniNER Uns (GPT-3.5) has a short advantage of 1 % in AMI for Restaurant, which is the only time an open-world baseline attains better results than OWNER (Pile-NER).

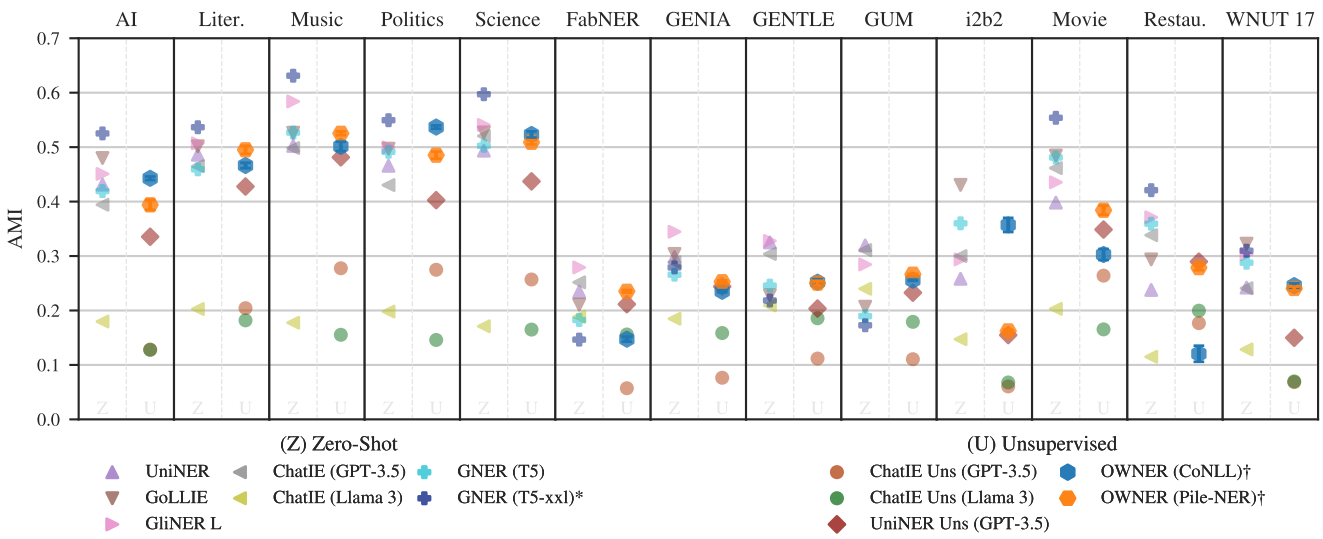
● OWNER (CoNLL), trained on a much more distant $\mathcal{D}_T$ dataset, surpasses UniNER Uns (GPT-3.5), ChatIE Uns (GPT-3.5), and ChatIE Uns (Llama 3) with average AMI gaps of 3.6 %, 17 %, and 19 %.

OWNER performs significantly better than LLM-based open-world NERs on a wide spectrum of domain-specific datasets. It demonstrates that our architecture effectively detects and types entities in an open-world and unsupervised setting. Finally, it is interesting to put the size of the compared baselines in perspective with the performances (see Table 2). OWNER is the smallest model with its 110 M parameters, yet it outperforms

[9] We are also aware of the V-measure [62] or the $B^3$ [2]. These metrics are, however, not adjusted for chance (see the subsequent paragraphs for an explanation).

[10] $\boldsymbol{e}$ and $\hat{\boldsymbol{e}}$ are not mentioned in the equations, but $t$ is dependent of $\boldsymbol{e}$ and $\hat{t}$ of $\hat{\boldsymbol{e}}$.

[11] We review the choice of EncLM embeddings in App. A.




* We could not run GNER (T5-xxl) on i2b2 due to excessive RAM consumption.
† The standard deviation of OWNER is displayed as a vertical bar.

Figure 5: NER performances (AMI) of OWNER, zero-shot, and unsupervised baselines. OWNER is less supervised and smaller than zero-shot baselines and smaller than unsupervised baselines. Exact values can be seen in Table 1.

Table 1: NER performances (AMI %) of OWNER, zero-shot, and unsupervised baselines. The best AMI for each $\mathcal{D}_T$ dataset and setting (zero-shot, unsupervised) is in **bold**, and the best AMI for each $\mathcal{D}_T$ dataset is in green.

| | AI | Liter. | Music | Politics | Science | FabNER | GENIA | GENTLE | GUM | i2b2 | Movie | Restau. | WNUT 17 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Zero-Shot* | | | | | | | | | | | | | |
| UniNER | 43.1 | 48.6 | 50.2 | 46.6 | 49.4 | 23.5 | 29.8 | 32.5 | **32.0** | 25.8 | 39.8 | 23.8 | 24.2 |
| GoLLIE | 48.0 | 50.2 | 52.6 | 49.7 | 52.7 | 21.1 | 30.4 | 22.8 | 20.7 | **43.1** | 48.5 | 29.4 | **32.3** |
| GliNER L | 45.1 | 50.7 | 58.4 | 50.0 | 54.1 | **27.9** | **34.5** | **32.8** | 28.4 | 29.4 | 43.6 | 37.1 | 30.3 |
| ChatIE (GPT-3.5) | 39.4 | 46.5 | 49.8 | 43.0 | 52.1 | 25.2 | 28.9 | 30.4 | 31.1 | 30.0 | 46.2 | 33.8 | 24.1 |
| ChatIE (Llama 3) | 18.0 | 20.3 | 17.8 | 19.8 | 17.1 | 18.9 | 18.5 | 20.9 | 24.0 | 14.7 | 20.3 | 11.5 | 12.8 |
| GNER (T5-xxl) | **52.5** | **53.7** | **63.1** | **54.9** | **59.7** | 14.7 | 27.9 | 21.8 | 17.3 | * | **55.4** | **42.1** | 31.0 |
| GNER (T5) | 41.9 | 45.9 | 52.7 | 49.1 | 50.2 | 18.3 | 26.6 | 24.6 | 19.0 | 36.0 | 48.1 | 35.9 | 28.8 |
| *Unsupervised* | | | | | | | | | | | | | |
| UniNER Uns (GPT-3.5) | 33.5 | 42.8 | 48.1 | 40.3 | 43.7 | 21.2 | 24.4 | 20.4 | 23.3 | 15.5 | 34.9 | **29.0** | 15.0 |
| ChatIE Uns (GPT-3.5) | 12.8 | 20.4 | 27.8 | 27.5 | 25.7 | 5.7 | 7.6 | 11.2 | 11.1 | 6.1 | 26.4 | 17.7 | 6.8 |
| ChatIE Uns (Llama 3) | 12.8 | 18.2 | 15.5 | 14.6 | 16.5 | 15.6 | 15.9 | 18.6 | 17.9 | 6.8 | 16.5 | 20.0 | 7.0 |
| OWNER (CoNLL) | **44.3(3)** | 46.6(5) | 50.1(9) | **53.7(3)** | **52.3(6)** | 14.7(5) | 23.5(2) | **25.2(5)** | 25.6(1) | **35.7(1.3)** | 30.3(1.0) | 12.1(1.5) | **24.6(4)** |
| OWNER (Pile-NER) | 39.4(9) | **49.5(8)** | **52.5(3)** | 48.5(7) | 50.9(4) | **23.5(2)** | **25.3(3)** | 25.0(4) | **26.7(1)** | 16.2(2) | **38.4(8)** | 27.9(5) | 24.0(3) |

The standard deviation of OWNER is printed in parentheses.
* We could not run GNER (T5-xxl) on i2b2 due to excessive RAM consumption.

Table 2: Number of parameters of OWNER, zero-shot, and unsupervised baselines. OWNER is more than 60-100 times smaller than LLM-based NERs.

| | Model | Backbone | # Parameters | |
|---|---|---|---|---|
| *Zero-Shot* | UniNER | Llama | 7 B | (×60) |
| | GoLLIE | Code-Llama | 7 B | (×60) |
| | GliNER L | DeBERTa v3 | 300 M | (×2.7) |
| | ChatIE (GPT-3.5) | GPT 3.5 | † | |
| | ChatIE (Llama 3) | Llama 3 | 8 B | (×70) |
| | GNER | Flan T5 | 275 M | (×2.5) |
| | GNER (T5-xxl) | Flan T5 XXL | 11 B | (×100) |
| *Unsupervised* | UniNER Uns (GPT-3.5) | GPT 3.5 | † | |
| | ChatIE Uns (GPT-3.5) | GPT 3.5 | † | |
| | ChatIE Uns (Llama 3) | Llama 3 | 8 B | (×70) |
| | OWNER | DeBERTa v3<br>BERT | 110 M* | |

† Although not disclosed, GPT-3.5 is expected to be larger than Llama 3.

* OWNER uses two encoders with a total of 200 M parameters (90 M for DeBERTa v3 and 110 M for BERT). But at any given time, only one is loaded.

much larger LLM baselines that are one to two orders of magnitude bigger.

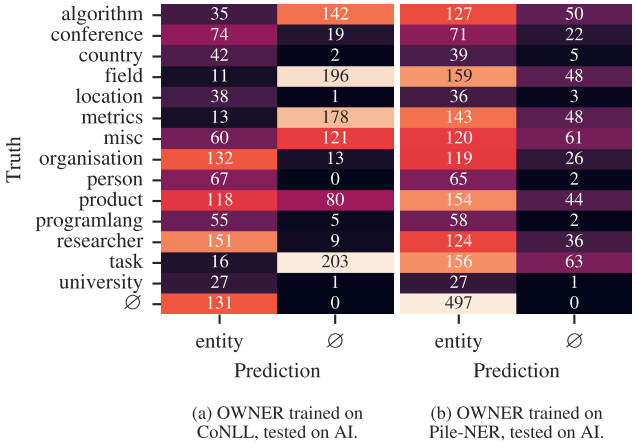
**Zero-Shot Baselines** Even when compared to the closed-world zero-shot models, OWNER is not out of the picture. ● OWNER (Pile-NER) performs significantly better than ◀ ChatIE (Llama 3), and matches or surpasses the performances of ▲ UniNER on six datasets, ◀ ChatIE (GPT-3.5) on five datasets, ✚ GNER (T5) on five datasets, ▼ GoLLIE on four datasets, ✚ GNER (T5-xxl) on three datasets, and ▶ GliNER L on one dataset. In general, zero-shot baselines outperform OWNER, which is expected, given they have access to the list of entity types (which is a form of supervision). Nevertheless, without accessing annotated data in $\mathcal{D}_T$ nor knowing the target entity types $\mathcal{T}_T$, OWNER attains honorable results when compared to the state-of-the-art zero-shot approaches.

To contextualize the performances of zero-shot models, the comparison between ChatIE (GPT-3.5) and ChatIE Uns (GPT-3.5) is interesting. Indeed, ChatIE Uns uses the same technique as ChatIE for NER, except it does not have access to the list of entity types. We can see in Table 1 that ChatIE Uns performances are extremely low compared to ChatIE: the average gap is 21 % in AMI. The small modification of removing the predefined list of entity types tremendously impacts performance. This demonstrates that entity type specification is a strong supervision signal and, thus, that unsupervised NER is a much more challenging task than zero-shot NER.

When looking closely, ChatIE Uns does not group entities together in coherent entity types and results in predicting over-specific entity types (see Table 6). For instance, ChatIE has identified 11,840 entity types (instead of 10) on the GUM dataset, such as *lantern festival*, *theme music*, *light show*, *laser light show*. It is also a problem of UniNER Uns, at a lesser degree, though (see Sect. 5.5).

Table 3: Comparison of precision (P) and recall (R) (in %) of OWNER for MD between CoNLL and Pile-NER. The standard deviation is displayed in parentheses. Each $\mathcal{D}_T$ dataset's best precision and recall are in **bold**.

| | $\mathcal{D}_S$ | CoNLL | | Pile-NER | |
|---|---|---|---|---|---|
| | | P | R | P | R |
| $\mathcal{D}_T$ | AI | **86.2(2)** | 46.6(5) | 74.1(6) | **77.5(5)** |
| | Liter. | **87.2(8)** | **80.9(4)** | 85.5(3) | 77.6(2) |
| | Music | 84.1(4) | 74.5(2) | **85.5(3)** | **82.8(4)** |
| | Politics | 78.9(3) | **82.3(2)** | **82.1(5)** | 81.2(3) |
| | Science | **82.8(5)** | 67.6(9) | 81.1(5) | **81.3(7)** |
| | FabNER | **52.0(7)** | 5.5(3) | 25.8(6) | **18.6(7)** |
| | GENIA | **46.5(1.2)** | 27.1(1.4) | 46.3(2) | **60.9(1.0)** |
| | GENTLE | 32.0(1.2) | 6.8(2) | **33.1(6)** | **20.6(3)** |
| | GUM | 25.8(2) | 6.4(1) | **28.6(1)** | **14.2(3)** |
| | i2b2 | **22.1(2.2)** | 26.8(1.4) | 5.5(2) | **29.6(9)** |
| | Movie | **89.9(1.6)** | 23.0(6) | 71.8(1.2) | **46.0(9)** |
| | Restau. | **57.4(3.1)** | 4.0(1.0) | 51.8(1.0) | **32.6(9)** |
| | WNUT 17 | **57.1(7)** | 74.1(1.2) | 41.1(4) | **76.6(4)** |




(a) OWNER trained on CoNLL, tested on AI.

(b) OWNER trained on Pile-NER, tested on AI.

Figure 6: Confusion matrices of OWNER for MD tested on AI. The ∅ row shows the false positives, and the ∅ column shows the false negatives per entity type.

## 5.2 Cross-Domain Capabilities & Synthetic Annotations

The results in Fig. 5 demonstrate that OWNER works well with a distant $\mathcal{D}_T$ (CoNLL) and with synthetic data (Pile-NER), as they both lead to better performances than unsupervised baselines. ● OWNER (Pile-NER) has a slight 0.7 % advantage in AMI compared to ● OWNER (CoNLL).

However, when looking closely, CoNLL and Pile-NER build models with different behaviors (although similar performances). In Table 3, we display the precision and recall of OWNER for mention detection. Overall, OWNER tends to have more precision when trained on CoNLL and more recall when trained on Pile-NER. This is expected: the diversity of Pile-NER helps OWNER detect entities better, while the human quality of annotations in CoNLL helps OWNER be more precise. This observation is confirmed when we examine the confusion matrices in Fig. 6. On one side, Pile-NER leads to better detections of domain-specific entity types (such as *algorithm*, *field*, *metrics*, or *task*), but we also see an increase in false positives (497 for Pile-NER vs. 151 for CoNLL). On the other

Table 4: MD performances (F1 %) for different architectures trained on CoNLL and tested on five $\mathcal{D}_T$ datasets. The standard deviation is displayed in parentheses. We did not repeat the experiment for PURE and SpanProto as they were very slow to train. Each $\mathcal{D}_T$ dataset's best F1 score is in **bold**.

| | AI | Liter. | Music | Politics | Science |
|---|---|---|---|---|---|
| BIO (OWNER) | **60.5(4)** | **83.9(5)** | **79.0(2)** | **80.6(2)** | **74.4(7)** |
| PURE | 39.8 | 37.1 | 33.8 | 32.4 | 35.7 |
| SpanProto | 54.1 | 62.9 | 59.6 | 68.7 | 59.7 |
| WL-Coref | 57.4(8) | 68.3(1.7) | 66.9(2.1) | 72.1(1.3) | 63.4(3.3) |

side, CoNLL has a slightly better recall for *person*, *location*, or *organization*, which are precisely the entity types annotated in this dataset.

As a side note, some performances displayed in Table 3 are low: precision below 10 % for FabNER, GENTLE, GUM, or Restaurant (CoNLL), and recall below 10 % for i2b2 (Pile-NER). They are far from ideal and satisfactory for production deployment and demonstrate the complexity of cross-domain learning and open-world NER. But the LLM-based baselines achieve even lower results than OWNER (as displayed in Fig. 5).

The question of higher false positives with Pile-NER is interesting. We manually checked the 497 false positives displayed in Fig. 6. 53 % of them are correct entities not annotated in AI, 42 % intersect with a true entity (boundary problem), and 5 % are wrong predictions. Overall, the boundary problem explains the false positive gap between CoNLL and Pile-NER, probably resulting from Pile-NER's imperfect annotations.

The 53 % correct entities not annotated in AI come from existing entity types (most missing entities are acronyms, for instance, FPR = false positive rate) and new entity types (not in the 14 entity types annotated in AI). The fact that OWNER identifies correct entities of new entity types highlights its novelty detection capabilities. This behavior cannot be observed with the other zero-shot and few-shot baselines as they have a predefined set of entity types.

In conclusion, the cross-domain capabilities of OWNER are highlighted by the good results of OWNER (CoNLL) on the $\mathcal{D}_T$ datasets. Broadly speaking, the manual annotations of CoNLL bring precise results, and the diversity of Pile-NER provides better recall at the cost of precision. In a novelty detection or exploratory scenario, where recall is key, we advise the reader to use Pile-NER. Additionally, the analysis of the confusion matrices shows that OWNER identifies entities of novel entity types that are unknown beforehand.

## 5.3 BIO Sequence Labeling

In Sect. 3.1, we propose to use a BIO extractor for MD, as we expect the simplicity of this architecture to bring better generalizability on new target domains $\mathcal{D}_T$. In Table 4, we report the F1 score of different MD architectures, trained on CoNLL and tested on five $\mathcal{D}_T$ datasets. We evaluate the following architectures:

- BIO. It is the architecture implemented by OWNER.
- PURE [91]. A span-based extractor that combines the start and end embeddings of a candidate span with a perceptron.
- SpanProto [78]. A span-based extractor that uses bilinear neurons to combine start and end embeddings of a candidate span (which provides faster predictions compared to PURE).
- WL-Coref [19]. A span-based extractor that identifies the "head" of the entity and recomposes its boundaries using a convolutional network. This model tackles the quadratic complexity problem of traditional span-based extractors.

In a fully supervised setting, PURE, SpanProto, and WL-Coref are shown to be slightly better than BIO sequence labeling [91, 78, 19]. However, in our unsupervised cross-domain setting, BIO performs significantly better than span-based extractors, with an average gap of 40 % with PURE, 15 % with SpanProto, and 10 % with WL-Coref, while being faster to train. We believe that the simplicity of the BIO architecture reduces overfitting and benefits generalizability on new domains. As an aside, this observation was made by Fang et al. [23], who also use BIO sequence labeling for their few-shot MANNER model.

## 5.4 Impact of Embedding Refinement

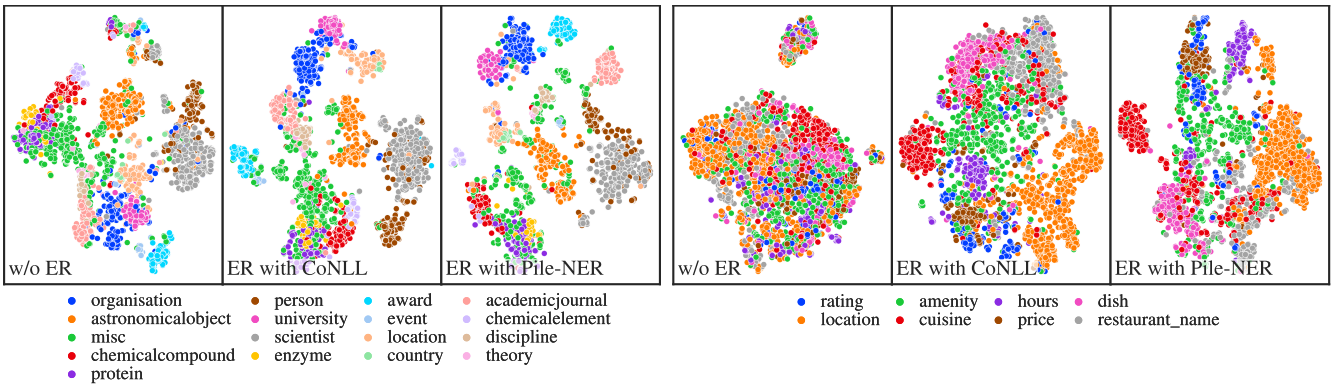
An important component of OWNER is embedding refinement (ER), which aims to improve EncLM representations for entity clustering using contrastive learning. In Table 5, we compare OWNER entity typing performance without ER and with ER trained on CoNLL or Pile-NER. We use the gold entity spans from $\mathcal{D}_T$ (no MD) to assess only the effect of ER. This is why the AMI scores are higher than in Table 1.

We see that ER has a significant positive impact with CoNLL and Pile-NER on each of the 13 $\mathcal{D}_T$ datasets, with an average AMI gain of 12.8 % for CoNLL and 16.7 % for Pile-NER compared to OWNER without ER. The gain is particularly impressive for datasets that are difficult for raw BERT embeddings, such as GENTLE, GUM, i2b2, Movie, Restaurant, or WNUT 17. Pile-NER's better performances can be explained by its diversity of entity types (13,000 entity types), which helps to fine-tune entity embeddings more precisely. Nevertheless, CoNLL achieves honorable performances despite only having four entity types. This validates the hypothesis that refining entity embeddings on $\mathcal{D}_S$ with contrastive learning benefits also distant $\mathcal{D}_T$.

To give a more visual representation of the effects of embedding refinement, we display in Fig. 7 two-dimensional t-SNE [76] representations of the entity embeddings of the Science and Restaurant datasets. The entities of Science are already well isolated without ER (see Table 5). Still, we can notice several improvements: better separation of *discipline*, *organization*, and *academicjournal* (CoNLL and Pile-NER); better separation of *chemicalelement* and *chemicalcompound* (Pile-NER); and the multi-type cluster at the top of the w/o ER figure has disappeared. The effects of ER are more visible with the difficult Restaurant dataset: without ER, ET cannot discriminate any entity type, and we see huge improvements with ER on CoNLL or Pile-NER. In particular, it is interesting to see that ER with CoNLL leads to a relatively good separation of *cuisine*, *hours*, or *price*, even though CoNLL does not contain such entities. The effects are more complete and more visible with Pile-NER.

Table 5: AMI scores (in %) of OWNER for ET on $\mathcal{D}_T$ datasets, without ER and with ER on CoNLL or Pile-NER. ET is evaluated using gold entity spans. The standard deviation is printed in parentheses. The best AMI for each $\mathcal{D}_T$ dataset is in **bold**.

| | AI | Liter. | Music | Politics | Science | FabNER | GENIA | GENTLE | GUM | i2b2 | Movie | Restau. | WNUT 17 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| w/o ER | 43.0(1.4) | 40.1(6) | 47.8(8) | 56.0(8) | 56.1(9) | 18.6(3) | 20.3(7) | 15.6(5) | 19.7(2) | 32.1(6) | 21.8(5) | 11.3(4) | 22.5(3) |
| ER on CoNLL | **56.8(1.4)** | 56.3(1.1) | 60.9(5) | 65.4(3) | **66.7(3)** | **26.7(7)** | 26.6(8) | 21.5(7) | 26.1(5) | 47.9(6) | 46.6(1.3) | 35.8(1.4) | 34.3(1.1) |
| ER on Pile-NER | 54.2(7) | **63.1(8)** | **64.2(5)** | **66.0(1.1)** | 66.0(9) | 24.1(8) | **31.7(6)** | **32.7(5)** | **37.0(2)** | **49.4(8)** | **52.1(8)** | **41.0(5)** | **41.1(8)** |




(a) OWNER tested on Science.

(b) OWNER tested on Restaurant.

Figure 7: Two-dimensional t-SNE visualizations of the entity embeddings of OWNER. For each subfigure from left to right: 1) without ER, 2) ER with CoNLL, and 3) ER with Pile-NER.

In conclusion, ER significantly improves ET performance with CoNLL and Pile-NER (a gap of resp. 12.8 % and 16.7 % in AMI). The best results are achieved with Pile-NER due to its diversity in entity types. ER works well with the distant $\mathcal{D}_T$ dataset CoNLL, with noticeable improvements on unseen entity types. It also shows that ER is beneficial even with a labeled dataset with a narrow set of entity types (4 for CoNLL).

## 5.5 Estimation of the Number of Clusters $\hat{k}$

As we do not have any information about entity types (contrary to zero-shot approaches), OWNER has to infer entity types and their number. In this part, we only consider the brute force cluster estimation. In Table 6, we display for each $\mathcal{D}_T$ dataset its true number of entity types $k$, the estimated number of clusters $\hat{k}$, the corresponding AMI score with $\hat{k}$ (similar to Fig. 5, that is end-to-end NER), and AMI score with the ideal $k$.

Overall, OWNER tends to overestimate the number of entity types; this effect is more pronounced with Pile-NER than with CoNLL. However, compared to UniNER Uns and ChatIE Uns, OWNER provides estimations that are much closer to the truth. Regarding Pile-NER, this overestimation behavior can be linked to its fine-grained entity types[12]. We can see this tendency in the visualization of Science in Fig. 7, where the *misc* class is divided into multiple small clusters (compared to CoNLL).

AMI scores with the ideal $k$ are close to AMI with $\hat{k}$ (AMI gap of 0.8 % for CoNLL and 1.5 % for Pile-NER on average), meaning that the clusterings are relatively similar from a qualitative point of view even with $\hat{k} \gg k$. The long-tail distribution of the cluster membership explains this. If we take the second confusion matrix of Fig. 9, a minority of clusters contains most entities, and the rest contain few specific entities. In fact, the 17 last clusters represent false positives[13] and members of the *misc* class (by definition, composed of multiple entity types). It explains why, even with this number of clusters, the performances do not plummet because the supplementary clusters model essentially false positives and composite classes.

## 5.6 Faster Estimation of the Number of Clusters $\hat{k}$

Up to this section, we used the brute force algorithm to estimate the number of clusters $\hat{k}$. The computational time is acceptable for the small datasets, but for the biggest $\mathcal{D}_T$ datasets (e.g., i2b2 or GUM), it can take up to hours (see Table 8), representing, in fact, the major part of the run. For instance, the cluster estimation lasts 13.6 h on average for $\mathcal{D}_S$ = Pile-NER and $\mathcal{D}_T$ = i2b2. This motivates the ternary search algorithm we presented in Sect. 3.2.2.

In Table 7, we display the comparison of the estimation of $\hat{k}$ between the brute force algorithm and the ternary search, and the corresponding NER AMI scores; and in Table 8 we display the corresponding execution time. We see that the estimation of $\hat{k}$ with ternary search equals the brute force algorithm or is in the standard deviation range. This results in ternary search AMI scores virtually identical to brute force scores.

More interesting is the gain in terms of computational time. As displayed in Table 8, the ternary search is 1.7 to 2.7 quicker to

[12] As a reminder, Pile-NER was annotated using UniNER Uns (GPT-3.5). De facto, Pile-NER exhibits the same fine-grained entity type weakness as UniNER Uns (GPT-3.5). Fortunately, OWNER partially mitigates this issue with a more reasonable estimation of the number of clusters, as shown in Table 6.

[13] That can be correct entities, as we have seen in Sect. 5.2.

Table 6: Estimation of the number of clusters $\hat{k}$ by OWNER using the brute-force approach and AMI scores (in %) for NER with true $k$ and estimated $\hat{k}$. The standard deviation is printed in parentheses. $k$ and $\hat{k}$ are displayed in green, and the best AMI score for each $\mathcal{D}_S$ and $\mathcal{D}_T$ dataset is in **bold**. We also include the number of entity types found by LLM-based unsupervised baselines.

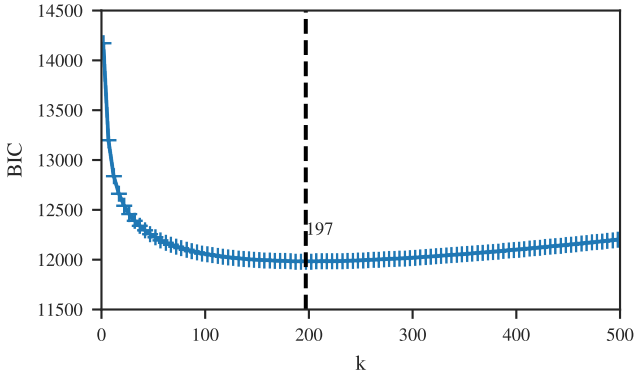
| | AI | Liter. | Music | Politics | Science | FabNER | GENIA | GENTLE | GUM | i2b2 | Movie | Restau. | WNUT 17 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| k | 14 | 12 | 13 | 9 | 17 | 12 | 5 | 10 | 11 | 23 | 12 | 9 | 6 |
| *OWNER (CoNLL)* | | | | | | | | | | | | | |
| $\hat{k}$ | 10 | 12(1) | 20(2) | 23(2) | 17(2) | 8(2) | 20(1) | 8 | 35(1) | 50(5) | 8 | 4 | 8(1) |
| AMI $\hat{k}$ | 44.3(3) | **46.6(5)** | 50.1(9) | 53.7(3) | 52.3(6) | 14.7(5) | 23.5(2) | **25.2(5)** | 25.6(1) | 35.7(1.3) | **30.3(1.0)** | **12.1(1.5)** | **24.6(4)** |
| AMI k | **44.5(3)** | 46.4(4) | **51.0(3)** | **56.5(1.9)** | **52.9(6)** | **14.8(5)** | **26.4(6)** | 25.1(7) | **27.0** | **38.7(9)** | 29.8(1.0) | 11.7(1.6) | **24.6(3)** |
| *OWNER (Pile-NER)* | | | | | | | | | | | | | |
| $\hat{k}$ | 18(1) | 16(1) | 26(1) | 32(2) | 29 | 32(3) | 35 | 22(1) | 59 | 197(4) | 26 | 14(2) | 16(1) |
| AMI $\hat{k}$ | **39.4(9)** | 49.5(8) | 52.5(3) | **48.5(7)** | 50.9(4) | 23.5(2) | 25.3(3) | 25.0(4) | 26.7(1) | 16.2(2) | 38.4(8) | 27.9(5) | 24.0(3) |
| AMI k | 39.2(7) | **50.2(6)** | **54.3(4)** | 47.8(6) | **51.7(5)** | **25.2(3)** | **29.0(5)** | **26.0(3)** | **28.8(1)** | **23.1(2)** | **38.7(9)** | **28.2(5)** | **25.1(6)** |
| *$\hat{k}$ estimated by the unsupervised baselines* | | | | | | | | | | | | | |
| UniNER Uns (GPT-3.5) | 155 | 92 | 115 | 103 | 195 | 292 | 319 | 250 | 830 | 1,033 | 176 | 117 | 266 |
| ChatIE Uns (GPT-3.5) | 1,427 | 954 | 1,074 | 1,141 | 1,480 | 5,108 | 4,342 | 1,323 | 11,840 | 14,680 | 1,214 | 899 | 1,707 |
| ChatIE Uns (Llama 3) | 197 | 61 | 123 | 74 | 408 | 1,276 | 1,643 | 374 | 1,433 | 6,014 | 107 | 88 | 714 |

Table 7: Estimation of the number of clusters $\hat{k}$ with brute force or ternary search and AMI scores (in %) for NER with true $k$ and estimated $\hat{k}$, when OWNER is trained on Pile-NER. The standard deviation is printed in parentheses. $k$ and $\hat{k}$ are displayed in green, and the best AMI score for each $\mathcal{D}_T$ dataset is in **bold**.

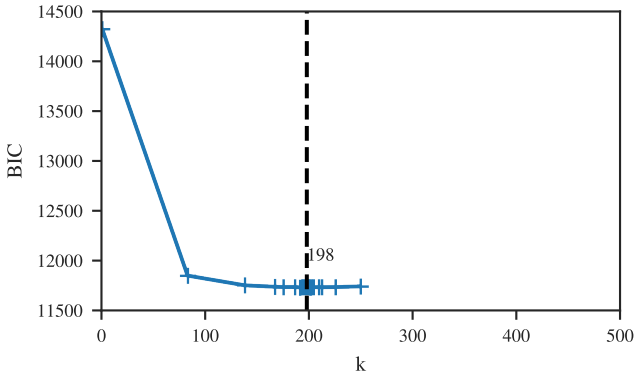
| | AI | Liter. | Music | Politics | Science | FabNER | GENIA | GENTLE | GUM | i2b2 | Movie | Restau. | WNUT 17 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $k$ | 14 | 12 | 13 | 9 | 17 | 12 | 5 | 10 | 11 | 23 | 12 | 9 | 6 |
| brute $\hat{k}$ | 18(1) | 16(1) | 26(1) | 32(2) | 29 | 32(3) | 35 | 22(1) | 59 | 197(4) | 26 | 14(2) | 16(1) |
| ternary $\hat{k}$ | 19(1) | 18(1) | 25(2) | 32(3) | 28(3) | 32(4) | 34(2) | 23(2) | 65(5) | 198(4) | 24(2) | 15(1) | 18(1) |
| AMI | 39.2(7) | **50.2(6)** | **54.3(4)** | 47.8(6) | **51.7(5)** | **25.2(3)** | **29.0(5)** | **26.0(3)** | **28.8(1)** | **23.1(2)** | **38.7(9)** | **28.2(5)** | **25.1(6)** |
| brute AMI | **39.4(9)** | 49.5(8) | 52.5(3) | 48.5(7) | 50.9(4) | 23.5(2) | 25.3(3) | 25.0(4) | 26.7(1) | 16.2(2) | 38.4(8) | 27.9(5) | 24.0(3) |
| ternary AMI | **39.4(7)** | 49.2(6) | 52.1(3) | **49.1(9)** | 50.6(7) | 23.4(2) | 25.3(2) | 25.0(5) | 26.5(2) | 16.2(2) | 38.7(6) | 28.0(3) | 24.1(4) |

Table 8: Execution time (in s) of the cluster estimation using the brute force or ternary search algorithms when OWNER is trained on Pile-NER.

| | AI | Liter. | Music | Politics | Science | FabNER | GENIA | GENTLE | GUM | i2b2 | Movie | Restau. | WNUT 17 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| brute AMI | 88(2) | 98(9) | 191(23) | 236(12) | 164(15) | 505(29) | 390(36) | 115(10) | 1,823(157) | 49,039(214) | 276(28) | 138(14) | 99(11) |
| ternary AMI | **48(1)** (÷1.8) | **52(1)** (÷1.9) | **69(1)** (÷2.8) | **89(3)** (÷2.6) | **69** (÷2.4) | **189(4)** (÷2.7) | **142(5)** (÷2.7) | **65(1)** (÷1.8) | **906(28)** (÷2.0) | **2,440(173)** (÷20.1) | **117(2)** (÷2.3) | **74** (÷1.9) | **57(1)** (÷1.7) |




(a) ET with the brute force algorithm.




(b) ET with the ternary search algorithm.

Figure 8: BIC curves computed to estimate the number of clusters $\hat{k}$, when OWNER is trained on Pile-NER and tested on i2b2. Each cross represents a computed clustering. With the brute force algorithm, 500 clusterings were calculated, and with ternary search, only 21.

run compared to the brute force algorithm, even on the smallest datasets. The gain is particularly impressive for the large i2b2 dataset with its large $\hat{k}$ where the gain is twenty-fold. Initially, the runs lasted 13.6 h, and with the ternary search, they are reduced to 41 min. The computational gain is less important for smaller sets of entity types (although still very significant) because of the slight rugosity of the BIC curve. This rugosity forces us to compute multiple clusterings sequentially once $k_{max} - k_{min} \leq 5$.

The case of the i2b2 dataset is especially interesting. In Fig. 8, ternary search quickly converges to the minimum value without evaluating every possible $\hat{k}$. In particular, the range $[0, 140]$ clusters is eliminated in two steps (5 min), whereas brute force needs 2 h to evaluate the same interval. Ternary search finds $\hat{k}$ after 21 clusterings, compared to the 500 needed for the brute-force algorithm (24 times less).

In conclusion, the computational gain of ternary search is particularly important with large $\mathcal{D}_T$ datasets with many different entity types. It is also relevant for smaller datasets, bringing a two-fold decrease in calculation time. Empirically, we find no significant difference in the estimation of $\hat{k}$ and AMI scores between brute force and ternary search.

### 5.7 Qualitative Analysis

We want to finish this analysis by giving a qualitative overview of the performances of OWNER. In Fig. 9, we display three confusion matrices of OWNER trained with different $\mathcal{D}_S$ datasets and tested on different $\mathcal{D}_T$.

The three confusion matrices show a relatively clear diagonal, meaning that OWNER correctly identifies most entity types. It is an impressive result: without annotated data in $\mathcal{D}_T$ nor any information on entity types or their count, OWNER detects and structures entities in a scheme similar to the ground truth.

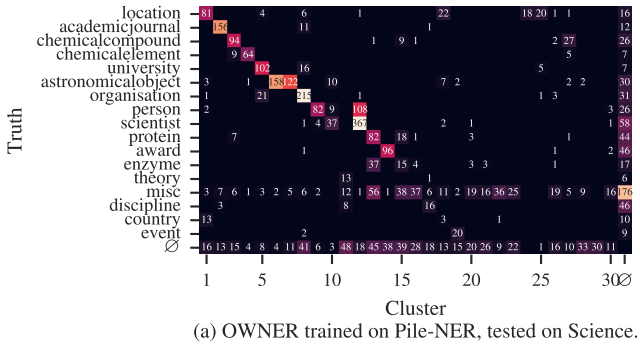
It is interesting to look at the confusions made by OWNER. OWNER merges *country* and *location* (Science and AI); *person* and *scientist/researcher* (Science and AI); *enzyme* and *protein* (Pile-NER Science); *task*, *product*, *field*, *algorithm* (AI); or *conference*, *university*, *organization* (AI). OWNER confuses semantically close entity types, which is a reassuring behavior. It is also a constraint linked to open-world NER. As we do not provide the list of entity types, OWNER organizes entities in a semantically coherent scheme that is a valid typing scheme but not exactly the dataset annotation schema.

Finally, OWNER organizes false positives and *misc* entities, a composite of multiple underlying types. It explains why OWNER tends to overestimate the true number of entity types.

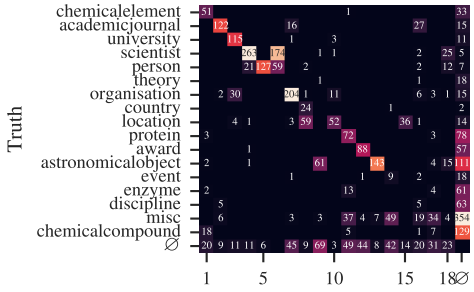
In conclusion, OWNER organizes entities in a coherent typing scheme that is close to the true entity types. This analysis also highlights OWNER's exploratory abilities. It can identify and organize entities into meaningful groups without labeled data in $\mathcal{D}_T$. OWNER efficiently processes unannotated documents to uncover primary entities and their types, setting the stage for further refinement through more supervised methods.

## 6 Conclusion

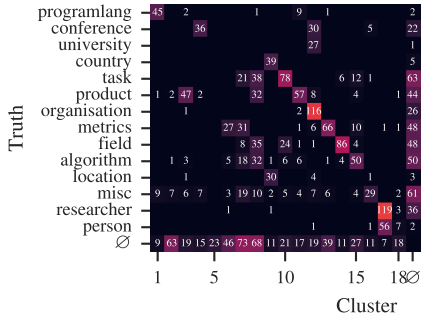
In this work, we introduce OWNER, our unsupervised and open-world NER model that transfers knowledge from $\mathcal{D}_S$ to $\mathcal{D}_T$ without supervision. The literature review showed that




(a) OWNER trained on Pile-NER, tested on Science.




(b) OWNER trained on CoNLL, tested on Science.




(c) OWNER trained on Pile-NER, tested on AI.

Figure 9: Confusion matrices of OWNER for NER tested on various $\mathcal{D}_T$ datasets. Columns and rows were reordered using the algorithm described in App. B. The ∅ row shows the false positives, and the ∅ column shows the false negatives.

unsupervised NER lags behind while significant progress has been made towards lower resource NER (in particular, zero-shot NER). OWNER is proposed to be the first NER compatible with an utterly unsupervised open-world scenario, to provide a strong baseline, and to stimulate further research. OWNER is built upon a simple yet innovative architecture with an EncLM prompting, clustering, and embedding refinement triangle.

Tests on 13 domain-specific datasets demonstrate that OWNER outperforms LLM-based open-world NERs and remains relevant when compared with state-of-the-art zero-shot NER models, without requiring prior knowledge of $\mathcal{D}_T$. This result is impressive, given that the simple EncLM embeddings of OWNER compete with much larger LLMs. We believe an essential point for OWNER's success is its architectural simplicity and parameter efficiency, which achieve state-of-the-art results.

Ablation studies show that embedding refinement brings significant performance gains and works well even with a distant $\mathcal{D}_T$ dataset. Ternary search shortens the computational time needed to estimate the number of clusters considerably (two times in general and up to twenty times faster on the largest dataset). Qualitative results demonstrate OWNER's exploratory capabilities and ability to organize entities in semantically coherent clusters close to actual entity types.

For future work, we aim to expand OWNER for use in a low-resource active learning context [67]. Specifically, we believe OWNER's capability to structure entities without supervision could help bootstrap an active learning cycle.

A second area for research is to combine open-world and closed-world NER. The objective would be to allow the user to predefine a typing scheme for entities he is aware of while leaving the door open to novel unseen knowledge, for which the model will provide a generated typing structure. Preliminary work [89, 21, 90] has been done in the related relation extraction field, but these models are not currently low-resource.

## Acknowledgements

This work is supported by Alteca and the French Association for Research and Technology (ANRT) under CIFRE PhD fellowship n°2021/0851.

# A EncLM Embeddings Impact

With OWNER, we primarily utilize DeBERTa v3 [29, 30] for mention detection and BERT [16] for entity typing. In this section, we evaluate the performances of other popular EncLM such as RoBERTa [43], ERNIE [73], or ELECTRA [9].

In Table 9, we display the MD performances of various EncLMs when OWNER is trained on Pile-NER, and in Table 10, we show the ET performances of the same EncLMs (also on Pile-NER). Broadly speaking, OWNER works relatively well, regardless of the EncLM used as a backbone. In fact, all the evaluated EncLM embeddings lead to better performances than UniNER Uns (GPT-3.5), ChatIE Uns (GPT-3.5), and ChatIE Uns (Llama 3). Interestingly, the "older" model, BERT, is not out of the picture and performs similarly to more recent alternatives.

Table 9: MD performances (F1 %) of OWNER trained on Pile-NER, using various EncLM embeddings. The standard deviation is printed in parentheses. The best F1 for each $\mathcal{D}_T$ dataset is in **bold**. The last column displays the average F1 across the 13 $\mathcal{D}_T$ datasets.

| | AI | Liter. | Music | Politics | Science | FabNER | GENIA | GENTLE | GUM | i2b2 | Movie | Restau. | WNUT 17 | Average |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BERT | 73.7(5) | 76.4(2) | 80.3(2) | 79.7(5) | 78.4(3) | 20.0(6) | 50.7(3) | 23.3(3) | **19.0(1)** | 9.2(3) | 56.9(7) | 38.4(6) | 47.6(6) | 50.3 |
| RoBERTa | 74.3(5) | 79.5(3) | 81.9(4) | 80.5(2) | 78.8(3) | 20.5(5) | 51.2(5) | 23.9(7) | 18.9(4) | **9.6(4)** | 52.2(1.9) | 39.8(6) | **54.2(8)** | 51.2 |
| ERNIE | 73.4(2) | 76.0(4) | 80.7(2) | 80.1(4) | 78.0(4) | 20.6(2) | 51.2(5) | 22.6(5) | **19.0(2)** | 9.4(2) | 57.9(5) | 40.0(7) | 48.2(5) | 50.5 |
| ELECTRA | 73.9(4) | 76.3(3) | 81.3(2) | 79.6(4) | 79.2(2) | 20.5(3) | 51.4(3) | 23.1(6) | 18.2(3) | 9.5(3) | **59.6(3)** | **41.5(6)** | 48.5(6) | 51.0 |
| DeBERTa v3 | **75.6(5)** | **81.4(4)** | **84.6(3)** | **81.6(3)** | **80.9(5)** | **21.2(6)** | **52.2(4)** | **25.2(3)** | 18.9(3) | 9.5(1) | 56.1(1.1) | 39.8(8) | 53.4(5) | **52.4** |

Table 10: ET performances (AMI %) of OWNER trained on Pile-NER, using various EncLM embeddings. ET is evaluated using gold entity spans. The standard deviation is printed in parentheses. The best AMI for each $\mathcal{D}_T$ dataset is in **bold**. The last column displays the average AMI across the 13 $\mathcal{D}_T$ datasets. The number of clusters is estimated using the ternary search algorithm, which explains why AMI scores are not identical to Table 5 (brute force). They are nevertheless in the range of standard deviation.

| | AI | Liter. | Music | Politics | Science | FabNER | GENIA | GENTLE | GUM | i2b2 | Movie | Restau. | WNUT 17 | Average |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BERT | **54.3(3)** | **64.1(1.0)** | 64.4(5) | **66.2(1.3)** | **65.9(4)** | 24.5(5) | **32.1(5)** | 33.8(7) | 35.7(1) | 50.2(5) | 52.0(3) | 40.7(8) | 40.9(1.2) | 48.1 |
| RoBERTa | 53.7(8) | 63.7(1.1) | **64.7(7)** | 63.3(1.7) | 65.7(1.0) | **25.0(7)** | 28.1(4) | **34.0(5)** | **36.2(3)** | **51.2(7)** | 47.4(6) | 47.2(6) | 44.1(5) | 48.0 |
| ERNIE | 54.2(7) | 62.7(1.4) | 64.4(9) | 63.0(1.2) | **65.9(8)** | 24.7(4) | 30.2(5) | 33.8(6) | 35.6(3) | 48.5(3) | **54.2(8)** | 47.3(1.0) | 44.2(3) | **48.4** |
| ELECTRA | 53.6(2) | 57.9(2.0) | 61.1(1.2) | 56.7(1.5) | 62.7(1.5) | 22.7(7) | 26.4(2.6) | 32.8(5) | 34.1(1.2) | 48.5(1.1) | 51.8(9) | 45.7(1.4) | 41.8(6) | 45.8 |
| DeBERTa v3 | 53.0(1.0) | 59.0(1.1) | 61.6(7) | 58.5(1.3) | 62.9(8) | **25.0(2)** | 25.4(4) | 33.4(3) | 34.3(1) | 50.8(5) | 48.7(6) | **47.6(9)** | **46.7(6)** | 46.7 |

For MD, we see an advantage of DeBERTa v3 over the other approaches, with an average gap of 1.2 % with the second-best model RoBERTa. We link these better performances to the richer and broader pre-training dataset compared to the other EncLM. BERT achieves the worst performances. This explains why we have chosen DeBERTa v3 as the backbone for MD.

The performances are closer for ET, with BERT, RoBERTa, and ERNIE nearly indistinguishable (especially given the standard deviation). ELECTRA and DeBERTa v3 have lower AMI scores. The behavior of DeBERTa v3 is surprising, as it is generally recognized as the best-performing EncLM currently available. The performances of DeBERTa v3 are even worse without ER (not shown), achieving half of those of BERT without ER. The same conclusion can be drawn with ELECTRA. DeBERTa v3 and ELECTRA seem to have a less entity-type-oriented embedding space than BERT. As a result, we have chosen BERT embeddings for OWNER. ERNIE and RoBERTa would have also been valid choices.

# B Unsupervised Confusion Matrix

A useful tool to qualitatively analyze the performance of a classifier is the confusion matrix [56]. Each row of the confusion matrix represents the instances in an actual class (e.g., entity type), and each column represents the instances in a predicted class. Thus, the matrix's diagonal shows correctly predicted instances, and the lower and upper triangles display the errors (also called confusions).

However, when implementing models based on unsupervised approaches (typically clustering), where classes are not predefined, a confusion matrix is harder to interpret. Indeed, contrary to the supervised case, there is no direct link between the class IDs and the cluster IDs (meaning the first class does not necessarily correspond to the first cluster), so there is no clear interpretable diagonal by default. To improve the readability and interoperability of a clustering confusion matrix, rows and columns must be reordered to display a diagonal and group the confusions together.

This appendix details the method employed to reorder the rows and columns. We take the example of the first figure of Fig. 9 (OWNER trained on Pile-NER and tested on Science). The initial confusion matrix, without processing, is displayed in Fig. 10 (a). It resembles a starry sky more than a confusion matrix and is nearly impossible to interpret.

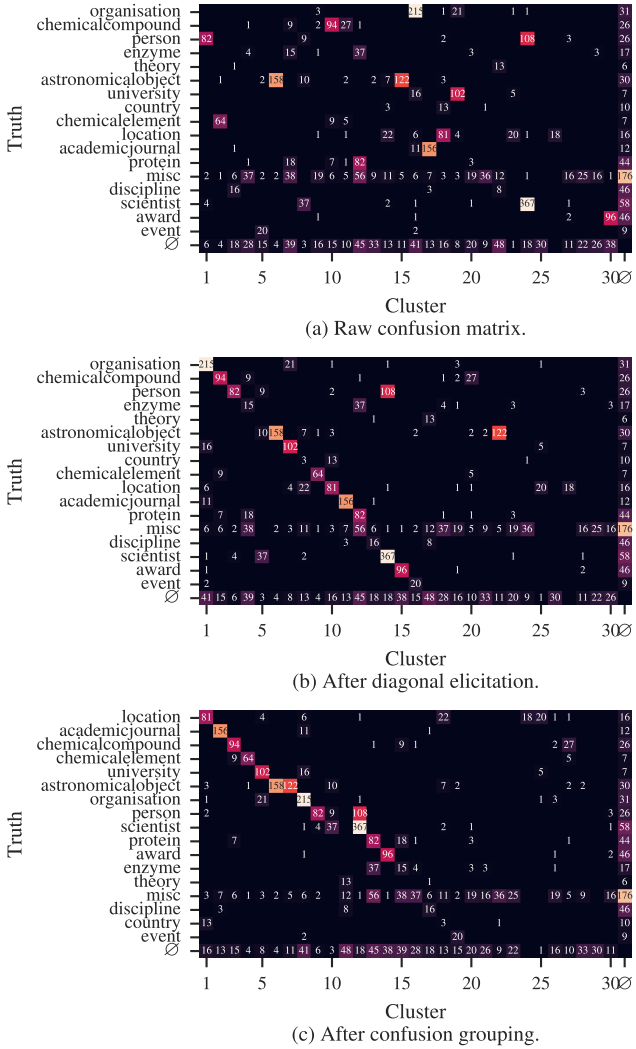
**Diagonal Elicitation** The first step is to find a diagonal in the confusion matrix. In a supervised scenario, if the model performs correctly, most instances are in the diagonal as the model correctly predicts them. By extension, we want to reorder the axes so that the unsupervised confusion matrix shows a clear diagonal: we want to find the "main" cluster corresponding to each class. For instance, in Fig. 10 (a), most instances of *organization* are in cluster 16, most *chemicalcompound* entities are in cluster 11, ...

This can be formulated as: "reorganizing the rows and columns so that the diagonal of the matrix is of maximal sum". This corresponds to an assignment problem (except that the canonical problem involves minimizing the sum). We solve this assignment problem using a modified version of the Jonker-Volgenant algorithm[14] [36, 10].

The resulting confusion matrix is displayed in Fig. 10 (b). It displays a clear diagonal that is much more interpretable than the initial confusion matrix. Nevertheless, some important values outside the diagonal are still scattered (e.g., *person*/cluster 14, *astronomicalobject*/cluster 22).

**Confusion Grouping** The second step aims to bring major confusions closer to make the matrix readable. An ideal confusion matrix is a band matrix, that is, a sparse matrix

[14] We employ the SciPy implementation of the algorithm.




(a) Raw confusion matrix.




(b) After diagonal elicitation.




(c) After confusion grouping.

Figure 10: Reordering of the unsupervised confusion matrix of OWNER for NER trained on Pile-NER and tested on Science.

where the non-zero entries are confined to a diagonal band. We propose implementing the reverse Cutthill–McKee algorithm[15] [13, 28], which aims to permute a sparse matrix into a band matrix with a small bandwidth. In practice, not all non-zero values are interesting (some represent noise or very rare edge cases), so we propose fixing a threshold (1 % of the total instances). Below this threshold, the value is not considered when reordering axes.

We obtain the final confusion matrix of Fig. 10 (c). We can see that the major confusions are now grouped closer (e.g., *person* and *scientist*, *protein* and *enzyme*, *university* and *organization*).

As a side note, the first diagonal elicitation step is optional, as the reverse Cuthill–McKee algorithm produces a band matrix (that is, with a diagonal). We have found, in practice, that the first diagonal elicitation step helped to produce a diagonal with the maximum sum, thus leading to a clearer interpretation.

# References

[1] Tatsuya Aoyama, Shabnam Behzad, Luke Gessler, Lauren Levine, Jessica Lin, Yang Janet Liu, Siyao Peng, Yilun Zhu, and Amir Zeldes. GENTLE: A Genre-Diverse Multilayer Challenge Set for English NLP and Linguistic Evaluation. In Jakob Prange and Annemarie Friedrich, editors, *Proceedings of the 17th Linguistic Annotation Workshop (LAW-XVII)*, pages 166–178, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.law-1.17. URL https://aclanthology.org/2023.law-1.17.

[2] Amit Bagga and Breck Baldwin. Algorithms for scoring coreference chain. In *Proceedings of the 1st International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*, pages 563–566, Granada, Spain, 1998. European Language Resources Association.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

[4] Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. Large Scale Online Learning of Image Similarity Through Ranking. *The Journal of Machine Learning Research*, 11:1109–1135, March 2010. ISSN 1532-4435. URL https://www.jmlr.org/papers/volume11/chechik10a/chechik10a.pdf.

[15] Following the SciPy implementation.

[5] Changyou Chen, Jianyi Zhang, Yi Xu, Liqun Chen, Jiali Duan, Yiran Chen, Son Tran, Belinda Zeng, and Trishul Chilimbi. Why do We Need Large Batchsizes in Contrastive Learning? A Gradient-Bias Perspective. *Advances in Neural Information Processing Systems*, 35: 33860–33875, December 2022.

[6] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, and others. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. *URL https://lmsys.org/blog/2023-03-30-vicuna*, 3(5), 2023.

[7] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume I, pages 539–546, San Diego, CA, USA, 2005. IEEE Computer Society. ISBN 0-7695-2372-2. doi: 10.1109/CVPR.2005.202.

[8] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling Instruction-Finetuned Language Models. *Journal of Machine Learning Research*, 25(70):1–53, 2024. ISSN 1533-7928. URL http://jmlr.org/papers/v25/23-0870.html.

[9] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *Proceedings of the Seventh International Conference on Learning Representations*, New Orleans, Louisiana, United States, September 2019. URL https://openreview.net/forum?id=r1xMH1BtvB.

[10] David F. Crouse. On implementing 2D rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1679–1696, August 2016. ISSN 1557-9603. doi: 10.1109/TAES.2016.140952. URL https://ieeexplore.ieee.org/document/7738348. Conference Name: IEEE Transactions on Aerospace and Electronic Systems.

[11] Alessandro Cucchiarelli and Paola Velardi. Unsupervised Named Entity Recognition Using Syntactic and Semantic Contextual Evidence. *Computational Linguistics*, 27(1):123–131, March 2001. ISSN 0891-2017. doi: 10.1162/089120101300346822. URL https://doi.org/10.1162/089120101300346822.

[12] Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. Template-Based Named Entity Recognition Using BART. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1835–1845, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.161. URL https://aclanthology.org/2021.findings-acl.161.

[13] E. Cuthill and J. McKee. Reducing the bandwidth of sparse symmetric matrices. In *Proceedings of the 1969 24th national conference*, ACM '69, pages 157–172, New York, NY, USA, 1969. Association for Computing Machinery. ISBN 978-1-4503-7493-4. doi: 10.1145/800195.805928. URL https://dl.acm.org/doi/10.1145/800195.805928.

[14] Sarkar Snigdha Sarathi Das, Arzoo Katiyar, Rebecca Passonneau, and Rui Zhang. CONTaiNER: Few-Shot Named Entity Recognition via Contrastive Learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6338–6353, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.439. URL https://aclanthology.org/2022.acl-long.439.

[15] Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. Results of the WNUT2017 Shared Task on Novel and Emerging Entity Recognition. In Leon Derczynski, Wei Xu, Alan Ritter, and Tim Baldwin, editors, *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4418. URL https://aclanthology.org/W17-4418.

[16] Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 4171–4186, Stroudsburg, PA, USA, 2019. Association for Computational Linguistics. ISBN 978-1-950737-13-0. doi: 10.18653/V1/N19-1423.

[17] Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Haitao Zheng, and Maosong Sun. OpenPrompt: An Open-source Framework for Prompt-learning. In Valerio Basile, Zornitsa Kozareva, and Sanja Stajner, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 105–113, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-demo.10. URL https://aclanthology.org/2022.acl-demo.10.

[18] Yuyang Ding, Juntao Li, Pinzheng Wang, Zecheng Tang, Bowen Yan, and Min Zhang. Rethinking Negative Instances for Generative Named Entity Recognition, June 2024. URL http://arxiv.org/abs/2402.16602. arXiv:2402.16602 [cs].

[19] Vladimir Dobrovolskii. Word-Level Coreference Resolution. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages

7670–7675, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.605. URL https://aclanthology.org/2021.emnlp-main.605.

[20] Guanting Dong, Zechen Wang, Jinxu Zhao, Gang Zhao, Daichi Guo, Dayuan Fu, Tingfeng Hui, Chen Zeng, Keqing He, Xuefeng Li, Liwen Wang, Xinyue Cui, and Weiran Xu. A Multi-Task Semantic Decomposition Framework with Task-specific Pre-training for Few-Shot NER. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, pages 430–440, New York, NY, USA, October 2023. Association for Computing Machinery. doi: 10.1145/3583780.3614766. URL https://doi.org/10.1145/3583780.3614766.

[21] Bin Duan, Shusen Wang, Xingxian Liu, and Yajing Xu. Cluster-aware Pseudo-Labeling for Supervised Open Relation Extraction. In Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na, editors, *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1834–1841, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics. URL https://aclanthology.org/2022.coling-1.158.

[22] Richard Duda, Peter Hart, and David Stork. *Pattern classification*. Wiley Hoboken, 2000. ISBN 0-471-05669-3.

[23] Jinyuan Fang, Xiaobin Wang, Zaiqiao Meng, Pengjun Xie, Fei Huang, and Yong Jiang. MANNER: A Variational Memory-Augmented Model for Cross Domain Few-Shot Named Entity Recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4261–4276, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.234. URL https://aclanthology.org/2023.acl-long.234.

[24] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1126–1135. Proceedings of Machine Learning Research, July 2017. URL https://proceedings.mlr.press/v70/finn17a.html. ISSN: 2640-3498.

[25] PENG Fuchun. Accurate information extraction from research papers using conditional random fields. In *Proc. of HLT/NAACL, 2004*, 2004.

[26] 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. The Pile: An 800GB Dataset of Diverse Text for Language Modeling, December 2020. URL http://arxiv.org/abs/2101.00027. arXiv:2101.00027 [cs].

[27] Pierre-Yves Genest, Pierre-Edouard Portier, Előd Egyed-Zsigmond, and Laurent-Walter Goix. PromptORE - A Novel Approach Towards Fully Unsupervised Relation Extraction. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*, Atlanta, USA, October 2022. Association for Computing Machinery. doi: 10.1145/3511808.3557422. URL https://hal.science/hal-03858264.

[28] Alan George and Joseph W Liu. *Computer solution of large sparse positive definite*. Prentice Hall Professional Technical Reference, 1981.

[29] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. In *Proceedings of the Ninth International Conference on Learning Representations*, Online, January 2021. URL https://openreview.net/forum?id=XPZIaotutsD.

[30] Pengcheng He, Jianfeng Gao, and Weizhu Chen. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. In *Proceedings of the Eleventh International Conference on Learning Representations*, Kigali, Rwanda, 2023. URL https://openreview.net/forum?id=sE7-XhLxHA.

[31] Yucheng Huang, Kai He, Yige Wang, Xianli Zhang, Tieliang Gong, Rui Mao, and Chen Li. COPNER: Contrastive Learning with Prompt Guiding for Few-shot Named Entity Recognition. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2515–2527, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics. URL https://aclanthology.org/2022.coling-1.222.

[32] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, December 1985. doi: 10.1007/BF01908075. URL https://link.springer.com/article/10.1007/BF01908075. Publisher: Springer.

[33] Andrea Iovine, Anjie Fang, Besnik Fetahu, Oleg Rokhlenko, and Shervin Malmasi. CycleNER: An Unsupervised Training Approach for Named Entity Recognition. In *Proceedings of the ACM Web Conference 2022*, WWW '22, pages 2916–2924, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 978-1-4503-9096-5. doi: 10.1145/3485447.3512012. URL https://dl.acm.org/doi/10.1145/3485447.3512012.

[34] Chen Jia, Xiaobo Liang, and Yue Zhang. Cross-Domain NER using Cross-Domain Language Modeling. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2464–2474, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1236. URL https://aclanthology.org/P19-1236.

[35] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B, October 2023. URL http://arxiv.org/abs/2310.06825. arXiv:2310.06825 [cs].

[36] Roy Jonker and Ton Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. In *DGOR/NSOR: Papers of the 16th annual meeting of DGOR in cooperation with NSOR/vorträge der 16. Jahrestagung der DGOR zusammen mit der NSOR*, pages 622–622. Springer, 1988.

[37] Jin-Dong Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. GENIA corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl_1):i180–i182, 2003. ISBN: 1367-4811 Publisher: Oxford University Press.

[38] Diederik Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations, Conference Track*, San Diego, CA, USA, 2015. URL http://arxiv.org/abs/1412.6980.

[39] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In Eduardo Blanco and Wei Lu, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-2012. URL https://aclanthology.org/D18-2012.

[40] Aman Kumar and Binil Starly. "FabNER": information extraction from manufacturing process science domain literature using named entity recognition. *Journal of Intelligent Manufacturing*, 33(8):2393–2407, December 2022. ISSN 1572-8145. doi: 10.1007/s10845-021-01807-x. URL https://doi.org/10.1007/s10845-021-01807-x.

[41] Zixuan Li, Yutao Zeng, Yuxin Zuo, Weicheng Ren, Wenxuan Liu, Miao Su, Yucan Guo, Yantao Liu, Xiang Li, Zhilei Hu, Long Bai, Wei Li, Yidan Liu, Pan Yang, Xiaolong Jin, Jiafeng Guo, and Xueqi Cheng. KnowCoder: Coding Structured Knowledge into LLMs for Universal Information Extraction, March 2024. URL http://arxiv.org/abs/2403.07969. arXiv:2403.07969 [cs].

[42] Jingjing Liu, Panupong Pasupat, Scott Cyphers, and Jim Glass. Asgard: A portable architecture for multilingual dialogue systems. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8386–8390, May 2013. doi: 10.1109/ICASSP.2013.6639301. URL https://ieeexplore.ieee.org/abstract/document/6639301. ISSN: 2379-190X.

[43] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach, July 2019. URL https://arxiv.org/abs/1907.11692v1.

[44] Zihan Liu, Genta Indra Winata, and Pascale Fung. Zero-Resource Cross-Domain Named Entity Recognition. In Spandana Gella, Johannes Welbl, Marek Rei, Fabio Petroni, Patrick Lewis, Emma Strubell, Minjoon Seo, and Hannaneh Hajishirzi, editors, *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 1–6, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.repl4nlp-1.1. URL https://aclanthology.org/2020.repl4nlp-1.1.

[45] Zihan Liu, Yan Xu, Tiezheng Yu, Wenliang Dai, Ziwei Ji, Samuel Cahyawijaya, Andrea Madotto, and Pascale Fung. CrossNER: Evaluating Cross-Domain Named Entity Recognition. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, volume 15, pages 13452–13460. Association for the Advancement of Artificial Intelligence, December 2021. ISBN 978-1-71383-597-4. doi: 10.48550/arxiv.2012.04373. URL https://arxiv.org/abs/2012.04373v2.

[46] Stuart Lloyd. Least squares quantization in PCM. *Technical Report RR-5497*, 1957.

[47] Jie Lou, Yaojie Lu, Dai Dai, Wei Jia, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. Universal Information Extraction as Unified Semantic Matching. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37 (11):13318–13326, June 2023. ISSN 2374-3468. doi: 10.1609/aaai.v37i11.26563. URL https://ojs.aaai.org/index.php/AAAI/article/view/26563. Number: 11.

[48] Ying Luo, Hai Zhao, and Junlang Zhan. Named Entity Recognition Only from Word Embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8995–9005, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.723. URL https://aclanthology.org/2020.emnlp-main.723.

[49] Tingting Ma, Huiqiang Jiang, Qianhui Wu, Tiejun Zhao, and Chin-Yew Lin. Decomposed Meta-Learning for Few-Shot Named Entity Recognition. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1584–1596, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.124. URL https://aclanthology.org/2022.findings-acl.124.

[50] James MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley symposium on mathematical statistics and probability*, pages 281–297, Berkeley, California, United States, 1967. University of California Press.

[51] Aniruddha Mahapatra, Sharmila Reddy Nangi, Aparna Garimella, and Anandhavelu Natarajan. Entity Extraction in Low Resource Domains with Selective Pre-training of Large Language Models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language*

*Processing*, pages 942–951, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.emnlp-main.61.

[52] David Nadeau, Peter Turney, and Stan Matwin. Unsupervised Named-Entity Recognition: Generating Gazetteers and Resolving Ambiguity. In Luc Lamontagne and Mario Marchand, editors, *Advances in Artificial Intelligence*, Lecture Notes in Computer Science, pages 266–277, Berlin, Heidelberg, 2006. Springer. ISBN 978-3-540-34630-2. doi: 10.1007/11766247_23.

[53] Adeiza James Onumanyi, Daisy Nkele Molokomme, Sherrin John Isaac, and Adnan M. Abu-Mahfouz. AutoElbow: An Automatic Elbow Detection Method for Estimating the Number of Clusters in a Dataset. *Applied Sciences*, 12(15):7515, January 2022. ISSN 2076-3417. doi: 10.3390/app12157515. URL https://www.mdpi.com/2076-3417/12/15/7515. Number: 15 Publisher: Multidisciplinary Digital Publishing Institute.

[54] OpenAI. ChatGPT — Release Notes | OpenAI Help Center, 2022. URL https://help.openai.com/en/articles/6825453-chatgpt-release-notes.

[55] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Lukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Lukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, C. J. Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. GPT-4 Technical Report, March 2024. URL http://arxiv.org/abs/2303.08774. arXiv:2303.08774 [cs].

[56] Karl Pearson and John Blakeman. *On the theory of contingency and its relation to association and normal correlation*, volume XIII of *Mathematical contributions to the theory of evolution*. Dulau and Co., London, 1904.

[57] Qi Peng, Changmeng Zheng, Yi Cai, Tao Wang, Haoran Xie, and Qing Li. Unsupervised cross-domain named entity recognition using entity-aware adversarial training. *Neural Networks*, 138:68–77, June 2021. ISSN 0893-6080. doi: 10.1016/j.neunet.2020.12.027. URL https://www.sciencedirect.com/science/article/pii/S0893608020304524.

[58] Ethan Perez, Douwe Kiela, and Kyunghyun Cho. True Few-Shot Learning with Language Models. In *Advances in Neural Information Processing Systems*, volume 34, pages

11054–11070. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/5c04925674920eb58467fb52ce4ef728-Abstract.html.

[59] Pushpankar Kumar Pushp and Muktabh Mayank Srivastava. Train Once, Test Anywhere: Zero-Shot Learning for Text Classification, December 2017. URL http://arxiv.org/abs/1712.05972. arXiv:1712.05972 [cs].

[60] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. ISSN 1533-7928. URL http://jmlr.org/papers/v21/20-074.html.

[61] Simone Romano, Nguyen Xuan Vinh, James Bailey, and Karin Verspoor. Adjusting for chance clustering comparison measures. *The Journal of Machine Learning Research*, 17(1):4635–4666, January 2016. ISSN 1532-4435. URL https://dl.acm.org/doi/abs/10.5555/2946645.3007087.

[62] Andrew Rosenberg and Julia Hirschberg. V-Measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 410–420, Prague, Czech Republic, 2007. Association for Computational Linguistics.

[63] Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. Code Llama: Open Foundation Models for Code, January 2024. URL http://arxiv.org/abs/2308.12950. arXiv:2308.12950 [cs].

[64] Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. GoLLIE: Annotation Guidelines improve Zero-Shot Information-Extraction. In *Proceedings of the Twelfth International Conference on Learning Representations*, Vienna, Austria, January 2024. URL https://openreview.net/forum?id=Y3wpuxd7u9.

[65] Gideon Schwarz. Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2):461–464, 1978. ISSN 0090-5364. URL https://www.jstor.org/stable/2958889. Publisher: Institute of Mathematical Statistics.

[66] Claude Elwood Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27 (3):379–423, July 1948. ISSN 0005-8580. doi: 10.1002/j.1538-7305.1948.tb01338.x. URL https://ieeexplore.ieee.org/abstract/document/6773024.

[67] Dan Shen, Jie Zhang, Jian Su, Guodong Zhou, and Chew Lim Tan. Multi-criteria-based active learning for named entity recognition. In *Proceedings of the 42nd annual meeting of the association for computational linguistics (ACL-04)*, pages 589–596, 2004.

[68] Yongliang Shen, Zeqi Tan, Shuhui Wu, Wenqi Zhang, Rongsheng Zhang, Yadong Xi, Weiming Lu, and Yueting Zhuang. 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, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.698. URL https://aclanthology.org/2023.acl-long.698.

[69] Taylor Shin, Yasaman Razeghi, Robert L. Logan, Eric Wallace, and Sameer Singh. AUTOPROMPT: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 4222–4235, Online, October 2020. Association for Computational Linguistics. ISBN 978-1-952148-60-6. doi: 10.18653/v1/2020.emnlp-main.346. URL https://arxiv.org/abs/2010.15980v2.

[70] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 2017-Decem, pages 4078–4088. Neural information processing systems foundation, March 2017. doi: 10.48550/arxiv.1703.05175. URL https://arxiv.org/abs/1703.05175v2.

[71] Douglas Steinley. Properties of the Hubert-Arabie adjusted Rand index. *Psychological Methods*, 9(3): 386–396, September 2004. doi: 10.1037/1082-989X.9.3.386. URL https://pubmed.ncbi.nlm.nih.gov/15355155/. Publisher: Psychol Methods.

[72] Amber Stubbs and Ozlem Uzuner. Annotating longitudinal clinical narratives for de-identification: The 2014 i2b2/UTHealth corpus. *Journal of Biomedical Informatics*, 58 Suppl(Suppl):S20–S29, December 2015. ISSN 1532-0480. doi: 10.1016/j.jbi.2015.07.020.

[73] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. ERNIE: Enhanced Representation through Knowledge Integration, April 2019. URL http://arxiv.org/abs/1904.09223. arXiv:1904.09223 [cs].

[74] Erik Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. In *Proceedings of the 7th conference on Natural language learning*, volume Volume 4 of *CONLL '03*, pages 142–147, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1119176.1119195. URL https://dl.acm.org/doi/10.3115/1119176.1119195.

[75] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar,

Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and Efficient Foundation Language Models, February 2023. URL http://arxiv.org/abs/2302.13971. arXiv:2302.13971 [cs].

[76] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11), 2008.

[77] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *Proceedings of the 26th annual international conference on machine learning*, pages 1073–1080, 2009.

[78] Jianing Wang, Chengyu Wang, Chuanqi Tan, Minghui Qiu, Songfang Huang, Jun Huang, and Ming Gao. SpanProto: A Two-stage Span-based Prototypical Network for Few-shot Named Entity Recognition. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3466–3476, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.227. URL https://aclanthology.org/2022.emnlp-main.227.

[79] Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. GPT-NER: Named Entity Recognition via Large Language Models, May 2023. URL http://arxiv.org/abs/2304.10428. arXiv:2304.10428 [cs].

[80] 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. InstructUIE: Multi-task Instruction Tuning for Unified Information Extraction, April 2023. URL http://arxiv.org/abs/2304.08085. arXiv:2304.08085 [cs].

[81] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *Proceedings of the 11th International Conference on Learning Representations*, Kigali, Rwanda, March 2023. doi: 10.48550/arXiv.2203.11171. URL http://arxiv.org/abs/2203.11171.

[82] Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, Yong Jiang, and Wenjuan Han. Zero-Shot Information Extraction via Chatting with ChatGPT, February 2023. URL http://arxiv.org/abs/2302.10205. arXiv:2302.10205 [cs].

[83] Tingyu Xie, Qi Li, Yan Zhang, Zuozhu Liu, and Hongwei Wang. Self-Improving for Zero-Shot Named Entity Recognition with Large Language Models. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 583–593, Mexico City, Mexico, June 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.naacl-short.49.

[84] Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, Jie Zhou, Siming Chen, Tao Gui, Qi Zhang, and Xuanjing Huang. A Comprehensive Capability Analysis of GPT-3 and GPT-3.5 Series Models, December 2023. URL http://arxiv.org/abs/2303.10420. arXiv:2303.10420 [cs].

[85] Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. Named Entity Recognition as Structured Span Prediction. In Wenjuan Han, Zilong Zheng, Zhouhan Lin, Lifeng Jin, Yikang Shen, Yoon Kim, and Kewei Tu, editors, *Proceedings of the Workshop on Unimodal and Multimodal Induction of Linguistic Structures (UMIoS)*, pages 1–10, Abu Dhabi, United Arab Emirates (Hybrid), December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.umios-1.1. URL https://aclanthology.org/2022.umios-1.1.

[86] Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. GLiNER: Generalist Model for Named Entity Recognition using Bidirectional Transformer, November 2023. URL http://arxiv.org/abs/2311.08526. arXiv:2311.08526 [cs].

[87] Amir Zeldes. The GUM corpus: creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612, September 2017. ISSN 1574-0218. doi: 10.1007/s10579-016-9343-x. URL https://doi.org/10.1007/s10579-016-9343-x.

[88] Mozhi Zhang, Hang Yan, Yaqian Zhou, and Xipeng Qiu. PromptNER: A Prompting Method for Few-shot Named Entity Recognition via k Nearest Neighbor Search, May 2023. URL http://arxiv.org/abs/2305.12217. arXiv:2305.12217 [cs].

[89] Jun Zhao, Tao Gui, Qi Zhang, and Yaqian Zhou. A Relation-Oriented Clustering Method for Open Relation Extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9707–9718, Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics.

[90] Jun Zhao, Yongxin Zhang, Qi Zhang, Tao Gui, Zhongyu Wei, Minlong Peng, and Mingming Sun. Actively Supervised Clustering for Open Relation Extraction. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4985–4997, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.273. URL https://aclanthology.org/2023.acl-long.273.

[91] Zexuan Zhong and Danqi Chen. A Frustratingly Easy Approach for Entity and Relation Extraction. In *Proceedings of the 2021 Annual Conference of the North American Chapter of the Association for Computational Linguistics*,

pages 50–61, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.5.

[92] Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. UniversalNER: Targeted Distillation from Large Language Models for Open Named Entity Recognition. In *Proceedings of the Twelfth International Conference on Learning Representations*, Vienna, Austria, 2024. URL https://openreview.net/forum?id=r65xfUb76p.