IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 14 February 2025, accepted 12 March 2025, date of publication 17 March 2025, date of current version 26 March 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3552122

RESEARCH ARTICLE

OWNER – Toward 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, LIRIS, UMR5205, Université Claude Bernard Lyon 1, 69621 Villeurbanne, France
[3]Caisse d'Epargne Rhône Alpes, 69003 Lyon, France

Corresponding author: Pierre-Yves Genest (pygenest@alteca.fr)

This work was supported by Alteca and the French Association for Research and Technology (ANRT) through Convention industrielle de formation par la recherche [Industrial Agreement for Training through Research (CIFRE)] Ph.D. Fellowship under Grant 2021/0851.

ABSTRACT Named Entity Recognition (NER) is a crucial task in Natural Language Processing (NLP), traditionally addressed through supervised learning, which requires extensive annotated corpora. This requirement poses challenges, particularly in specialized domains with limited labeled data. In response, the field has shifted towards lower-resource approaches, such as few-shot and zero-shot learning, which reduce the dependency on annotated data. However, even zero-shot models require prior knowledge of entity types, limiting their applicability in exploratory scenarios. In this context, we introduce OWNER, our unsupervised and open-world NER model, designed to operate without annotated documents or predefined entity types. OWNER leverages Encoder-only Language Models like BERT to infer and organize entities into dynamic entity types through a two-step process: mention detection and entity typing. Mention detection employs a BIO sequence labeling approach to locate entities, while entity typing uses BERT-based embeddings, refined through contrastive learning, for clustering and naming entity types. This method allows OWNER to automatically identify and structure unknown entity types, offering advantages for exploratory dataset analysis and knowledge graph construction. Our experimental evaluation on 13 domain-specific datasets demonstrates that OWNER surpasses existing LLM-based open-world NER models and remains competitive with more supervised and closed-world zero-shot models. OWNER's architecture provides a lightweight, easily deployable solution that advances the state of the art in unsupervised and open-world NER. The source code of OWNER is publicly available at https://github.com/alteca/OWNER, facilitating future research in this domain.

INDEX TERMS Named entity recognition, open information extraction, open-world named entity recognition, unsupervised named entity recognition.

I. INTRODUCTION

Named Entity Recognition (NER) is a fundamental NLP task that identifies entities in text and classifies them into specific entity types. Traditionally, NER has been treated as a supervised task [1], [2]. It poses challenges in specific domains, such as scientific and biomedical fields, where large labeled corpora are scarce.

Consequently, there is growing interest in low-resource approaches [3], especially with the rise of Encoder-only Language Models (encoders) like BERT [4]. Notably, few-shot models [5], [6], which require only a small set of annotated documents, achieve impressive performance levels given their minimal supervision. However, these approaches still need some annotated documents.

To further reduce the need for supervision, zero-shot NERs have emerged. These models do not need annotated documents but only require the specification of expected entity types, sometimes including their names, descriptions,

The associate editor coordinating the review of this manuscript and approving it for publication was Dominik Strzalka.

and a few examples. Recent models often transfer knowledge from a source domain $\mathcal{D}_S$, where annotated data is abundant, to a target domain $\mathcal{D}_T$, which lacks labeled documents [7], [8]. With the advent of Large Language Models (LLMs) [9], these zero-shot approaches have achieved remarkable results. For example, the zero-shot model GliNER [10] surpasses the fully supervised generalist NER spaCy, which was developed a few years earlier [11]. Nonetheless, zero-shot approaches still require knowledge of the list of entity types.

Unsupervised and open-world approaches have been proposed to address this challenge. These approaches do not require prior knowledge of entity types, and they automatically organize extracted entities into dynamic and semantically meaningful types that are not predefined. In essence, they represent the logical progression beyond zero-shot models in the effort to reduce supervision. Formally, unsupervised and open-world models 1) do not utilize annotated documents from the target domain and 2) do not require prior knowledge of entity types, including their number. This setting represents one of the lowest levels of supervision for information extraction. It is well-established in the information extraction domain, particularly in relation extraction, with consistent research spanning over twenty years [12], [13], [14], [15], [16]. However, our literature review indicates that this setting has been relatively unexplored for NER. To the best of our knowledge, the most recent research dates back to 2020 [17] and lacks reproducibility due to the absence of source code and implementation details.

This article seeks to revisit the unsupervised and open-world setting in light of recent advancements in NLP. Our primary research question is:

> *How can we extract named entities in an unsupervised and open-world setting, specifically 1) without annotated documents from the target domain and 2) without prior knowledge of the entity types to be identified?*

At first glance, one might question the relevance of this study, given the significant advancements made with zero-shot models. However, we believe there are compelling motivations.

Open-world and unsupervised NER is advantageous in exploratory contexts. Its ability to automatically identify and structure entity types is beneficial for understanding and analyzing information in an unknown dataset. Specifically, the model can identify entity types that were not initially anticipated,[1] providing a more comprehensive view of the dataset than classical closed-world approaches.

This advantage extends to the task of constructing knowledge graphs from documents. The capability to self-structure and detect novelties facilitates the creation of a more complete knowledge graph compared to traditional closed-world methods.

We present **OWNER**, our "Unsupervised **O**pen-**W**orld **N**amed **E**ntity **R**ecognition" model. OWNER is an unsupervised and open-world model that infers and structures entities into non-predefined entity types. Similar to zero-shot models, OWNER uses annotated data from a source domain $\mathcal{D}_S$, which can be either manually or automatically annotated documents, to learn named entity recognition and transfer it to a target domain $\mathcal{D}_T$, where no annotated documents are available. Inspired by the recent successes of the non-LLM GliNER [10], we explore using Encoder-only Language Model embeddings, such as BERT [4], to make predictions, aiming to create a lightweight and easily deployable model.

We divide NER into two subtasks: 1) mention detection, which locates entities, and 2) entity typing, which classifies the extracted entities into types. For mention detection, we implement a BIO sequence labeling NER (see Sect. II), anticipating it will generalize better in specific domains than more complex architectures [6].[2]

For entity typing, we employ an entity embedding approach inspired by few-shot and zero-shot methods. The goal is to compute a vector representation for each entity that is characteristic of its entity type. These embeddings are generated using BERT prompting and are then clustered to identify entity types. Clustering relies on k-means, complemented by cluster estimation using the Bayesian Information Criteria [18] and ternary search. To better isolate entity types in the target domain, we implement an embedding refinement approach based on contrastive learning. Finally, we propose deriving entity cluster names from BERT embeddings using its Masked Language Modeling (MLM) head.

This simple yet innovative architecture empirically outperforms LLM-based open-world NER and competes with closed-world zero-shot models. We expect OWNER to serve as 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 $\mathcal{D}_T$ without requiring annotations in $\mathcal{D}_T$, without prior knowledge of the target entity types $\mathcal{T}_T$, or their number $|\mathcal{T}_T|$.
- For entity typing, we introduce a novel architecture that includes 1) prompt-based entity encoding, 2) unsupervised clustering to classify entities into types, and 3) contrastive learning to more precisely identify entity types.
- Experimental results on 13 domain-specific datasets demonstrate that OWNER surpasses LLM-based open-world and unsupervised NER, achieving a 2% – 18% improvement in AMI, and performs comparably to state-of-the-art closed-world zero-shot models.
- Qualitative analysis shows that OWNER organizes entities into semantically coherent clusters closely aligned with true entity types, and the derived names accurately describe the content of these clusters.

[1] We experimentally observe this property in Sects. V-B and V-F.

[2] Experimental results confirm this hypothesis (see Sect. V-C).

II. RELATED WORK

Before starting this section, we clarify the mathematical notations, which are summarized in Table 11. NER analyzes documents represented as $\boldsymbol{X} = [x_0, x_1, \ldots, x_{|\boldsymbol{X}|-1}]$. Each $x_i \in \boldsymbol{X}$ is a token.[3] The objective is to extract entities $\boldsymbol{e} = [x_i, \ldots, x_j]$ and classify their entity type t. The set of entity types is denoted by $\mathcal{T}$. We assume access to labeled documents from a source domain $\mathcal{D}_S$ with its set of entity types $\mathcal{T}_S$. The goal is to generalize to a target domain $\mathcal{D}_T$, which is associated with the entity types $\mathcal{T}_T$ and lacks annotated data. Closed-world models require knowledge of $\mathcal{T}_T$, including their number, names, and sometimes descriptions, whereas open-world methods such as OWNER do not have access to this information.

A. FEW-SHOT AND ZERO-SHOT NER

As a reminder, few-shot and zero-shot models assume prior knowledge of the target entity types list $\mathcal{T}_T$. Most approaches rely on labeled data from a source domain $\mathcal{D}_S$ to learn and subsequently transfer to a target domain $\mathcal{D}_T$. The domains $\mathcal{D}_S$ and $\mathcal{D}_T$ may differ stylistically (type of text), semantically (topic), or in terms of entity types ($\mathcal{T}_S \neq \mathcal{T}_T$). $\mathcal{D}_S$ can consist of a manually annotated dataset [6], [20], a distantly labeled dataset [21], or a synthetically generated dataset [8], [10], [22].

Recent approaches are categorized into two families: 1) two-stage NER, and 2) one-stage or integrated NER.

Two-stage approaches divide NER into Mention Detection (MD) and Entity Typing (ET) [6], [23], [24]. Mention detection identifies spans of $\boldsymbol{X}$ that are entities, while entity typing classifies the type of each extracted entity. Integrated models combine MD and ET in a single step to reduce cascading errors [8], [20], [21], [25], [26]. In practice, both paradigms achieve state-of-the-art results [6], [25]. Until recently, most approaches used Encoder-only Language Models (encoders) such as BERT [4]. We now observe the increasing use of Large Language Models (LLMs) in these low-resource settings [8], [20], where LLMs have demonstrated significant effectiveness [9].

1) MENTION DETECTION (MD)

Few-shot and zero-shot approaches adopt architectures similar to supervised models for mention detection. They typically implement span-based extractors [5], [10], [27], although BIO sequence labeling remains in use [6], [24]. These extractors are trained in a supervised manner on entities from $\mathcal{D}_S$. The challenge lies in transferring the learned patterns from $\mathcal{D}_S$ to entities in $\mathcal{D}_T$. BIO sequence labeling classifies each token x in $\boldsymbol{X}$ as either B (beginning of an entity), I (inside an entity), or O (outside any entity). A decoding algorithm then reconstructs the entity boundaries based on the predicted classes. Greedy algorithms are particularly used with recent language models [6]. Conditional random fields [28] can also be employed to enhance decoding. The primary limitation of the BIO approach is its inability to predict nested entities. This limitation serves as the main motivation for employing span-based extractors.

In general, span-based extractors evaluate each potential span in $\boldsymbol{X}$ to identify true entities [2], [5]. They achieve this by computing vector representations for the *start of span* and *end of span*, typically using embeddings of the first and last tokens of the candidate span. Zhong et al. [2] concatenate the *start* and *end* embeddings and apply them in a perceptron to score the candidate span. Wang et al. [5] replace the perceptron with bilinear layers, enabling more efficient computations compared to Zhong et al. [2]. Span-based approaches face challenges with the quadratic number of possible spans, making candidate span scoring costly for lengthy documents. Dobrovolskii [29] addresses this issue with a hybrid approach. Initially, each word in $\boldsymbol{X}$ is classified as an entity head or not. An entity head is the main word of an entity, considered by Dobrovolskii as the root of the entity's syntactic subtree. This method reduces the quadratic span complexity to a linear (word) complexity. Once entity heads are identified, a convolutional neural network determines the boundaries of each entity. Finally, Zaratiana et al. [30] propose adapting conditional random fields for span-based extractors to ensure non-overlapping spans.

2) ENTITY TYPING (ET)

The general principle involves computing vector representations of extracted entities (entity embeddings) and comparing them to those of the exemplars (few-shot) or the target entity type names or descriptions (zero-shot and few-shot). Zhang et al. [23] propose using k-nearest neighbors with few-shot exemplars to identify the entity type. Prototypical networks [31] are commonly preferred for classifying entities [5], [6], [27]. These networks compute entity-type prototypes using the exemplars.

Entity embeddings are derived by aggregating the encoder embeddings of individual tokens forming the entity in the case of a BIO extractor [6] or by employing the span representation generated by the span extractor [5]. Shen et al. [21] and Ding et al. [32] investigate prompting techniques with BERT (using the `[MASK]` token) to generate entity embeddings.

Meta-learning [33] is utilized to enhance transfer learning efficacy [24]. The approach involves generating numerous few-shot episodes using annotated data from $\mathcal{D}_S$; each episode includes a subset of $\mathcal{T}_S$, randomly selected few-shot exemplars associated with $\mathcal{T}_S$, and test documents to evaluate performance. The model is then trained on these episodes to optimize transfer in the fewest fine-tuning steps possible, thus the term meta-learning. This method allows effective fine-tuning even on limited few-shot exemplars by enabling the model to converge quickly and reliably.

Finally, Liu et al. [3] and Mahapatra et al. [34] explore adapting encoder embeddings to the target domain. They

[3] A token can be a word, part of a word, or punctuation as defined by SentencePiece [19].

use large amounts of unannotated documents from $\mathcal{D}_T$ and fine-tune BERT weights through a masked language modeling task. Empirically, they observe a correlation between decreased perplexity and increased NER performance. Mahapatra et al. [34] reduce training time required for domain adaptation by filtering unannotated documents from $\mathcal{D}_T$ to retain those more aligned with the actual documents where entities need to be extracted.

3) LARGE LANGUAGE MODELS

Recently, LLMs [35], [36] have been successfully applied to few-shot and zero-shot NER, achieving state-of-the-art results in the zero-shot setting. The proposed methods fall into three categories: raw prompting, LLM fine-tuning, and LLM distillation.

a: RAW PROMPTING

First, raw prompting achieves impressive results compared to previous works [37], [38], [39]. Wang et al. [37] and Ye et al. [38] require few-shot exemplars to specify the output format. Wei et al. [39] (ChatIE) propose a multi-turn framework that operates in a zero-shot setting without the need for exemplars. Surprisingly, they reverse the usual order of MD and ET steps. They first ask the LLM which entity types are present in the document, given a predefined list of entity types. In subsequent turns, they inquire about the entities associated with each entity type. Xie et al. [40], [41] propose automatically generating few-shot instances using GPT-3.5 [42] and refining them with an ensemble method, involving multiple generations with temperature and a voting system to gather entity predictions. They empirically observe that these automatically generated few-shot instances significantly enhance zero-shot performance.

The weaknesses in their works [39], [41] include the multiple turns required to analyze a document, which is costly when using APIs of the largest LLMs, and the complexity of scaling to large datasets with numerous documents and diverse entity types.

b: LLM FINE-TUNING

Sainz et al. [20], Zhou et al. [8], and Wang et al. [43] investigate the fine-tuning of small LLMs [36], [44], [45] on manually or synthetically labeled datasets. This approach creates NER-specialized LLMs that outperform generalist LLMs while being significantly smaller. Zhou et al. [8] annotate documents from the Pile corpus [46] using GPT-3.5, creating the Pile-NER dataset, and fine-tune Vicuna [44] on it. Their UniNER model surpasses GPT-3.5 in a zero-shot context. Furthermore, fine-tuning with extensive synthetic data allows them to specify a custom JSON format that UniNER reliably follows. GoLLIE [20] uses Code-Llama [47] as its backbone and is fine-tuned on manually labeled datasets from the news and biomedical domains. Sainz et al. [20] and Li et al. [48] use 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 that the metadata of description and exemplars positively impacts the performances of GoLLIE and KnowCoder.

In terms of prediction format, most approaches follow a surface form extraction scheme [8], [20], [38], [39], except for GPT-NER [37]. The models output only the entity text, requiring a subsequent algorithm to localize the entity in the document. The output format is generally JSON, but Sainz et al. [20] use Python code, which allows them to elegantly add metadata in comments, such as descriptions and exemplars. GPT-NER [37] proposes a sequence labeling scheme. It asks the LLM to repeat the input document with special markup delimiting the boundaries of entities: `@@` as the opening tag and `##` as the closing tag. This format eliminates the need for a decoding algorithm, as the detected entities are localized within the document by design. However, it is incompatible with a zero-shot setting, as in-context exemplars are required to describe the output format.

c: LLM DISTILLATION

Zaratiana et al. [10] (GliNER) and Ding et al. [49] (GNER) fine-tune encoder embeddings (DeBERTa v3 [50]) or full transformers (Flan-T5 [51]) on GPT-3.5 generated annotations of Pile-NER [8]. GliNER specifically implements a span-based extractor for MD, coupled with a method similar to prototypical networks for ET. They achieve very competitive results compared to the much larger fine-tuned LLMs UniNER [8] and GoLLIE [20]. This model strikes an interesting balance between the flexibility of LLM-based zero-shot NER and the relatively small number of parameters in encoder embeddings.

d: FOUNDATION MODELS

Finally, an interesting orthogonal axis of analysis is the formation of foundation models. Unlike the traditional approach of training specifically for a particular domain, there is an increasing trend towards models pre-trained specifically to perform entity detection or multiple information extraction tasks on any domain.

A foundational work in this area is ERNIE [52], which complements the usual masked language modeling pre-training task with an entity modeling task. In this task, entire entities (comprising one or more tokens) are masked to enhance the contextual association between entities and their context. For example, it is easy to predict *San* in the sentence "The Golden Gate Bridge is located in `[MASK]` Francisco."; however, predicting *San Francisco* from "The Golden Gate Bridge is located in `[MASK] [MASK]`." requires more integration of contextual knowledge. Similarly, Soares et al. [53] propose extending this task to pairs of entities to improve relation embeddings, and Wang et al. [54] extend it to two other information extraction tasks: relation typing and entity typing. These models achieve better results than raw BERT embeddings for information extraction tasks and can serve as drop-in replacements.

With recent advances in LLMs and their impressive generative capabilities, Bogdanov et al. [55] and Peng et al. [56] propose using GPT-4 [22] to annotate data and employing these silver labels to fine-tune BERT embeddings, providing encoders that can be used as backbones. This principle, known as *targeted distillation*, underlies the recent successes of UniNER, GliNER, and GNER, all of which use LLM-generated annotated data to distill smaller models, whether encoders, decoders, or full transformers.

4) ZERO-SHOT MODELS ARE NOT OPEN-WORLD

In the preceding paragraphs, we have observed that the field of zero-shot named entity recognition is bustling with activity, with numerous ideas explored and significant advancements made. However, these approaches fall short of addressing our research question: they require prior knowledge of entity types, including their number, names, and sometimes their descriptions and exemplars. In brief, they are not open-world.

a: CAN ZERO-SHOT MODELS BE DIRECTLY TRANSLATED TO AN OPEN-WORLD SETTING?

The question that naturally arises is: Is it possible to adapt zero-shot models to function within a truly open-world framework? At first glance, one might think that zero-shot approaches can easily translate to an open-world setting, as the two are closely related. However, the reality is more complex. As described in the previous paragraphs, zero-shot approaches can be divided into fine-tuned models and frozen LLM prompting.

Fine-tuned approaches (based on encoders [10], full transformers [49], or LLMs [8]) all require a predefined entity type schema, which is heavily utilized during their training process. For example, Ding et al. [49] or Lou et al. [57] experimentally find that negative sampling (i.e., specifying entity types not mentioned in the current document) is key to achieving state-of-the-art performance. However, if entity types are not specified (as in an open-world setting), it is impossible to replicate such a training procedure, nullifying the main contribution of these methods. Similarly, Zaratiana et al. [10] require a list of predefined entity types as input because they use the embeddings of the entity type names for their predictions. 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 necessitates the definition of new input formats or training procedures.

Prompting of frozen LLMs [39], [41] is easier to adapt, as it involves adjusting the prompt to remove the dependency on pre-specified entity types (see Sect. IV-A). However, the impact on performance when entity types are unspecified in the prompt remains unevaluated, and we expect a performance drop compared to zero-shot prompting.[4]

[4]We experimentally observe this in Sect. V-A.

B. UNSUPERVISED AND OPEN-WORLD NER

1) MOST UNSUPERVISED MODELS ARE NOT OPEN-WORLD

In theory, unsupervised models should be open-world since the lack of annotated data necessitates auto-structuration and type discovery techniques, such as clustering. However, this is not always the case. Historically, unsupervised NER has implemented rule and pattern-based models [58], [59]. These models are specific to a limited set of entity types, which restricts the discovery of unspecified types. In fact, the most recent unsupervised NER models face the same issue and require prior knowledge of the target entity types [7], [60], [61], [62]. Formally, they are zero-shot models since they need the specification of entity types, rather than being unsupervised approaches.

Jia et al. [60], Liu et al. [61], and Peng et al. [7] attempt to generalize transfer learning from $\mathcal{D}_S$ to $\mathcal{D}_T$, a method used in zero-shot settings. They train entity-type-specific models based on BERT embeddings, which they combine using a mixture of experts. These models require prior knowledge of the target entity types and access to labels for each entity type, albeit from a different domain. CycleNER [62] introduces a seq-to-seq model with a dual translation mechanism between text and entities. It consists of two models: S2E, which translates the document into a list of entities, and E2S, which generates text from a list of entities. These models are trained jointly, with S2E retained for predictions. CycleNER also needs to know the target entity types beforehand and requires lists of entities from $\mathcal{D}_T$.

In conclusion, these purportedly unsupervised models are neither truly unsupervised nor open-world, as they require a predefined specification of the target entity types. They are, in fact, more aligned with a zero-shot setting.

2) TRUE UNSUPERVISED AND OPEN-WORLD NER

To the best of our knowledge, only UNER [17] is compatible with true unsupervised and open-world scenarios. UNER employs clustering for mention detection and uses self-learning with autoencoders for entity typing. However, UNER is prone to drifting due to its reliance on self-learning and requires careful hyperparameter tuning, such as the number of training steps and learning rate, to avoid catastrophic performance drops. Unfortunately, UNER lacks both source code and a detailed explanation of how these hyperparameters are adjusted in an unsupervised manner, rendering their results unreproducible.

Interestingly, related fields such as unsupervised and few-shot relation extraction also face similar critiques regarding hyperparameter tuning [14], [63]. These fields depend on training procedures, like self-learning, that are sensitive to hyperparameter settings, which cannot be optimized without access to labeled data.

C. CONCLUSION AND RESEARCH GAPS

This concludes our literature review. Firstly, we observe remarkable dynamism in the research on zero-shot models,

characterized by a wide array of experiments involving LLMs [20], [48], full transformers [49], and encoders [10]. However, zero-shot models are not open-world and thus fail to address our research question. Moreover, most zero-shot approaches cannot be adapted to an open-world framework, as the specification of entity types is integral to their training procedure.

Similarly, so-called unsupervised approaches suffer from the same issue: they are unable to detect entity types not defined in the training set.

Throughout our literature review, we found only one unsupervised and open-world approach, UNER [17]. Yet, it has several shortcomings. Currently, the method is not reproducible due to the lack of source code and experimental details. Additionally, it relies on self-learning techniques known for their instability and hyperparameters that are difficult, if not impossible, to adjust. Genest et al. [14] highlighted the same problem in the field of unsupervised relation extraction.

In conclusion, numerous challenges persist in the task of unsupervised and open-world NER, motivating us to present OWNER.

III. DESCRIPTION OF OWNER

OWNER aims to extract and classify 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 $\boldsymbol{e} = [x_i, \ldots, x_j] \in \boldsymbol{X}$ that constitute entities and determine the type t for each $\boldsymbol{e}$. OWNER operates without prior knowledge of $\mathcal{D}_T$ and lacks access to:

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

Similar to recent zero-shot and few-shot models [8], [10], [21], OWNER utilizes a cross-domain transfer-learning approach. The strategy involves learning the NER task on a source domain $\mathcal{D}_S$, where annotated data is available, and transferring this knowledge to $\mathcal{D}_T$. $\mathcal{D}_S$ differs from $\mathcal{D}_T$ in style, semantics, and/or entity type perspective ($\mathcal{T}_S \neq \mathcal{T}_T$). We advance beyond zero-shot and few-shot methodologies by not predefining $\mathcal{T}_T$.

As illustrated in Fig. 1, OWNER employs a two-step process:

1) Mention detection. It identifies the spans $\boldsymbol{e}$ within $\boldsymbol{X}$ that are entities.
2) Entity typing. It classifies the type t for each detected entity. In practice, OWNER identifies clusters of entities that share the same type t.

A. MENTION DETECTION (MD)

Mention detection identifies entities $\boldsymbol{e}$ within a given document $\boldsymbol{X}$.

As discussed in the previous section, there are two primary prediction paradigms for MD: BIO sequence labeling extractors [5], [10], [27] and span-based extractors [6], [24]. Generally, span-based extractors perform slightly better than BIO models in supervised settings [2], [10]. However, since we lack supervision for $\mathcal{D}_T$, we opt to use BIO sequence labeling for MD due to its lower expressivity and complexity compared to span-based models. This choice is expected to enhance generalizability to unseen domains and new entity types [6]. BIO labeling models classify each token $x_i \in \boldsymbol{X}$ as B (beginning of an entity), I (inside an entity), or O (outside any entity).

We utilize encoder embeddings from pre-trained language models like BERT [4], combined with a linear classifier:

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

where $\boldsymbol{W}$ and $\boldsymbol{b}$ are learned weights, $\mathrm{Encoder}(x_i, \boldsymbol{X})$ is the encoder embedding of x_i in the context of $\boldsymbol{X}$, and σ is the softmax function. We fine-tune f_{MD}, including encoder weights, $\boldsymbol{W}$, and $\boldsymbol{b}$, on annotated documents from $\mathcal{D}_S$. Finally, BIO labels are decoded to identify the boundaries of each entity (the indices of its first and last tokens).

In fact, MD is the primary motivation for annotated data. The only MD model that operates without labels relies on self-learning [17]. However, self-learning tends to drift when overtrained. Preventing drift requires careful hyperparameter tuning, particularly concerning the number of training steps and the learning rate. Luo et al. [17] do not specify how to adjust these parameters without external annotated $\boldsymbol{X}$ from $\mathcal{D}_T$. Therefore, we propose using annotated documents from $\mathcal{D}_S$ to train MD in a supervised, cross-domain fashion to reduce the risk of unstable results. Annotations for $\mathcal{D}_S$ can originate from manually labeled datasets, distantly annotated datasets [21], or synthetically generated data [8]. In this article, we train OWNER on both manually labeled and synthetically generated datasets (see Sect. V-B).

B. ENTITY TYPING (ET)

Entity typing classifies the entities previously extracted through mention detection. In an unsupervised setting, the objective is to group entities that share the same entity type $t \in \mathcal{T}_T$. As depicted in Fig. 1, ET consists of three modules. These modules utilize well-established technologies that have demonstrated efficacy in NER, such as BERT prompting, clustering, and contrastive learning [64]. To our knowledge, these technologies have never been combined in this manner,[5] and it is their combination that facilitates open-world and unsupervised entity typing.

1) ENTITY ENCODER

The first module of ET is the entity encoder, which computes a vector representation (or entity embedding) for the current entity. We aim for this embedding to represent the entity type: two entities $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ with similar embeddings should share the same type t. Conversely, entities with different types t_1 and t_2 should have distinct embeddings. To encode entities, we propose using BERT prompting [14], [65]. A prompt $\mathcal{P}$ is a text containing one `[MASK]` token, which

[5] UNER [17] uses a distinctly different auto-encoder approach for ET.

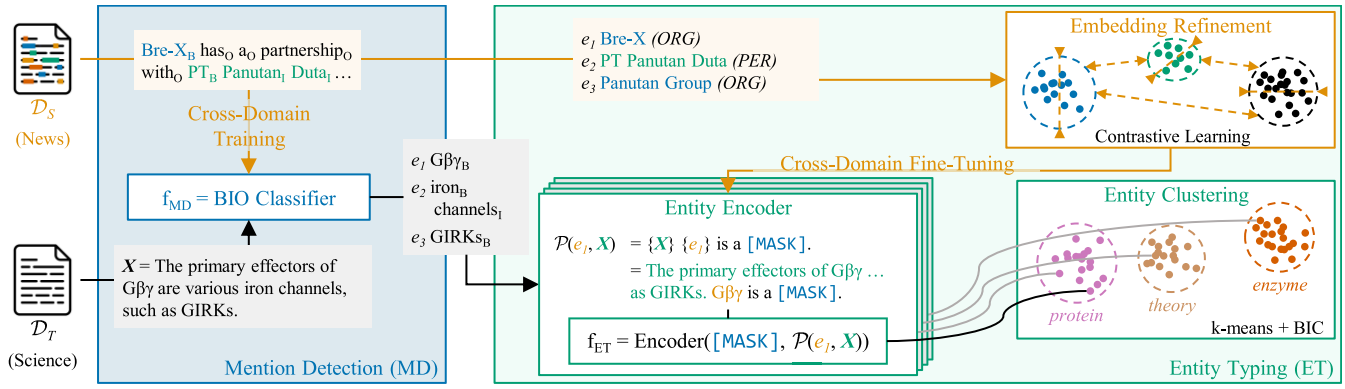


FIGURE 1. Overall architecture of OWNER.

the encoder processes. [MASK] signifies an unknown token, and the encoder computes an embedding representative of the missing word. By carefully designing $\mathcal{P}$, we can prompt the encoder to determine the type t of the current entity and use the [MASK] embedding as our entity embedding. The formulation of $\mathcal{P}$ is crucial for prompting performance and is typically adjusted using labels from $\mathcal{D}_T$ [66]. In our unsupervised setting, we choose not to tune it and select the simplest template possible:

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

with $\{\boldsymbol{X}\}$ variable substitution, where $\{\boldsymbol{X}\}$ is replaced by the text of $\boldsymbol{X}$. For instance (also shown in Fig. 1):

$$\begin{aligned}\boldsymbol{X} &= \text{“The primary effectors of G}\beta\gamma\text{are various iron channels, such as GIRKs.”},\\ \boldsymbol{e} &= \text{“G}\beta\gamma\text{”},\\ \mathcal{P}(\boldsymbol{e}, \boldsymbol{X}) &= \text{“The primary effectors of G}\beta\gamma\text{are various iron channels, such as GIRKs. G}\beta\gamma\text{is a [MASK].”}\end{aligned}$$

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

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

The choice of encoder embeddings over LLM or full transformer embeddings is motivated by two reasons. First, it allows us to define a fill-in-the-blank task that compels the model to predict precisely one word, likely describing the entity type. This approach simplifies the computation of the entity embedding and eliminates the need for aggregation techniques (e.g., mean pooling, attention [29]). Second, encoders are significantly smaller than LLMs (10–100 times smaller), making them more suitable for resource-constrained environments.

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 closely related and thus likely to share the same type $t \in \mathcal{T}_T$. We apply the simple k-means algorithm [67], [68]. Since the number of entity types is unknown, we must estimate the number of entity types (clusters) k. The only unsupervised prior work, UNER [17], required k to be predetermined. This requirement is counter-intuitive and unrealistic: if $\mathcal{T}_T$ is unknown, we cannot ascertain $|\mathcal{T}_T|$ (and thus k). Therefore, we aim to estimate k automatically.

Regarding the choice of the clustering algorithm, Genest et al. [14] observed empirically that k-means was the best-performing algorithm for unsupervised relation extraction. More complex clustering algorithms yielded lower results, likely due to their increased expressivity, which tended to model noise instead of valuable information. We review the suitability of k-means in comparison to other clustering algorithms (Gaussian Mixture Models [69], HDBSCAN [70], OPTICS [71]) in App. C.

a: BRUTE-FORCE ESTIMATION

Interestingly, k-means can be viewed as a simplification and approximation of a spherical Gaussian Mixture Model (GMM) [69]. The primary distinction lies in cluster membership: with k-means, each point belongs exclusively to one cluster (Dirac probability distribution), whereas GMM allows for soft-clustering assignments. One method to estimate the number of clusters in a GMM is to set an upper bound K, compute a clustering for each $k, 2 \leq k \leq K$, calculate the Bayesian Information Criteria (BIC) [18] for each clustering, and select $\hat{k}$ that minimizes BIC. BIC evaluates the clustering quality and adjusts it according to the model's complexity. Indeed, examining the right-hand side of Eq. (4), the left term assesses the fit's quality, while the right term estimates the model's complexity. BIC finds an optimal balance between clustering quality and complexity (number of clusters). We propose applying this same procedure to estimate the number of clusters with k-means, using the k-means BIC formula of Onumanyi et al. [72]:

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

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

with n representing the number of entities $\boldsymbol{e}_i$ extracted by MD, $\boldsymbol{X}_i$ denoting the document containing $\boldsymbol{e}_i$, and $\boldsymbol{c}_i$ being the centroid of the cluster containing $\boldsymbol{e}_i$. We refer to this procedure as *brute force cluster estimation*. This is the primary approach we use during OWNER's evaluation.

b: TERNARY SEARCH

A limitation of the previous approach is that it requires computing a clustering for each $2 \leq k \leq K$, which is computationally expensive. Empirically, we find that the BIC curve for ET is smooth, globally convex, and has a single minimum (see Fig. 9 (a)). This was observed across the 13 $\mathcal{D}_T$ datasets used during evaluation (see Sect. IV-B), with different encoder embeddings, and for every variation of OWNER. Given this experimental observation, it is possible to find the global minimum BIC without testing every possible k. One such method is the ternary search. We propose implementing this method and refer to it as *ternary search cluster estimation*. The ternary search follows an iterative approach, with each cycle as follows:

1) The input consists of a lower bound k_{min} and an upper bound k_{max} for the number of clusters.
2) Select k_1 and k_2 such that they divide the search space between k_{min} and k_{max} into thirds.
3) Compute the clustering and calculate the BIC for k_1 and k_2.
4) If k_1 has a lower BIC than k_2, set $k_{max} = k_2$; otherwise, set $k_{min} = k_1$.

The cycle repeats until $k_{max} = k_{min}$. With each cycle, the search space is reduced by a third, resulting in a logarithmic complexity of $\mathcal{O}(\log_3(K) \cdot$ k-means), compared to $\mathcal{O}(K \cdot$ 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 $k_{max} = k_1$; conversely, if the lowest BIC is at k_{max}, we set $k_{min} = k_2$. This allows us to eliminate two-thirds of the search space in one cycle.

Secondly, we propose removing the need to set a fixed upper bound K. We initially estimate $k_{max} = \sqrt{n}$ and permit the ternary search to increase k_{max} if the minimum BIC is located beyond it. During the first cycle, if the lowest BIC is at k_{max}, instead of updating k_{min}, we set $k_{max} = k_{max} + \frac{k_{max} - k_{min}}{3}$. This adjustment can continue in subsequent cycles until the lowest BIC is no longer at k_{max}.

Finally, the BIC curve is not entirely smooth locally. To improve the accuracy of the minimum estimation, 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 for the ternary search cluster estimation is shown in Fig. 2. The function call from the user should be *Ternary-Search*$(1, \sqrt{n}, \text{true})$. In practice, memoization is implemented to prevent recomputing the BIC multiple times for the same k, but this is omitted in the figure for clarity.

3) EMBEDDING REFINEMENT (ER)

ET is not trained using labeled documents. However, since MD uses labeled data in $\mathcal{D}_T$, we can also utilize this data for ET to more clearly isolate entity types during clustering. Contrastive learning has been applied for this purpose in the context of low-resource NER [25], [26]. The objective is to bring entities of the same type closer together and separate entities of different types by optimizing encoder representations. Existing models apply contrastive learning on the annotated data of $\mathcal{D}_T$, which we do not have. Therefore, 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 $\mathcal{L}_{\mathrm{TM}}$ [73]. $\mathcal{L}_{\mathrm{TM}}$ considers entity triplets $(\boldsymbol{e}^a, \boldsymbol{e}^+, \boldsymbol{e}^-)$. $\boldsymbol{e}^a$ is called the anchor. The positive entity $\boldsymbol{e}^+$ shares the same type as the anchor $\boldsymbol{e}^a$, while 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 $\boldsymbol{e}^-$ by a certain margin. The loss is defined as:

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

where $\mathrm{d}(\boldsymbol{e}^a, \boldsymbol{e}^+)$ is the Euclidean 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 set the $\mathcal{L}_{\mathrm{TM}}$ margin at 1. Empirically, we have not found that the margin significantly impacts performance.

Contrary to the usual encoder/BERT fine-tuning, a larger batch size is beneficial with contrastive learning [74], as it helps regularize the embedding space reorganization. The limiting factor for increasing 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 requires a substantial GPU memory footprint, hindering large batch sizes. To address this issue, we change our approach and consider batches of entities instead of batches of triplets. Each entity $\boldsymbol{e}$ is associated with the document $\boldsymbol{X}_{\boldsymbol{e}} \in \mathcal{D}_S$ 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 within the batch, adhering to the condition $(t_{\boldsymbol{e}^+} = t_{\boldsymbol{e}^a}) \wedge (t_{\boldsymbol{e}^-} \neq t_{\boldsymbol{e}^a})$. In our experimental setup, we can encode 128 entities per batch. Without this optimization, one batch comprises 42 triplets, and $\mathcal{L}_{\mathrm{TM}}$ does not converge. With this optimization, one batch contains, on average, more than 100 000 valid triplets.

C. NAMING THE CLUSTERS

The final step is to name the clusters to describe the entity types that have been extracted. It is important to note that previous unsupervised and open-world NERs [17], as well as related relation extraction models [15], [16], do not name the clusters. These models leave the task of naming the clusters to the user, who must inspect the entities within. We aim to advance this process by suggesting names that illustrate the cluster, which the user can then refine and complete. We propose two methods for this purpose.

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 $\arg\min_{k_{min} \le k \le k_{max}} (\mathrm{BIC}(k))$
  $k_1 = k_{min} + \mathrm{floor}\left(\frac{k_{max} - k_{min}}{3}\right)$
  $k_2 = k_{max} - \mathrm{floor}\left(\frac{k_{max} - k_{min}}{3}\right)$
  $k_{best} = \arg\min_{k \in \{k_{min}, k_1, k_2, k_{max}\}} (\mathrm{BIC}(k))$
  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 for the ternary search algorithm estimating the number of clusters.

1) USING BERT

To reiterate, our entity encoder employs BERT prompting, using the prompt defined in Eq. (2), to generate entity embeddings. Examining the prompt's formulation reveals that BERT will likely replace the `[MASK]` token with a word describing the entity type.

To name our clusters, we propose using BERT's Masked Language Model (MLM) head to predict the masked token of the prompt, leveraging the entity embedding. By iterating over each entity within a cluster, we can identify the most frequent predicted words that seem to describe it.

The main advantage of this method is that it does not require the use of LLMs and relies solely on the encoder. Moreover, since the embeddings have already been computed for entity typing, deriving name suggestions for the cluster is highly cost-effective.

However, there is a limitation. The prompt from Eq. (2) contains only a single `[MASK]` token. Thus, BERT will predict only one token, which could be a word or a sub-word. While some general and specific entity types can be expressed in a single token (such as *person*, *location*, *organization*, *protein*, *algorithm*, etc.), not all can (e.g., *astronomical object*, *programming language*, *chemical element*). This raises questions about the relevance of the predicted tokens for multi-word entity types. It will be interesting to observe how this method performs in practical experiments.

2) USING LLMs

Our second approach involves selecting a sample of entities (of size n) from a cluster and providing it as input to an LLM with the prompt described in Fig. 3. This prompt instructs the LLM to identify the entity type that encompasses the various entities presented. This approach could potentially address the limitations of the method discussed in the previous paragraph because it can easily predict multi-word entity type names.

System Message: A virtual assistant answers questions from a user based on the provided text.
Prompt: Given the list of entities $[\{\boldsymbol{e}_1, ..., \boldsymbol{e}_n\}]$, which all belong to the same type, predict the entity type corresponding to all these entities. Respond only with the name of the entity type.

FIGURE 3. LLM prompt for generating a name to describe an entity type cluster.

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 4. UniNER Uns prompt. Zhou et al. [8] proposed it to annotate the Pile-NER dataset.

However, it is more expensive and resource-intensive because it requires the use of an LLM. Nonetheless, we highlight that zero-shot NER approaches based on LLMs require at least one LLM call per document ($\mathcal{O}(|\mathcal{D}_T|)$), or even one call per document and entity type ($\mathcal{O}(|\mathcal{D}_T| \cdot |\mathcal{T}_T|)$). In contrast, our method requires only one call per cluster ($\mathcal{O}(\hat{k}) \sim \mathcal{O}(|\mathcal{T}_T|)$), making it significantly more cost-effective.

IV. EXPERIMENTAL SETUP

A. BASELINES

Luo et al. [17] did not release the source code for UNER, the only comparable unsupervised and open-world baseline, and we were unable to reproduce their results. To address this limitation, we propose an evaluation focusing on two directions.

1) 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 provide a context for evaluating the performance of unsupervised and open-world NER against more common and standard low-resource approaches.

We include *UniNER* [8], *GoLLIE* [20], and *ChatIE* [39]. UniNER and GoLLIE are LLMs fine-tuned on synthetically or manually labeled datasets, whereas ChatIE employs raw prompting. We also test *GliNER L* [10] and *GNER* [49], which respectively use an encoder (DeBERTav3 [50]) and a full transformer (Flan-T5 [51], [75]), both fine-tuned on the same dataset as UniNER.

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

1) Type Elicitation Prompt

System Message: A virtual assistant answers questions from a user based on the provided text.
Prompt: Given document: {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 5. ChatIE Uns prompt. ChatIE Uns employs a multi-turn question-answering setup, using the first prompt to identify entity types mentioned in the current document, followed by subsequent questions to identify entities for each elicited entity type.

2) UNSUPERVISED BASELINES CREATION

Since no unsupervised and open-world baseline is currently available for evaluation, we propose creating two baselines derived from zero-shot NERs. As discussed in Sect. II-A4, not all zero-shot models can be adapted for use in an open-world setting. Only LLM prompting can be directly adapted to function in an unsupervised and open-world manner.

First, Zhou et al. [8] annotated the Pile-NER dataset by prompting GPT-3.5 without specifying entity types, thereby operating in an open-world setting. The prompt they used is shown in Fig. 4. They did not evaluate this approach, so we include it to provide reference values for open-world GPT-3.5 prompting. We refer to this baseline as *UniNER Uns*. We also evaluate UniNER Uns using the more recent GPT-4o mini (*UniNER Uns (GPT-4o mini)*). Additionally, we attempted to replace GPT with Llama 3.1 8B Instruct and Qwen 2.5 7B Instruct, but these smaller models failed to adhere to the format specified in the prompt, resulting in null scores.

Second, the dual-stage method implemented by ChatIE [39], involving type elicitation and entity extraction, can be adapted for use in an unsupervised setting. Initially, type elicitation requires the list of entity types $\mathcal{T}_T$, but we can reformulate it to eliminate this dependency. The prompts we developed are shown in Fig. 5. We refer to this baseline as *ChatIE Uns*. We also evaluate ChatIE Uns using GPT-4o mini (*ChatIE Uns (GPT-4o mini)*) and the open-weight Llama 3.1 8B Instruct (*ChatIE Uns (Llama 3.1)*).[6] ChatIE Uns allows us to compare the performance of very similar zero-shot (ChatIE) and open-world (ChatIE Uns) models and to observe the impact of not specifying entity types beforehand.

3) IMPLEMENTATION DETAILS

We utilize the source code and hyperparameter values specified by the authors of the zero-shot models. For the adaptations of the unsupervised models, we apply the hyperparameter values defined by Wei et al. [39] and Zhou et al. [8]. The temperature is set to 0. The backbones and language model versions are detailed in Table 2.

[6]We also evaluated Qwen 2.5 7B Instruct, but found its performance lower than Llama 3.1 8B Instruct.

B. DATASETS

1) TARGET DOMAIN $\mathcal{D}_T$

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

- Five CrossNER datasets [3] (*AI*, *Literature*, *Music*, *Politics*, and *Science*). These datasets cover specific topics (scientific and literary) and unseen entity types.
- Two MIT datasets [76] (*Movie* and *Restaurant*). These datasets encompass new styles of text (reviews and search engine queries), specific topics, and unseen entity types.
- *FabNER* [77], which includes physics and chemistry articles labeled with scientific entity types.
- *GENIA* [78] and *i2b2* [79], which contain biomedical articles (sourced from PubMed) annotated with biomedical entities.
- *GENTLE* [80] and *GUM* [81], which cover unusual styles of text, such as dictionary entries, travel guides, legal notes, or poetry.
- *WNUT 17* [82], which comprises social network posts.

These datasets encompass a wide spectrum of text types (encyclopedic, scientific, biomedical, social networks, customer reviews, dictionary entries, etc.); domains (computer science, physics, chemistry, natural science, biomedical, literature, music, etc.); and entity types (*algorithm*, *protein*, *cell type*, *poem*, *mechanical property*, *animal*, *political party*, among many others). This diversity allows us to gain a comprehensive understanding of the quality and generalizability of OWNER. Full dataset statistics are available in App. E.

2) SOURCE DOMAIN $\mathcal{D}_S$

We propose to train OWNER using two datasets: *CoNLL-2003* [1] and *Pile-NER* [8]. These datasets represent two different approaches to the unsupervised setting.

CoNLL-2003 (hereafter referred to as CoNLL) embodies the cross-domain perspective. It contains general-domain newspaper articles that are manually annotated with four entity types (*person*, *location*, *organization*, and *misc*). CoNLL is chosen to be stylistically, semantically, and in terms of entity types, distinct from the $\mathcal{D}_T$ datasets. This allows us to assess the cross-domain capabilities of OWNER.

Pile-NER represents the synthetic data perspective. It consists of 50 000 documents collected from the Pile corpus [46] and automatically annotated by GPT-3.5,[7] resulting in 13 000

[7]Interestingly, Gao et al. [46] used real documents from the Pile corpus instead of generating them with GPT-3.5. They argue that achieving diverse documents and broad domain coverage with LLM-generated documents is challenging, which may lead to lower performance.

fine-grained entity types. The premise is that large and diverse $\mathcal{D}_T$ datasets enhance generalizability and help bridge the stylistic, semantic, or entity type gap between $\mathcal{D}_S$ and $\mathcal{D}_T$. Moreover, since the annotation process is automatic and does not require human intervention, it is neither time-consuming nor costly. In fact, recent few-shot and zero-shot models utilize large amounts of automatically annotated $\mathcal{D}_S$ data (e.g., UniNER, GliNER L, and GNER train on Pile-NER), and the results demonstrate the advantages of these automatically labeled corpora.

C. METRICS

We categorize evaluation metrics into two components:

1) Mention detection. These metrics assess whether the model accurately extracts entities, disregarding their types.
2) Entity typing. These metrics evaluate whether the model accurately classifies the entity types.

Entity typing metrics are also used to assess end-to-end NER, which integrates mention detection and entity typing.

1) MENTION DETECTION

Following previous studies (e.g., Zhong et al. [2]), we regard a predicted entity as correct if its boundaries match those of a ground truth entity:

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

where $\text{start}(\boldsymbol{e})$ (and respectively $\text{end}(\boldsymbol{e})$) is the index in $\boldsymbol{X}$ of the first (and respectively last) token of $\boldsymbol{e}$. In summary, a predicted entity matches a true entity if its starting and ending token indices coincide with those of the true entity. We define true positives (TP), false positives (FP), and false negatives (FN) as follows:

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

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

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

where $\mathbb{1}_{\{\cdot\}}$ is the indicator function, which equals one if the statement is true and zero otherwise. By convention, this approach excludes true negatives (TN).[8] In this single-label prediction context, the F1 score equals accuracy. We have:

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

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

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

[8] A TN is a span that is neither a true entity nor a predicted one. Since the number of spans increases quadratically with the document size and entities are relatively rare, TNs would overshadow TPs, FPs, and FNs, resulting in non-discriminative scores. Hence, the consensus (e.g., [2], [10]) is to omit true negatives.

BIO mention detection models, such as OWNER and GNER, must first decode the BIO labels to determine the boundaries of the entities. An entity begins with the presence of a *B* label, and the end is marked by the last token in the contiguous series of *I* labels following the *B* label.

Recently, some LLM-based approaches [8], [20], [39] have replaced the boundary check with a surface form check, which involves verifying that a predicted entity has the same text as a true entity. This change is less precise than an exact boundary check and can be problematic when multiple entities with the same surface form but different types appear in the same document (e.g., in "French persons speak French," the first *French* refers to a nationality, while the second refers to a language). In our evaluation, we assess all baselines and OWNER using the same boundary check metrics to ensure maximum fairness.

2) ENTITY TYPING AND END-TO-END NER

Since open-world methods automatically determine entity types, the set of predicted clusters (predicted entity types) does not necessarily match the set of true entity types. Moreover, predicted clusters cannot be directly associated with true entity types because there is no direct correspondence between cluster IDs and class IDs. Consequently, traditional classification metrics such as precision, recall, and F1 score are unsuitable for evaluating open-world NER models.

Inspired by the standard evaluation procedure for unsupervised and open-world relation extraction models (e.g., [14], [15]), we employ clustering metrics to evaluate entity typing and NER. These metrics, unlike classification metrics, are robust to permutations (i.e., different cluster IDs do not affect the final score) and to partial matches (e.g., when multiple clusters correspond to a single class or vice versa). Two widely used metrics for comparing clusterings with labels are the Adjusted Rand Index (ARI) [83], [84] and the Adjusted Mutual Information (AMI) [85].[9] In our experiments with unbalanced datasets, Romano et al. [88] suggest using AMI over ARI. AMI is adjusted for chance, meaning that random clustering will reliably yield a score close to 0. Furthermore, it is defined over the range $[-1, 1]$: scores below zero indicate methods less effective than random clustering.

We recall the definition of AMI. AMI is the adjustment for chance of the Mutual Information (MI) score. MI measures the mutual dependence between the true entity type and predicted entity type random variables. It is defined as:

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

where $\text{H}(t)$ denotes the Shannon entropy, and $\text{H}(t|\hat{t})$ represents the conditional entropy [85]. In these definitions, $\boldsymbol{e}$, $\hat{\boldsymbol{e}}$, t, and $\hat{t}$ are treated as random variables. Although $\boldsymbol{e}$ and $\hat{\boldsymbol{e}}$ are not explicitly mentioned in the equations, t depends on $\boldsymbol{e}$, and $\hat{t}$

[9] We acknowledge other metrics like the V-measure [86] or the B^3 [87]. However, these are not adjusted for chance (see subsequent paragraphs for an explanation).

depends on $\hat{e}$. To determine the actual values, the reader must enumerate all $e \in \mathcal{D}$ and $\hat{e} \in \mathcal{D}$. Entropy and conditional entropy are defined as follows:

$$\mathrm{H}(t) = -\sum_{t} \mathrm{P}(t) \log_2 \mathrm{P}(t), \tag{15}$$

$$\mathrm{H}(\hat{t}|t) = -\sum_{t,\hat{t}} \mathrm{P}(t,\hat{t}) \log_2 \frac{\mathrm{P}(t,\hat{t})}{\mathrm{P}(t)}. \tag{16}$$

Adjusted Mutual Information (AMI) adjusts the mutual information for chance, ensuring that a random clustering yields scores close to or equal to zero. It is defined as:

$$\mathrm{AMI}(t,\hat{t}) = \frac{\mathrm{MI}(t,\hat{t}) - \mathbb{E}_{t',\hat{t}'}\{\mathrm{MI}(t',\hat{t}')\}}{\max\{\mathrm{H}(t),\mathrm{H}(\hat{t})\} - \mathbb{E}_{t',\hat{t}'}\{\mathrm{MI}(t',\hat{t}')\}}. \tag{17}$$

Here, $\mathbb{E}_{t',\hat{t}'}\{\mathrm{MI}(t',\hat{t}')\}$ represents the expected MI between two random clusterings, estimated using a hypergeometric model of randomness [89].

Finally, the correspondences between the true and predicted entities are established using the equality defined in Eq. (7). A *predicted* placeholder with a specific error entity type is created for the true entities that were not predicted (FN). Conversely, for predicted entities that do not exist in the ground truth (FP), a *true* placeholder with a specific error entity type is created.

D. IMPLEMENTATION DETAILS

OWNER adopts a *train once, test anywhere* [90] methodology: it requires training only once on $\mathcal{D}_S$ and can then be applied to multiple $\mathcal{D}_T$ datasets without additional effort.[10] Regarding hyperparameters, since OWNER is unsupervised, we cannot use validation data to adjust them. We choose standard hyperparameter values as defined by Devlin et al. [4].

1) MENTION DETECTION

We utilize DeBERTa v3 embeddings [50], [91],[11] training the model for 4 epochs using the Adam optimizer [92]. We apply a decreasing linear schedule without warmup, a learning rate of 2×10^{-5}, a batch size of 32, and a dropout rate of $p = 0.1$ between the encoder and the linear classifier.

2) ENTITY TYPING

We employ BERT embeddings.[11] We use the simplest prompt possible, defined in Eq. (2), and train the model for 4 epochs using the Adam optimizer [92]. We apply a decreasing linear schedule without warmup, a learning rate of 2×10^{-5}, and a batch size of 128 as discussed in Sect. III-B3, with a dropout rate of $p = 0.1$. For brute force cluster estimation, we set 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.

[10]This does not imply perfect performance on distant domains, but rather that no adaptation or retraining is necessary to make predictions in an unseen domain.

[11] The choice of encoder embeddings is reviewed in App. B.

3) CLUSTER NAMING

For the BERT-based method, we use the same hyperparameters as outlined in the previous paragraph. With LLM-base naming, we employ the prompt defined in Fig. 3 using LLama 3.1 8B Instruct.[12] The temperature is set to 0 and the sample size n is set to 16.

4) COMPUTATIONAL RESOURCES

Experiments were conducted 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 typically lasts 50 min, and with Pile-NER, 5 h.

V. RESULTS AND ANALYSIS

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

A. COMPARISON WITH THE BASELINES

We first evaluate OWNER against unsupervised and zero-shot baselines on the 13 $\mathcal{D}_T$ datasets. The NER evaluation results, covering mention detection and entity typing, are presented in Fig. 6 and Table 1.

1) UNSUPERVISED OPEN-WORLD BASELINES

Overall, ● OWNER (Pile-NER) outperforms all open-world baselines, with an average AMI gap of 2.1 % compared to ◆ UniNER Uns (GPT-4o mini), 4.4 % compared to ◆ UniNER Uns, 11 % compared to ● ChatIE Uns (GPT-4o mini), 13 % compared to ● ChatIE Uns (Llama 3.1), and 18 % compared to ● ChatIE Uns.

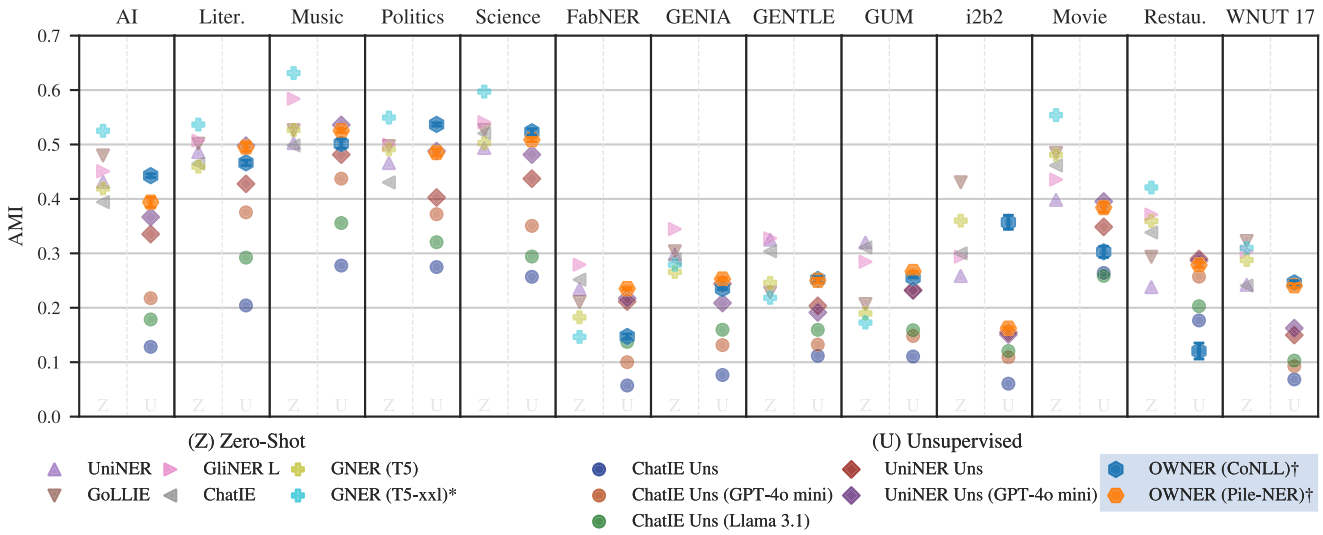
Even ⬢ OWNER (CoNLL), trained on a much smaller and more distant $\mathcal{D}_T$ dataset, surpasses the open-world baselines with at least a 1.3 % gap (UniNER Uns (GPT-4o mini)).

This demonstrates that our architecture effectively detects and types entities in an open-world and unsupervised setting. This result is significant when considering the size of the compared baselines relative to their performances (see Table 2). OWNER is the smallest model with its 110 M parameters, yet it outperforms much larger LLM baselines that are one to two orders of magnitude bigger. In compute-constrained environments, OWNER is a viable alternative to larger models, especially LLMs.

2) ZERO-SHOT BASELINES

Even compared to zero-shot models, which are more supervised, OWNER remains competitive. In general, zero-shot baselines outperform OWNER, as expected, since they have access to the list of entity types, including their names, descriptions, and some exemplars. Nevertheless, OWNER matches or surpasses the performance of ▲ UniNER on six datasets, ◀ ChatIE on five datasets, ✚ GNER (T5) on five datasets, ▼ GoLLIE on four datasets, ✚ GNER (T5-xxl) on

[12]The version is *Llama-3.1-8B-Instruct*.



* 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 6. NER evaluation results of OWNER, zero-shot baselines, and unsupervised baselines. Performance is measured using AMI. It encompasses mention detection and entity typing. Exact values are displayed in Table 1.

TABLE 1. NER evaluation results of OWNER, zero-shot baselines, and unsupervised baselines. Performance is measured using AMI. It encompasses mention detection and entity typing. The best AMI for each $\mathcal{D}_T$ dataset and setting (zero-shot, unsupervised) is highlighted in bold.

	AI	Liter.	Music	Politics	Science	FabNER	GENIA	GENTLE	GUM	i2b2	Movie	Restau.	WNUT 17	Avg.
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	36.1
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**	38.6
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	40.2
ChatIE	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	37.0
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	36.7
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	**41.2**
Unsupervised & Open-World														
ChatIE Uns	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	15.9
ChatIE Uns (GPT-4o mini)	21.8	37.5	43.7	37.2	35.0	10.0	13.1	13.2	14.8	10.9	38.5	25.7	9.3	23.9
ChatIE Uns (Llama 3.1)	17.9	29.2	35.6	32.1	29.4	13.7	16.0	15.9	15.9	12.1	25.8	20.3	10.3	21.1
UniNER Uns	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	30.1
UniNER Uns (GPT-4o mini)	36.7	**49.8**	**53.6**	48.8	48.1	21.7	20.9	19.1	23.2	15.1	**39.5**	28.6	16.3	32.4
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)**	33.7
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)	**34.5**

The standard deviation of OWNER is shown in parentheses: 44.3(3) indicates 44.3 ± 0.3.
* We could not run GNER (T5-xxl) on i2b2 due to excessive RAM consumption.

three datasets, and ▶ GliNER L on one dataset. Without accessing annotated data in $\mathcal{D}_T$ or knowing the target entity types $\mathcal{T}_T$, OWNER achieves commendable results compared to recent zero-shot approaches.

3) IMPACT OF THE OPEN-WORLD CONSTRAINT ON LLMS

In the introduction, we stated that zero-shot models are more supervised than unsupervised and open-world approaches because they have access to the list of entity types. Indeed, as observed in Table 1, zero-shot models typically perform better than open-world approaches. However, two concerns may arise:

- Perhaps the performance gap is due to architectural differences between zero-shot and open-world baselines.
- Can we quantify the loss caused by the open-world constraint?

These questions can be addressed by comparing ChatIE (zero-shot) with ChatIE Uns (unsupervised and open-world). ChatIE Uns uses the same LLM, architecture (dual-stage prompting with 1) type elicitation and 2) entity extraction), and prompts as ChatIE. The only distinction is that ChatIE Uns omits the list of entity type names from the type elicitation prompt. In Table 1, ChatIE Uns shows markedly lower performance compared to ChatIE, with an average gap of 21 % in AMI. This represents a relative 57 % performance loss. The small modification of removing the predefined list of entity types significantly impacts performance. This demonstrates that entity type specification

serves as a strong supervision signal, making unsupervised and open-world NER much more challenging than zero-shot NER. By comparison, OWNER (Pile-NER) only experiences a 2.5 % AMI gap compared to ChatIE (a relative 7 % loss). This highlights the architectural superiority of OWNER over LLM-based prompting baselines.

Another conclusion is that, when entity types are known in advance, employing zero-shot models is advisable due to their superior performance. However, for exploratory scenarios where entity types are not predetermined, unsupervised NERs, such as OWNER, lead in performance.

To conclude, we sought to explain the significant performance drop between ChatIE and ChatIE Uns. Upon closer examination, we attribute this to the overspecific entity types. In Table 6, which reports the estimation of the number of clusters for OWNER and unsupervised baselines, ChatIE Uns identified 11,840 entity types on the GUM dataset (instead of 10) or 14 680 types on i2b2 (instead of 20). The predicted entity types are overly specific, often concerning just a few entities: *lantern festival*, *theme music*, *light show*, *laser light show*. This occurs because ChatIE Uns lacks a merging or consensus mechanism for entity types, relying solely on the entity type text as the identifier. This issue also affects other unsupervised baselines, such as UniNER Uns.

4) SMALL VS. LARGE LANGUAGE MODELS

We conclude this baseline evaluation by reflecting on the size of the models involved. Notably, while most baselines are based on LLMs or transformers with at least 7 billion parameters, smaller models like GliNER and OWNER perform remarkably well despite their modest size (see Table 1). In particular, OWNER outperforms LLMs that are 60 to 100 times larger, and similarly, GliNER competes well against models two orders of magnitude larger. As a result, OWNER and GliNER are particularly appealing for real-world applications since they require significantly less computational power and more affordable infrastructure to operate.

Another observation is the shared characteristic of small size and high performance: their foundation on encoders. GNER (T5), based on a full transformer, performs less effectively at an equivalent size (Table 2). We believe that information extraction tasks such as NER are well-suited to an encoder formalism (making predictions using embeddings through neural networks), which benefits performance. In contrast, full transformers and decoders (LLMs) generate text, which is then interpreted to make predictions. This introduces a dual challenge: performing the task correctly (as with the encoder) while precisely adhering to the output format and avoiding hallucinations. Thus, the task becomes more complex than with an encoder, requiring a larger number of parameters to achieve the same level of performance.

Finally, an interesting aspect to discuss is distillation. OWNER and GliNER were trained on the Pile-NER dataset, which was automatically annotated by UniNER Uns. This process is somewhat related to distillation (Zhou et al. [57] refer to it as *targeted distillation*). In practice, we observe that the student models outperform the teacher model UniNER Uns (see Table 1). This is a compelling result. It suggests that the most promising approaches for practical deployments are hybrid methods (like OWNER and GliNER), which leverage LLMs solely for data generation and annotation, while the final model is a small encoder that is cost-effective to run but remains highly effective.

B. CROSS-DOMAIN CAPABILITIES

1) VALIDATING OWNER'S CROSS-DOMAIN CAPABILITIES

To validate OWNER's cross-domain capability, we trained it on CoNLL. CoNLL is a general-purpose dataset containing only four types of generic entities, not domain-specific ones. As shown in Table 1, OWNER (CoNLL) surpasses unsupervised and open-world baselines, clearly demonstrating OWNER's ability to generalize across distant domains. However, we observe a performance decrease of 0.7 % in AMI compared to OWNER (Pile-NER), indicating that using a more diverse dataset in terms of domain and entity types is beneficial for performance.

2) MENTION DETECTION ANALYSIS

In mention detection, CoNLL and Pile-NER result in models with different behaviors. In Table 3, we present the precision and recall of OWNER for mention detection. Overall, OWNER tends to have higher precision when trained on CoNLL and higher recall when trained on Pile-NER. This is expected; the diversity of Pile-NER helps OWNER detect entities more effectively, while the human quality of annotations in CoNLL enhances OWNER's precision. This observation is confirmed by examining the confusion matrices in Fig. 7. On one hand, Pile-NER leads to better detection of domain-specific entity types, such as *algorithm*, *field*, *metrics*, or *task*, but it also results in more false positives (497 for Pile-NER vs. 151 for CoNLL). On the other hand, CoNLL achieves better recall for *person*, *location*, or *organization*, which are precisely the entity types annotated in this dataset.

Finally, some performances shown in Table 3 are low: precision below 10 % for FabNER, GENTLE, GUM, or Restaurant (CoNLL), and recall below 10 % for i2b2 (Pile-NER). These are far from satisfactory for production deployment and illustrate the complexity of cross-domain learning and open-world NER. In fact, LLM-based baselines achieve even lower results than OWNER, as displayed in Fig. 6. This indicates that significant progress is still needed to achieve strong performance in very specific domains with specific entity types, where annotated data is lacking.

3) ARE FALSE POSITIVES A PROBLEM?

The issue of higher false positives with Pile-NER is intriguing and warrants further in-depth study. We manually examined the 497 false positives shown in Fig. 7. Of these, 53 %

TABLE 2. Comparative analysis of the language model backbones and the number of parameters in OWNER, zero-shot baselines, and unsupervised baselines.

Model	Backbone	Type	Version	# Parameters	Size compared to OWNER
Zero-Shot					
UniNER	Llama	Decoder	*llama-7b*	7 B	×60
GoLLIE	Code-Llama	Decoder	*CodeLlama-7b-Instruct*	7 B	×60
GliNER L	DeBERTa v3	Encoder	*deberta-v3-base*	300 M	×2.7
ChatIE	GPT 3.5	Decoder	*gpt-3.5-turbo-0613*	>8 B[1]	×70
GNER	Flan T5	Full transformer	*flan-t5-base*	275 M	×2.5
GNER (T5-xxl)	Flan T5 XXL	Full transformer	*flan-t5-xxl*	11 B	×100
Unsupervised					
ChatIE Uns	GPT 3.5	Decoder	*gpt-3.5-turbo-0613*	>8 B[1]	×70
ChatIE Uns (GPT-4o mini)	GPT-4o mini	Decoder	*gpt-4o-mini-2024-07-18*	>8 B[1]	×70
ChatIE Uns (Llama 3.1)	Llama 3.1 Instruct	Decoder	*Llama-3.1-8B-Instruct*	8 B	×70
UniNER Uns	GPT 3.5	Decoder	*gpt-3.5-turbo-0613*	>8 B[1]	×70
UniNER Uns (GPT-4o mini)	GPT-4o mini	Decoder	*gpt-4o-mini-2024-07-18*	>8 B[1]	×70
OWNER	DeBERTa v3 BERT	Encoder	*deberta-v3-base* *bert-base-uncased*	110 M[2]	

[1] Although not disclosed, GPT-3.5 and GPT-4o mini are expected to be larger than Llama 3.1 8B Instruct.
[2] OWNER uses two encoders with a total of 200 M parameters (90 M for DeBERTa v3 and 110 M for BERT). However, only one is loaded at any given time.

TABLE 3. OWNER mention detection evaluation results. The best precision and recall for each $\mathcal{D}_T$ dataset are highlighted in bold.

	OWNER (CoNLL)		OWNER (Pile-NER)	
	Precision	Recall	Precision	Recall
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)**

The standard deviation of OWNER is displayed in parentheses.

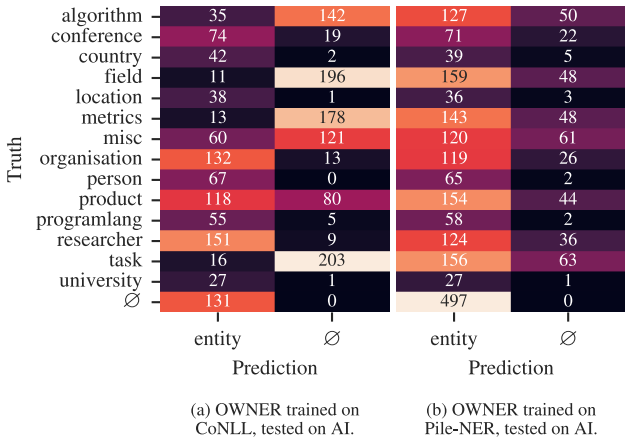


FIGURE 7. Mention detection confusion matrices of OWNER on AI. The ∅ row indicates the false positives, while the ∅ column indicates the false negatives per entity type.

are correct entities not annotated in AI, 42 % intersect with true entities (boundary problem), and 5 % are incorrect predictions. Overall, the boundary problem accounts for the false positive difference between CoNLL and Pile-NER, likely due to Pile-NER's imperfect annotations.

The 53 % correct entities not annotated in AI derive from existing entity types (many missing entities are acronyms, such as FPR = false positive rate) and new entity types (not among the 14 entity types annotated in AI). OWNER's ability to identify correct entities of new entity types underscores its novelty detection capabilities. This behavior is not observed with other zero-shot and few-shot baselines, which have a predefined set of entity types.

In conclusion, OWNER's cross-domain capabilities are highlighted by the strong performance of OWNER (CoNLL) on the $\mathcal{D}_T$ datasets. Broadly, CoNLL's manual annotations yield precise results, while the diversity of Pile-NER enhances recall at the expense of precision. Additionally, the analysis of the confusion matrices shows that OWNER identifies entities of novel entity types that were previously unknown. In scenarios focused on novelty detection or exploration, where recall is crucial, we recommend using Pile-NER. Its diversity of domains and entity types boosts performance and it is relatively inexpensive to produce as it does not depend on manual annotations.

C. BIO SEQUENCE LABELING

In Sect. III-A, we propose using a BIO extractor for MD, as we anticipate that the simplicity of this architecture will enhance generalizability across new target domains $\mathcal{D}_T$. In Table 4, we report the F1 scores of various MD

TABLE 4. Mention detection evaluation results for different architectures trained on CoNLL and tested on five $\mathcal{D}_T$ datasets. Performance is measured using the F1 score. The best F1 score for each $\mathcal{D}_T$ dataset is highlighted 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)

The standard deviation of OWNER is displayed in parentheses.
Only one run (no standard deviation) for PURE and SpanProto due to the slowness of their training.

architectures, trained on CoNLL and tested on five $\mathcal{D}_T$ datasets. We evaluate the following architectures:

- BIO: This is the architecture implemented by OWNER.
- PURE [2]: A span-based extractor that combines the start and end embeddings of a candidate span with a perceptron.
- SpanProto [5]: A span-based extractor that uses bilinear neurons to combine start and end embeddings of a candidate span, offering faster predictions compared to PURE.
- WL-Coref [29]: A span-based extractor that identifies the head of the entity and reconstructs its boundaries using a convolutional network, addressing the quadratic complexity issue of traditional span-based extractors.

In a fully supervised setting, PURE, SpanProto, and WL-Coref are slightly superior to BIO sequence labeling [2], [5], [29]. However, in our unsupervised cross-domain setting, BIO significantly outperforms span-based extractors, with an average gap of 40 % with PURE, 15 % with SpanProto, and 10 % with WL-Coref, while also being faster to train. We believe that the simplicity of the BIO architecture reduces overfitting and enhances generalizability to new domains. This observation aligns with the findings of Fang et al. [6], who also employ BIO sequence labeling for their few-shot MANNER model.

D. IMPACT OF EMBEDDING REFINEMENT

An important component of OWNER is embedding refinement (ER), which aims to enhance encoder representations for entity clustering using contrastive learning. In Table 5, we compare OWNER's entity typing performance without ER and with ER trained on CoNLL or Pile-NER. We use the gold entity spans from $\mathcal{D}_T$ (perfect MD) to assess only the effect of embedding refinement. This is why the AMI scores are higher than in Table 1.

ER significantly improves performance with both 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 notable for datasets challenging for raw BERT embeddings, such as GENTLE, GUM, i2b2, Movie, Restaurant, or WNUT 17. Pile-NER's superior performance can be attributed to its diversity of entity types (13 000 entity types), which allows for more precise fine-tuning of entity embeddings. Nevertheless, CoNLL achieves commendable performances despite having only four entity types. This supports the hypothesis that refining entity embeddings on $\mathcal{D}_S$ with contrastive learning also benefits distant $\mathcal{D}_T$.

To visually represent the effects of embedding refinement, we display two-dimensional t-SNE [93] representations of the entity embeddings for the Science and Restaurant datasets in Fig. 8. The Science entities are already well isolated without ER (see Table 5). However, several improvements are noticeable: better separation of *discipline*, *organization*, and *academicjournal* (CoNLL and Pile-NER); better separation of *chemicalelement* and *chemicalcompound* (Pile-NER); and the disappearance of the multi-type cluster at the top of the w/o ER figure. The effects of ER are more evident with the challenging Restaurant dataset: without ER, OWNER cannot discriminate any entity type, and we see significant improvements with ER on CoNLL or Pile-NER. Notably, ER with CoNLL leads to relatively good separation of *cuisine*, *hours*, or *price*, even though CoNLL does not contain such entities. The effects are more comprehensive and pronounced with Pile-NER.

In conclusion, embedding refinement significantly enhances entity typing performance with CoNLL and Pile-NER, showing respective AMI improvements of 12.8 % and 16.7 %. The best results are achieved with Pile-NER due to its diversity in entity types. ER performs well with the distant $\mathcal{D}_T$ dataset CoNLL, demonstrating noticeable improvements on unseen entity types. It also shows that ER is beneficial even with a labeled dataset with a limited set of entity types (4 for CoNLL).

E. ESTIMATING THE NUMBER OF CLUSTERS $\hat{K}$

1) COMPARISON WITH THE BASELINES

Since we lack information about entity types, unlike zero-shot approaches, OWNER must infer both the types of entities and their number. In this section, we focus solely on the brute force cluster estimation. In Table 6, we present each $\mathcal{D}_T$ dataset's true number of entity types k, the estimated number of clusters $\hat{k}$, the corresponding AMI score with $\hat{k}$ (as shown in Fig. 6), and the AMI score with the ideal k.

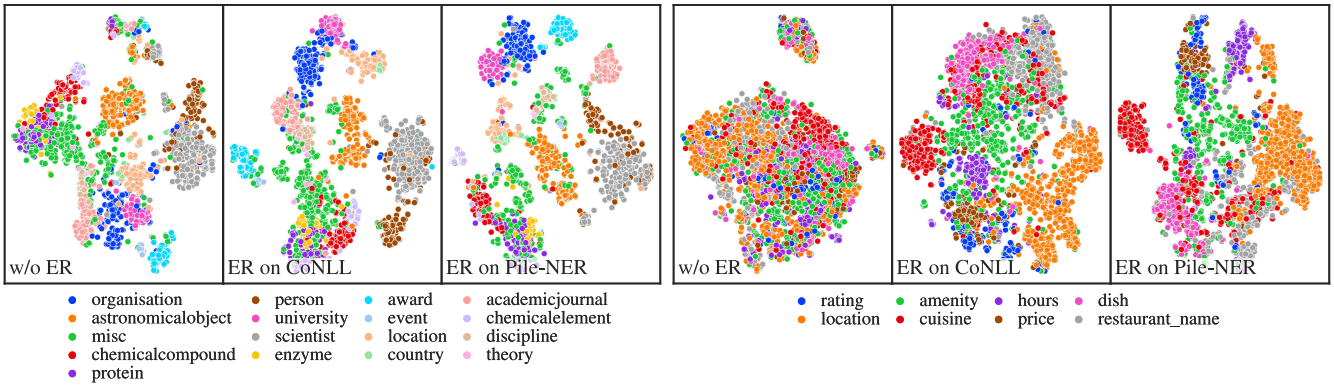
Overall, OWNER tends to overestimate the number of entity types, a trend more pronounced with Pile-NER than with CoNLL. For Pile-NER, this overestimation can be attributed to its fine-grained entity types. Pile-NER was annotated using UniNER Uns (with GPT-3.5). Thus, Pile-NER shares the fine-grained entity type weakness of UniNER Uns. Fortunately, OWNER somewhat alleviates this issue by providing a more reasonable estimate of the number of clusters, as shown in Table 6. In any case, compared to UniNER Uns and ChatIE Uns, OWNER's estimations are much closer to the actual values.

AMI scores with the ideal k are close to those with $\hat{k}$ (an AMI gap of 0.8 % for CoNLL and 1.5 % for Pile-NER on average), indicating that the clusterings are qualitatively

TABLE 5. Ablation study for embedding refinement (ER). Entity typing evaluation results of OWNER with and without embedding refinement, trained on CoNLL or Pile-NER. Performance is measured using AMI on gold entity spans (perfect mention detection). The best AMI for each $\mathcal{D}_T$ dataset is highlighted in bold.

	AI	Liter.	Music	Politics	Science	FabNER	GENIA	GENTLE	GUM	i2b2	Movie	Restau.	WNUT 17
without 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)**

The standard deviation of OWNER is displayed in parentheses.



(a) OWNER tested on Science.

(b) OWNER tested on Restaurant.

FIGURE 8. Visualization of the entity embeddings of OWNER with or without embedding refinement (ER). Entity embeddings are displayed using two-dimensional t-SNE. Each subfigure, from left to right, shows: 1) without embedding refinement, 2) embedding refinement on CoNLL, and 3) embedding refinement on Pile-NER.

similar even when $\hat{k} \gg k$. This can be explained by the long-tail distribution of cluster membership. In the second confusion matrix of Fig. 10, a minority of clusters contain most entities, while the rest contain a few specific entities. The last 17 clusters actually represent false positives[13] and members of the *misc* class, which by definition consists of multiple entity types. This explains why, despite the number of clusters, performance does not drastically decline, as the additional clusters mainly account for false positives and composite classes.

Finally, we focus on the baselines. ChatIE Uns and UniNER Uns make very poor cluster number estimations, often 10 to 100 times larger than the actual number. This is because they explicitly express entity types using words, and the same entity type can be represented by different terms. For example, *person* may be expressed as *PER*, *person*, *individual*, etc., all considered different clusters. Moreover, ChatIE Uns and UniNER Uns tend to predict overly fine-grained entity types (e.g., *lantern festival*, *theme music*, *light show*, or *laser light show*). In practice, they predict so many different entity types that the results from UniNER Uns and ChatIE Uns become unmanageable and practically useless due to insufficient synthesis. In contrast, OWNER's more reasonable number of clusters is easier to analyze in practice. We believe this highlights an architectural advantage of OWNER: clustering the embeddings, rather than specific entity type names, allows for the creation of more synthetic clusters.

[13]These can be correct entities, as discussed in Sect. V-B.

2) FASTER ESTIMATION USING TERNARY SEARCH

Up to this point, we have used the brute force algorithm to estimate the number of clusters $\hat{k}$. While the computational time is acceptable for small datasets, it can take hours for the largest $\mathcal{D}_T$ datasets, such as i2b2 or GUM (see Table 8), representing a significant portion of the runtime. For example, cluster estimation takes an average of 13.6 h for $\mathcal{D}_S$ = Pile-NER and $\mathcal{D}_T$ = i2b2. This motivates the use of the ternary search algorithm we introduced in Sect. III-B2.

In Table 7, we compare the estimation of $\hat{k}$ using the brute force algorithm and ternary search, along with the corresponding NER AMI scores. In Table 8, we show the corresponding execution times. We observe that the estimation of $\hat{k}$ with ternary search matches the brute force algorithm or falls within the standard deviation range. As a result, ternary search AMI scores are virtually identical to those from brute force.

The gain in computational time is particularly noteworthy. As shown in Table 8, ternary search is 1.7 to 2.7 times faster than the brute force algorithm, even for the smallest datasets. The benefit is especially impressive for the large i2b2 dataset with its large $\hat{k}$, where the gain is twenty-fold. Originally, runs lasted 13.6 h, but with ternary search, they are reduced to just 41 min. The computational gain is less pronounced

TABLE 6. Evaluation of OWNER's estimation of the number of clusters $\hat{k}$ using BIC. NER evaluation results of OWNER with the exact number of entity types (AMI with k) and with the automatic estimation (AMI with $\hat{k}$) are also shown. The best AMI for each $\mathcal{D}_S$ and $\mathcal{D}_T$ dataset is highlighted in bold. The true number of entity types k and the predicted number of clusters $\hat{k}$ are displayed in blue. We also include the $\hat{k}$ estimated by the unsupervised baselines.

	AI	Liter.	Music	Politics	Science	FabNER	GENIA	GENTLE	GUM	i2b2	Movie	Restau.	WNUT 17
True 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 with $\hat{k}$	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 with 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 with $\hat{k}$	**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 with 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													
ChatIE Uns	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 (GPT-4o mini)	895	383	407	572	973	3,524	3,002	1,081	7,346	6,426	526	433	1,245
ChatIE Uns (Llama 3.1)	724	426	368	517	798	2,030	1,872	702	5,374	4,149	387	202	760
UniNER Uns	155	92	115	103	195	292	319	250	830	1,033	176	117	266
UniNER Uns (GPT-4o mini)	165	108	126	136	244	334	414	281	1,033	1,386	158	131	320

The standard deviations for $\hat{k}$ and AMI are displayed in parentheses.

TABLE 7. Comparison of OWNER's estimation of the number of clusters $\hat{k}$ using the brute force algorithm or ternary search. OWNER was trained on Pile-NER. NER evaluation results of OWNER with the exact number of entity types (AMI with k) and with the automatic estimation (brute AMI with $\hat{k}$ and ternary AMI with $\hat{k}$) are also shown. The best AMI for each $\mathcal{D}_T$ dataset is highlighted in bold. The true number of entity types k and the predicted number of clusters $\hat{k}$ are displayed in blue.

	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 with 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)**
brute AMI with $\hat{k}$	**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)
ternary AMI with $\hat{k}$	**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)

The standard deviations for $\hat{k}$ and AMI are displayed in parentheses.

TABLE 8. Comparison of the execution time (in seconds) for OWNER's estimation of the number of clusters using the brute force algorithm or ternary search. OWNER was trained on Pile-NER.

	AI	Liter.	Music	Politics	Science	FabNER	GENIA	GENTLE	GUM	i2b2	Movie	Restau.	WNUT 17
brute	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	**48(1)**	**52(1)**	**69(1)**	**89(3)**	**69**	**189(4)**	**142(5)**	**65(1)**	**906(28)**	**2,440(173)**	**117(2)**	**74**	**57(1)**
	(÷1.8)	(÷1.9)	(÷2.8)	(÷2.6)	(÷2.4)	(÷2.7)	(÷2.7)	(÷1.8)	(÷2.0)	(÷20.1)	(÷2.3)	(÷1.9)	(÷1.7)

The standard deviation of the execution time is displayed in parentheses.

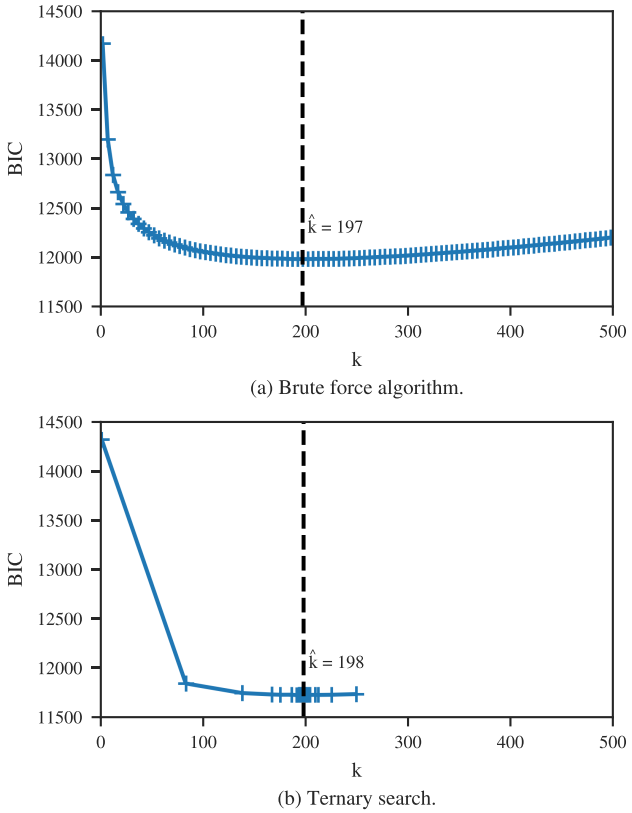
for smaller sets of entity types (though still significant) due to the slight rugosity of the BIC curve. This rugosity necessitates multiple sequential cluster computations once $k_{max} - k_{min} \leq 5$.

The i2b2 dataset case is particularly interesting. In Fig. 9, ternary search quickly converges to the minimum value without evaluating every possible $\hat{k}$. Specifically, it eliminates the range [0, 140] clusters in two steps (5 min), whereas brute force requires 2 h to evaluate the same interval. Ternary search determines $\hat{k}$ after 21 clusterings, compared to the 500 needed by the brute-force algorithm (a 24-fold reduction).

In conclusion, the computational gain from ternary search is particularly significant for large $\mathcal{D}_T$ datasets with many different entity types. It is also relevant for smaller datasets, achieving 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.

F. QUALITATIVE ANALYSIS

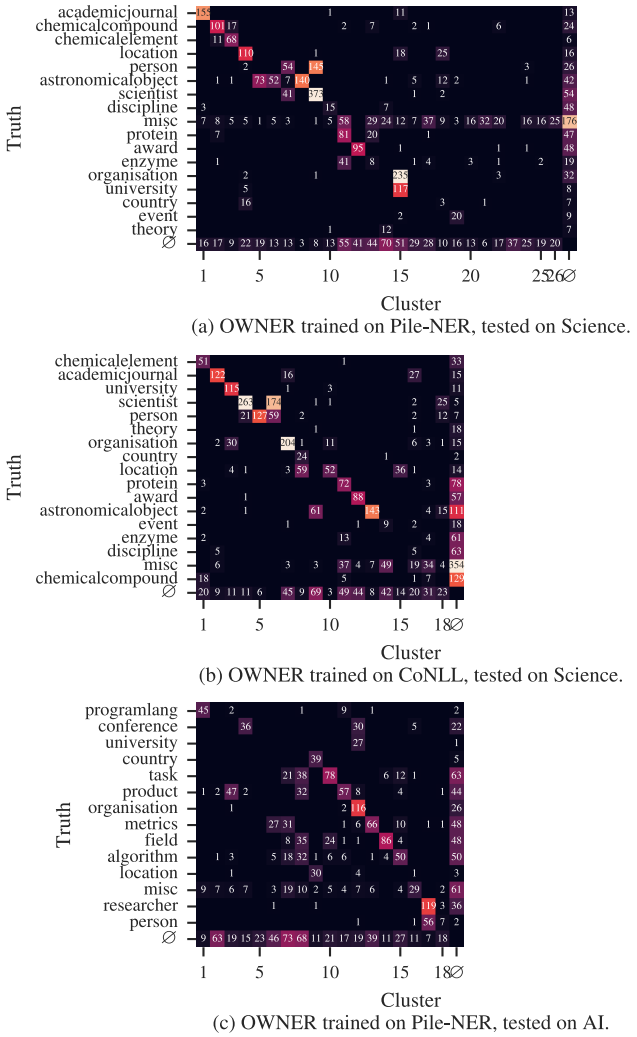
We conclude this analysis by providing a qualitative overview of OWNER's performance from two perspectives: 1) an examination of the confusion matrices to identify OWNER's



(a) Brute force algorithm.

(b) Ternary search.

FIGURE 9. Visualization of the BIC curves computed to estimate the number of clusters k̂. OWNER is trained on Pile-NER and tested on i2b2. Each cross represents a computed clustering. Using the brute force algorithm, 500 clusterings were calculated, while ternary search required only 21.



(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 10. NER confusion matrices of OWNER tested on various $\mathcal{D}_T$ datasets. Columns and rows were reordered using the algorithm described in App. D. The ∅ row indicates the false positives, and the ∅ column indicates the false negatives.

strengths and weaknesses, and 2) an evaluation of OWNER's naming capabilities.

1) CONFUSION MATRICES

In Fig. 10, we present three confusion matrices for OWNER, trained with different $\mathcal{D}_S$ datasets and tested on various $\mathcal{D}_T$ datasets.

These matrices reveal a distinct diagonal pattern, indicating that OWNER accurately identifies most entity types. This is remarkable, as OWNER can detect and structure entities in a manner resembling the ground truth, despite the absence of annotated data in $\mathcal{D}_T$ and any prior knowledge of entity types or their numbers.

To analyze OWNER's weaknesses, we focus on the confusions. Impure clusters can be problematic if the model groups unrelated entity types. However, this issue is not present here. 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);
- *conference*, *university*, *organization* (AI).

These impure clusters arise because OWNER confuses semantically similar entity types. This behavior is expected and reassuring. It is a limitation inherent to open-world NER. Without a predefined list of entity types, OWNER organizes entities into a semantically coherent scheme, which is a valid typing system, though not identical to the dataset's annotation schema.

Finally, OWNER categorizes false positives and *misc* entities as composites of multiple underlying types. This tendency explains why OWNER often overestimates the actual number of entity types.

In conclusion, OWNER forms a coherent typing scheme closely aligned with the true entity types. This analysis underscores OWNER's exploratory capabilities. It can identify and organize entities into meaningful groups without labeled data in $\mathcal{D}_T$. OWNER effectively processes unannotated documents to uncover primary entities and their types, paving the way for further refinement through more supervised methods.

TABLE 9. Predicted names (using BERT and LLMs) for six entity type clusters identified by OWNER on Science when trained on Pile-NER. The true entity type distributions are also displayed. The corresponding confusion matrix is shown in Fig. 10 (a).

Cluster	True entity types	Example entities	BERT-based naming	LLM-based naming
3	68 % chemicalelement 17 % chemicalcompound 9 % *false positives* 6 % *other*	alpha-elements HDPE Thallium HCl gases	semiconductor mineral compound	chemical compound
6	78 % astronomicalobject 20 % *false positives* 2 % *other*	216 Kleopatra Ceres 39882 44 Nysa	crater asteroid	asteroid
8	98 % astronomical object 2 % *other*	Saturn Jupiter Eris	planet moon star	planet
9	70 % scientist 27 % person 3 % *other*	Leonardo da Vinci Candy Brown Robert Riddell Georg Stetter	physicist Christian vegetarian historian biologist	person
11	34 % protein 24 % misc 23 % *false positives* 17 % enzyme 2 % *other*	ZO-1 P-type calcium channel ERK 1/2 TGFβ	protein receptor kinase	gene
23	100 % *false positives*	1993 May 2008 1985 2011	year	date

2) NAMING THE CLUSTERS

The second part of our qualitative analysis focuses on naming the clusters. We proposed two approaches: one using BERT embeddings generated by the entity encoder, and the other employing LLM prompting. In Table 9, we present six clusters with names predicted by either BERT or Llama 3.1 Instruct.

We observe that the proposed names, whether from BERT or the LLM, are generally relevant and descriptive of the entity types in the clusters (see the columns *True entity types* and *Example entities* in Table 9). However, most names are not perfect enough to entirely replace human intervention. For instance, *year* (BERT) is less appropriate than *date*, and *person* (Llama 3.1) is less specific than *scientist*. Nevertheless, they provide a fairly precise and concise idea of the cluster's contents, effectively serving the purpose of describing and naming clusters.

Regarding the advantage or difference between BERT and LLM naming, the results are quite similar. The LLM tends to provide a single name that generally represents the cluster well, whereas the multiple names from BERT can sometimes offer a more precise understanding of the content. Additionally, BERT's limitation to predict a single token does not seem particularly impactful, as the names remain descriptive (even for cluster 3, which the LLM described with two words).

The case of cluster 9 is particularly interesting. This cluster comprises *persons* and *scientists*. We expected BERT to predict the term person, but it did not. In fact, BERT never predicted the name person for this cluster. We believe this is because the phrase "X is a person" is not commonly used (the mere presence of a name implies it's a person), so BERT does not predict it. Instead, BERT predicts terms like *physicist*, *historian*, *biologist* (jobs), and *Christian* and *vegetarian* (philosophy and religion). It is indeed more common to say "X is a [job]" or "X is a [religion/philosophy]" than "X is a person." This property is a weakness of BERT naming and is directly linked to its pretraining corpus.

Overall, we find that using BERT to name clusters is relevant because the results are generally coherent and descriptive, and importantly, there is no additional cost to its use, as BERT embeddings are already generated for clustering (unlike the LLM).

Naming the clusters also provides insight into OWNER's internal reasoning, particularly regarding potential weaknesses: clusters composed solely of false positives (cluster 23) and types of entities split into multiple clusters (clusters 6, 8).

Cluster 23 consists entirely of false positives, yet in practice, it is made up solely of dates, a type of entity that had not been annotated in Science. This effectively illustrates the conclusion of Sect. V-B that OWNER can detect types of

entities not anticipated in advance, which is impossible with zero-shot models.

Finally, clusters 6 and 8 mostly contain astronomical objects, and the separation of these two types of entities is an error from the dataset's annotation perspective. However, upon closer inspection of the proposed names, we notice that cluster 6 contains small celestial objects (asteroids), while cluster 8 focuses more on planets and other massive astronomical objects. Although this separation is technically an error (from the annotation and metrics point of view), the decision is qualitatively and semantically justified and highlights the quality of OWNER's predictions.

VI. 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 indicates that while significant progress has been made in lower-resource NER, particularly zero-shot NER, unsupervised and open-world NER still lags behind. OWNER is proposed as the first NER model compatible with a fully unsupervised open-world scenario, aiming to provide a strong baseline and stimulate further research. OWNER is built upon a simple yet innovative architecture, featuring an encoder prompting, clustering, and embedding refinement triangle.

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

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

Finally, we demonstrate that an unsupervised and open-world NER is achievable without the use of LLMs, achieving better performance than LLMs and also being capable of naming the clusters. A key advantage of OWNER over LLM-based approaches is its ability to group entities into clusters that are semantically coherent and closely aligned with the true entity types. In contrast, LLMs struggle to generalize and tend to predict entity types that are overly specific, making them impractical and difficult to analyze.

For future work, we aim to expand OWNER for use in a low-resource active learning context [94]. We believe OWNER's ability to structure entities without supervision could help bootstrap an active learning cycle.

Another research area is combining open-world and closed-world NER. The objective is to allow users to predefine a typing scheme for known entities while leaving room for novel, unseen knowledge, for which the model will generate a typing structure. Preliminary work [16], [95], [96] has been done in the related field of relation extraction, but these models are not currently low-resource.

TABLE 10. List of abbreviations.

Abbreviation	Full Name
AMI	Adjusted Mutual Information [85], defined between $[-1, 1]$
BIO	Begin-Inside-Outside sequence labeling [97]
Encoder	Encoder-Only Language Model, for instance, BERT [4]
ER	Embedding Refinement
FN	False Negatives
FP	False Positives
KG	Knowledge Graph
LLM	Large Language Model
MD	Mention Detection
NER	Named Entity Recognition
TN	True Negatives
TP	True Positives

TABLE 11. List of mathematical notations.

Notation	Meaning
$\mathcal{D}_S$	Source domain
$\mathcal{D}_T$	Target domain
$\boldsymbol{e}$	Entity
k	Number of entity types (or number of clusters)
K	Upper bound for the number of clusters
$\mathcal{P}$	Prompt
σ	Softmax function
t	Entity type
$\mathcal{T}_S$	Set of the entity types present in $\mathcal{D}_S$
$\mathcal{T}_T$	Set of the entity types present in $\mathcal{D}_T$
x	Token (word, part of word, or punctuation as defined by SentencePiece [19])
$\boldsymbol{X}$	Document

APPENDIX A
ABBREVIATIONS AND MATHEMATICAL NOTATIONS

Abbreviations are listed in Table 10. Mathematical notations are listed in Table 11.

APPENDIX B
ENCODER EMBEDDINGS IMPACT

With OWNER, we primarily use DeBERTa v3 [50], [91] for mention detection and BERT [4] for entity typing. In this section, we evaluate the performance of other popular encoders, such as RoBERTa [98], ERNIE [52], and ELECTRA [99].

In Table 12, we present the mention detection performance of various encoders when OWNER is trained on Pile-NER, and in Table 13, we show the entity typing performance of the same encoders (also on Pile-NER). Overall, OWNER performs relatively well, regardless of the encoder used as

TABLE 12. Mention detection evaluation results of OWNER trained on Pile-NER, using various encoder embeddings. Performance is measured using the F1 score. The best F1 score for each $\mathcal{D}_T$ dataset is highlighted in bold.

	AI	Liter.	Music	Politics	Science	FabNER	GENIA	GENTLE	GUM	i2b2	Movie	Restau.	WNUT 17	Avg.
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**

The standard deviation of OWNER is displayed in parentheses.

TABLE 13. Entity typing evaluation results of OWNER trained on Pile-NER, using various encoder embeddings. Performance is measured using AMI. Entity typing is evaluated using gold entity spans (perfect mention detection). The best AMI for each $\mathcal{D}_T$ dataset is highlighted in bold.

	AI	Liter.	Music	Politics	Science	FabNER	GENIA	GENTLE	GUM	i2b2	Movie	Restau.	WNUT 17	Avg.
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

The standard deviation of OWNER is displayed in parentheses.
The number of clusters is estimated using the ternary search algorithm, which explains why AMI scores are not identical to those in Table 5 (brute force). However, they remain within the range of standard deviation.

TABLE 14. Entity typing evaluation results of OWNER trained on Pile-NER, using various clustering algorithms. Performance is measured using AMI. Entity typing is evaluated using gold entity spans (perfect mention detection). The best AMI for each $\mathcal{D}_T$ dataset is highlighted in bold.

	AI	Liter.	Music	Politics	Science	FabNER	GENIA	GENTLE	GUM	i2b2	Movie	Restau.	WNUT 17	Avg.
k-means (CITRUN)	**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**
GMM	25.5	39.9	28.5	30.6	32.2	7.4	**45.8**	6.8	26.8	46.0	32.6	24.0	33.5	29.2
OPTICS	20.4	16.5	22.2	20.5	23.4	9.1	8.5	10.3	8.1	10.3	10.6	12.5	12.5	14.2
HDBSCAN	24.4	21.3	27.2	25.1	29.0	11.8	9.8	12.3	10.0	14.4	37.0	20.5	15.6	19.9

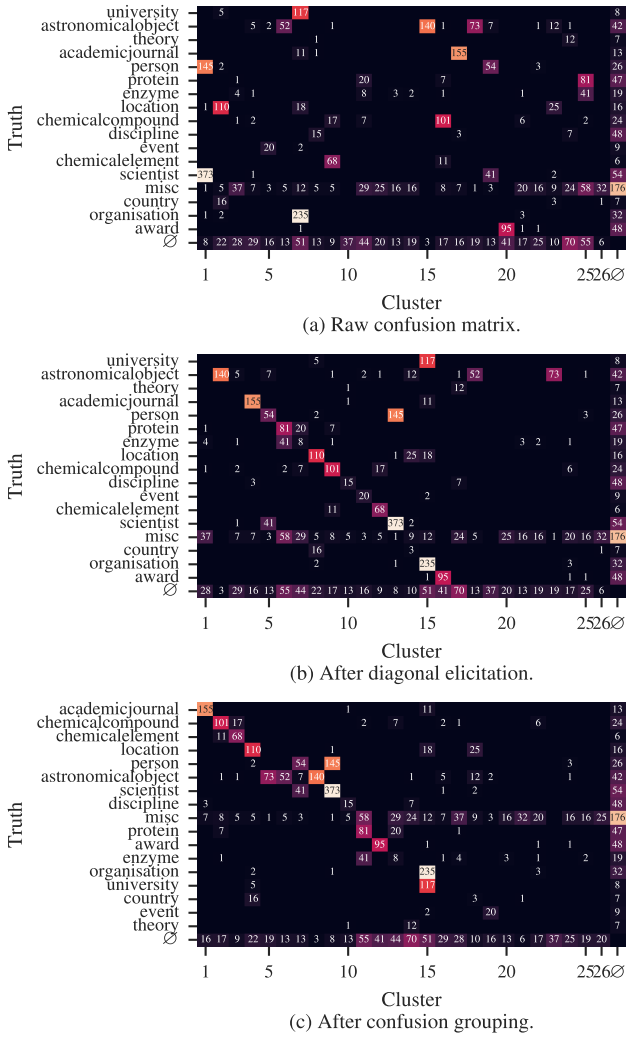
TABLE 15. Statistics of the $\mathcal{D}_S$ and $\mathcal{D}_T$ datasets in our benchmark.

	Dataset	Domain	# Documents Train	Dev	Test	# Types	Avg. words	Avg. entities
$\mathcal{D}_S$	CoNLL [1]	general	946	216	231	4	216	25.2
	Pile-NER [8]	various	44,660	0	0	11,930	210	19.5
$\mathcal{D}_T$	AI [3]	scientific	100	350	431	14	31	4.4
	Liter. [3]	literature	100	400	416	12	38	5.4
	Music [3]	music	100	380	465	13	41	7.1
	Politics [3]	politics	200	541	651	9	44	6.5
	Science [3]	scientific	200	450	543	17	36	5.6
	FabNER [77]	scientific	9,435	2,183	2,064	12	26	4.7
	GENIA [78]	biomedical	1,599	190	213	5	235	28.0
	GENTLE [80]	various	0	0	26	10	684	119.0
	GUM [81]	various	0	0	235	10	971	141.8
	i2b2 [79]	biomedical	521	269	514	23	854	22.1
	Movie [76]	social networks	9,775	0	2,443	12	10	2.2
	Restau. [76]	social networks	7,660	0	1,521	8	9	2.0
	WNUT 17 [82]	social networks	3,394	1,009	1,287	6	18	0.7

a backbone. All the evaluated encoder embeddings lead to better performance than UniNER Uns and ChatIE Uns. Interestingly, the older model BERT remains competitive, performing similarly to more recent alternatives.

For mention detection, DeBERTa v3 outperforms the other approaches, with an average gap of 1.2 % compared to the second-best model, RoBERTa. We attribute this superior performance to DeBERTa v3's richer and broader pre-training dataset compared to the other encoders. BERT shows the weakest performance, which supports our choice of DeBERTa v3 as the backbone for MD.

For entity typing, the performance is closer among BERT, RoBERTa, and ERNIE, which are nearly indistinguishable, especially considering the standard deviation. ELECTRA



(a) Raw confusion matrix.

(b) After diagonal elicitation.

(c) After confusion grouping.

FIGURE 11. *Visualization of the different reordering steps used to reorganize the rows and columns of OWNER's confusion matrices. OWNER was trained on Pile-NER and tested on Science.*

and DeBERTa v3 have lower AMI scores. The behavior of DeBERTa v3 is surprising, as it is generally regarded as the best-performing encoder currently available. DeBERTa v3's performance is even worse without embedding refinement (not shown), achieving only half of BERT's performance without embedding refinement. The same conclusion applies to ELECTRA. DeBERTa v3 and ELECTRA appear 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 also have been valid choices.

APPENDIX C
CLUSTERING ALGORITHMS COMPARISON

As presented in Sect. III-B, OWNER employs k-means to cluster entity embeddings, using a heuristic to estimate the number of clusters. In this section, we evaluate the performance of other clustering algorithms in comparison to k-means.

We focus on clustering models that do not require a predefined number of clusters. Available models include DBSCAN [100], OPTICS [71], HDBSCAN [70], affinity propagation [101], Agglomerative Clustering (HAC) [102], and Gaussian Mixture Models (GMM) using BIC [69]. However, HAC and DBSCAN depend on hyperparameters (such as density) that are difficult to adjust without prior knowledge of k, and affinity propagation does not scale well for large datasets. Therefore, we limit our evaluation to HDBSCAN, OPTICS, and GMM. HDBSCAN and OPTICS are density-based, allowing nonspherical clusters. GMM generalizes k-means by providing continuous cluster assignments (soft clustering) and forming ellipsoidal clusters.

In Table 14, we present the entity typing performance of these clustering algorithms when OWNER is trained on Pile-NER. K-means surpasses the other clustering algorithms, with an average AMI gap of 19 % compared to GMM, 28 % compared to HDBSCAN, and 34 % compared to OPTICS.

The performance gap is particularly pronounced with HDBSCAN and OPTICS, suggesting that density-based clustering is not suitable for entity-type clustering. Genest et al. [14] reached a similar conclusion. The primary issue is that these methods significantly overestimate the number of clusters, resulting in poor performance.

Interestingly, although GMM is generally considered an improved version of k-means, it yields lower results than k-means across our 13 $\mathcal{D}_T$ datasets. The issue is the opposite: GMM tends to underestimate the number of clusters. The only exception is the GENIA dataset, where GMM outperforms k-means. This is likely because GENIA has only four entity types, aligning well with GMM's tendency to underestimate k.

The conclusion of this experiment is that, despite its simplicity, k-means is the best-performing clustering algorithm. We believe its simplicity and low complexity are advantageous in our unsupervised setting.

APPENDIX D
UNSUPERVISED CONFUSION MATRIX

A useful tool for qualitatively analyzing the performance of a classifier is the confusion matrix [103]. Each row of the confusion matrix represents instances in an actual class (e.g., entity type), and each column represents instances in a predicted class. Thus, the matrix's diagonal displays correctly predicted instances, while the lower and upper triangles show errors (also called confusions).

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

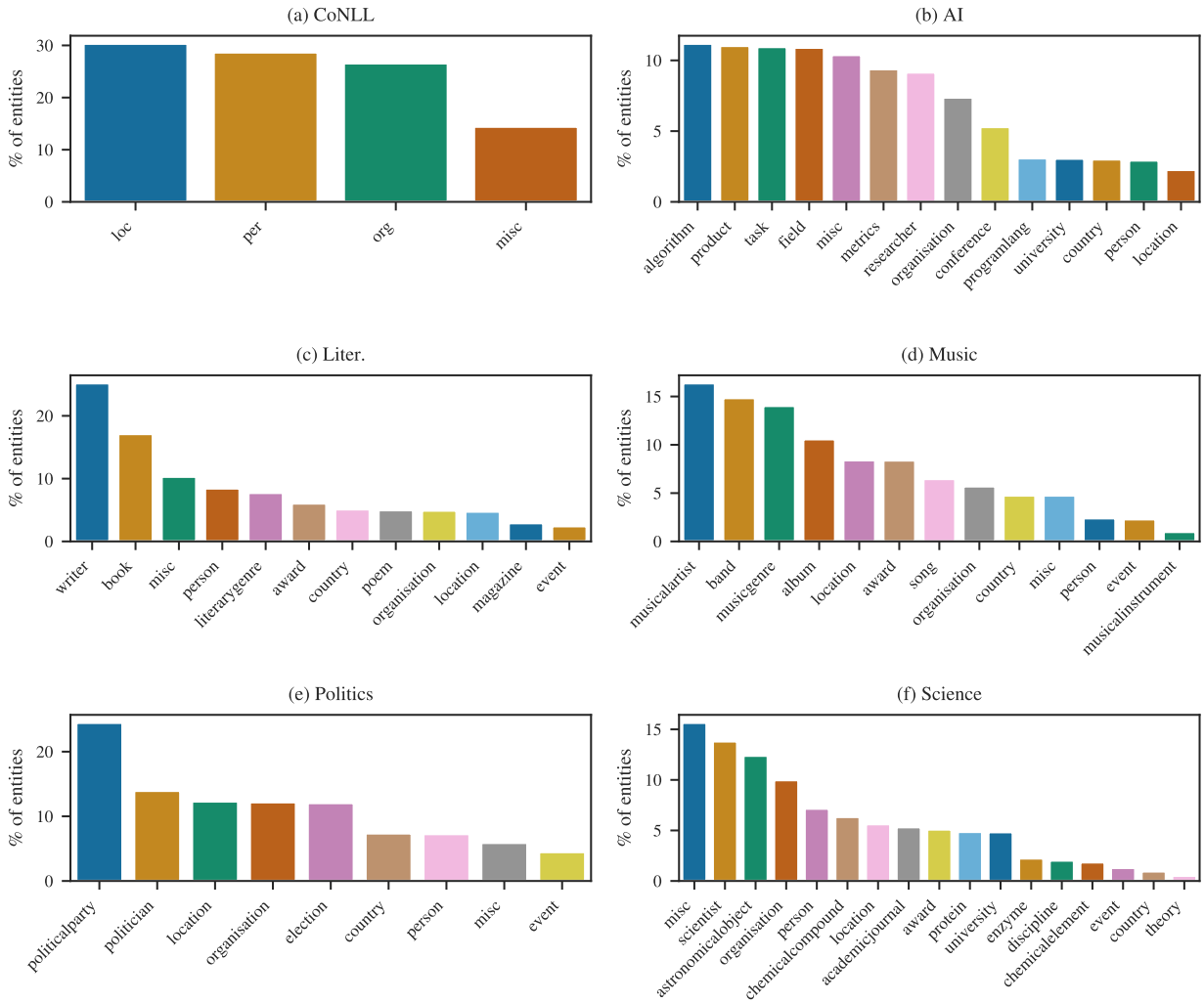


FIGURE 12. $\mathcal{D}_S$ and $\mathcal{D}_T$ dataset entity type distributions (I).

This appendix details the method used to reorder the rows and columns. We use the example of the first figure of Fig. 10 (OWNER trained on Pile-NER and tested on Science). The initial confusion matrix, without processing, is displayed in Fig. 11 (a). It resembles a starry sky more than a confusion matrix and is nearly impossible to interpret.

A. 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 on the diagonal as the model correctly predicts them. Similarly, we want to reorder the axes so that the unsupervised confusion matrix shows a clear diagonal: we aim to find the main cluster corresponding to each class. For instance, in Fig. 11 (a), most instances of *organization* are in cluster 16, and 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 the Jonker-Volgenant algorithm[14] [104], [105].

The resulting confusion matrix is displayed in Fig. 11 (b). It displays a clear diagonal that is much more interpretable than the initial confusion matrix. Nevertheless, some important values outside the diagonal remain scattered (e.g., *person*/cluster 14, *astronomicalobject*/cluster 22).

B. CONFUSION GROUPING

The second step aims to bring major confusions closer to make the matrix more readable. An ideal confusion matrix is a band matrix, which is a sparse matrix where the non-zero entries are confined to a diagonal band. We propose implementing the reverse Cuthill-McKee algorithm[15] [106],

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

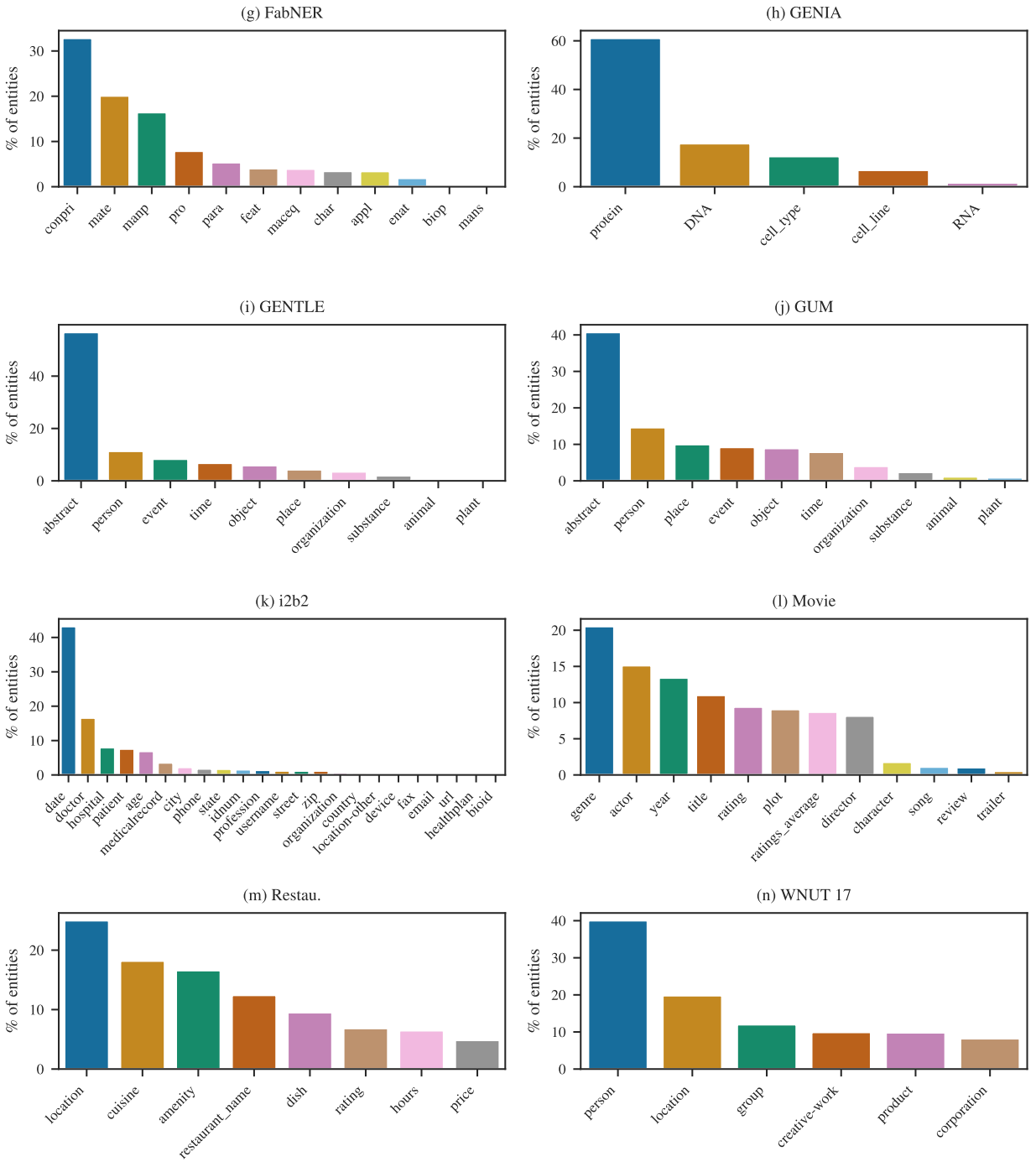


FIGURE 13. $\mathcal{D}_S$ and $\mathcal{D}_T$ dataset entity type distributions (II).

[107], which aims to permute a sparse matrix into a band matrix with a small bandwidth. In practice, not all non-zero values are significant (some represent noise or very rare edge cases), so we propose setting a threshold (1 % of the total instances). Below this threshold, the value is not considered when reordering axes.

We obtain the final confusion matrix in Fig. 11 (c). We can see that the major confusions are now grouped closer (e.g., *person* and *scientist*, *protein* and *enzyme*, *university* and *organization*).

It is worth noting that the first diagonal elicitation step is optional, as the reverse Cuthill-McKee algorithm produces

a band matrix (i.e., with a diagonal). In practice, we have found that this initial diagonal elicitation step helps produce a diagonal with the maximum sum, leading to a clearer interpretation.

APPENDIX E
DATASET STATISTICS

We present the $\mathcal{D}_S$ and $\mathcal{D}_T$ dataset statistics in Table 15. Entity type distributions are shown in Figs. 12 and 13.

REFERENCES

[1] E. F. T. K. Sang and F. De Meulder, "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition," in *Proc. 7th Conf. Natural Lang. Learn.*, vol. 4, 2003, pp. 142–147, doi: 10.3115/1119176.1119195.

[2] Z. Zhong and D. Chen, "A frustratingly easy approach for entity and relation extraction," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol.*, 2021, pp. 50–61.

[3] Z. Liu, Y. Xu, T. Yu, W. Dai, Z. Ji, S. Cahyawijaya, A. Madotto, and P. Fung, "CrossNER: Evaluating cross-domain named entity recognition," in *Proc. 35th AAAI Conf.*, vol. 15, Dec. 2021, pp. 13452–13460.

[4] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. 2019 Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Language Technologies*, vol. 1. Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 4171–4186.

[5] J. Wang, C. Wang, C. Tan, M. Qiu, S. Huang, J. Huang, and M. Gao, "SpanProto: A two-stage span-based prototypical network for few-shot named entity recognition," in *Proc. Conf. Empirical Methods Natural Lang. Process.* Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, 2022, pp. 3466–3476. [Online]. Available: https://aclanthology.org/2022.emnlp-main.227

[6] J. Fang, X. Wang, Z. Meng, P. Xie, F. Huang, and Y. Jiang, "MANNER: A variational memory-augmented model for cross domain few-shot named entity recognition," in *Proc. 61st Annu. Meeting Assoc. Comput. Linguistics*. Toronto, ONT, Canada: Association for Computational Linguistics, 2023, pp. 4261–4276. [Online]. Available: https://aclanthology.org/2023.acl-long.234

[7] Q. Peng, C. Zheng, Y. Cai, T. Wang, H. Xie, and Q. Li, "Unsupervised cross-domain named entity recognition using entity-aware adversarial training," *Neural Netw.*, vol. 138, pp. 68–77, Jun. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608020304524

[8] W. Zhou, S. Zhang, Y. Gu, M. Chen, and H. Poon, "UniversalNER: Targeted distillation from large language models for open named entity recognition," in *Proc. 12th Int. Conf. Learn. Represent.*, Vienna, Austria, 2024. [Online]. Available: https://openreview.net/forum?id=r65xfUb76p

[9] T. B. Brown et al., "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates, Jan. 2020, pp. 1877–1901. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html

[10] U. Zaratiana, N. Tomeh, P. Holat, and T. Charnois, "GLiNER: Generalist model for named entity recognition using bidirectional transformer," 2023, *arXiv:2311.08526*.

[11] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd, "spaCy: Industrial-strength natural language processing in Python," Explosion AI, Berlin, Germany, Tech. Rep., 2020.

[12] B. M. Sundheim, "Overview of the third message understanding evaluation and conference," in *Proc. 3rd Conf. Message Understand.*, San Diego, CA, USA, 1991, p. 3.

[13] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni, "Open information extraction from the web," in *Proc. 20th Int. Joint Conf. Artif. Intell.* Hyderabad, India: Morgan Kaufmann, 2007, pp. 2670–2676.

[14] P.-Y. Genest, P.-E. Portier, E. Egyed-Zsigmond, and L.-W. Goix, "PromptORE—A novel approach towards fully unsupervised relation extraction," in *Proc. 31st ACM Int. Conf. Inf. Knowl. Manage.* Atlanta, GA, USA: Association for Computing Machinery, Oct. 2022, pp. 561–571. [Online]. Available: https://hal.science/hal-03858264

[15] X. Hu, L. Wen, Y. Xu, C. Zhang, and P. Yu, "SelfORE: Self-supervised relational feature learning for open relation extraction," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 3673–3682.

[16] J. Zhao, T. Gui, Q. Zhang, and Y. Zhou, "A relation-oriented clustering method for open relation extraction," in *Proc. Conf. Empirical Methods Natural Lang. Process.* Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021, pp. 9707–9718.

[17] Y. Luo, H. Zhao, and J. Zhan, "Named entity recognition only from word embeddings," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 8995–9005. [Online]. Available: https://aclanthology.org/2020.emnlp-main.723

[18] G. Schwarz, "Estimating the dimension of a model," *Ann. Statist.*, vol. 6, no. 2, pp. 461–464, Mar. 1978. [Online]. Available: https://www.jstor.org/stable/2958889

[19] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *Proc. Conf. Empirical Methods Natural Lang. Process., Syst. Demonstrations*. Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 66–71. [Online]. Available: https://aclanthology.org/D18-2012

[20] O. Sainz, I. García-Ferrero, R. Agerri, O. L. D. Lacalle, G. Rigau, and E. Agirre, "GoLLIE: Annotation guidelines improve zero-shot information-extraction," in *Proc. 12th Int. Conf. Learn. Represent.*, Vienna, Austria, Jan. 2024. [Online]. Available: https://openreview.net/forum?id=Y3wpuxd7u9

[21] Y. Shen, Z. Tan, S. Wu, W. Zhang, R. Zhang, Y. Xi, W. Lu, and Y. Zhuang, "PromptNER: Prompt locating and typing for named entity recognition," in *Proc. 61st Annu. Meeting Assoc. Comput. Linguistics*. Toronto, ONT, Canada: Association for Computational Linguistics, 2023, pp. 12492–12507. [Online]. Available: https://aclanthology.org/2023.acl-long.698

[22] J. Achiam et al., "GPT-4 technical report," 2023, *arXiv:2303.08774*.

[23] M. Zhang, H. Yan, Y. Zhou, and X. Qiu, "PromptNER: A prompting method for few-shot named entity recognition via K nearest neighbor search," 2023, *arXiv:2305.12217*.

[24] T. Ma, H. Jiang, Q. Wu, T. Zhao, and C.-Y. Lin, "Decomposed meta-learning for few-shot named entity recognition," in *Proc. Findings Assoc. Comput. Linguistics*. Dublin, Ireland: Association for Computational Linguistics, 2022, pp. 1584–1596. [Online]. Available: https://aclanthology.org/2022.findings-acl.124

[25] S. S. S. Das, A. Katiyar, R. Passonneau, and R. Zhang, "CONTaiNER: Few-shot named entity recognition via contrastive learning," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*. Dublin, Ireland: Association for Computational Linguistics, 2022, pp. 6338–6353. [Online]. Available: https://aclanthology.org/2022.acl-long.439

[26] Y. Huang, K. He, Y. Wang, X. Zhang, T. Gong, R. Mao, and C. Li, "COPNER: Contrastive learning with prompt guiding for few-shot named entity recognition," in *Proc. 29th Int. Conf. Comput. Linguistics (COLING)*. Gyeongju, Republic of Korea: International Committee on Computational Linguistics, Oct. 2022, pp. 2515–2527. [Online]. Available: https://aclanthology.org/2022.coling-1.222

[27] G. Dong, Z. Wang, J. Zhao, G. Zhao, D. Guo, D. Fu, T. Hui, C. Zeng, K. He, X. Li, L. Wang, X. Cui, and W. Xu, "A multi-task semantic decomposition framework with task-specific pre-training for few-shot NER," in *Proc. 32nd ACM Int. Conf. Inf. Knowl. Manage.* New York, NY, USA: Association for Computing Machinery, Oct. 2023, pp. 430–440, doi: 10.1145/3583780.3614766.

[28] P. Fuchun, "Accurate information extraction from research papers using conditional random fields," in *Proc. HLT/NAACL*, 2004.

[29] V. Dobrovolskii, "Word-level coreference resolution," in *Proc. Conf. Empirical Methods Natural Lang. Process.* Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021, pp. 7670–7675. [Online]. Available: https://aclanthology.org/2021.emnlp-main.605

[30] U. Zaratiana, N. Tomeh, P. Holat, and T. Charnois, "Named entity recognition as structured span prediction," in *Proc. Workshop Unimodal Multimodal Induction Linguistic Struct. (UM-IoS)*, W. Han, Z. Zheng, Z. Lin, L. Jin, Y. Shen, Y. Kim, and K. Tu, Eds., Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, 2022, pp. 1–10. [Online]. Available: https://aclanthology.org/2022.umios-1.1

[31] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, Mar. 2017, pp. 4078–4088.

[32] N. Ding, S. Hu, W. Zhao, Y. Chen, Z. Liu, H. Zheng, and M. Sun, "OpenPrompt: An open-source framework for prompt-learning," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics, Syst. Demonstrations*, V. Basile, Z. Kozareva, and S. Stajner, Eds., Dublin, Ireland: Association for Computational Linguistics, 2022, pp. 105–113. [Online]. Available: https://aclanthology.org/2022.acl-demo.10

[33] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. 34th Int. Conf. Mach. Learn.*, Jul. 2017, pp. 1126–1135. [Online]. Available: https://proceedings.mlr.press/v70/finn17a.html

[34] A. Mahapatra, S. R. Nangi, A. Garimella, and N. Anandhavelu, "Entity extraction in low resource domains with selective pre-training of large language models," in *Proc. Conf. Empirical Methods Natural Lang. Process.* Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, 2022, pp. 942–951. [Online]. Available: https://aclanthology.org/2022.emnlp-main.61

[35] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou, "Self-consistency improves chain of thought reasoning in language models," in *Proc. 11th Int. Conf. Learn. Represent.*, Kigali, Rwanda, Mar. 2023.

[36] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "LLaMA: Open and efficient foundation language models," 2023, *arXiv:2302.13971*.

[37] S. Wang, X. Sun, X. Li, R. Ouyang, F. Wu, T. Zhang, J. Li, and G. Wang, "GPT-NER: Named entity recognition via large language models," 2023, *arXiv:2304.10428*.

[38] J. Ye, X. Chen, N. Xu, C. Zu, Z. Shao, S. Liu, Y. Cui, Z. Zhou, C. Gong, Y. Shen, J. Zhou, S. Chen, T. Gui, Q. Zhang, and X. Huang, "A comprehensive capability analysis of GPT-3 and GPT-3.5 series models," 2023, *arXiv:2303.10420*.

[39] X. Wei, X. Cui, N. Cheng, X. Wang, X. Zhang, S. Huang, P. Xie, J. Xu, Y. Chen, M. Zhang, Y. Jiang, and W. Han, "ChatIE: Zero-shot information extraction via chatting with ChatGPT," 2023, *arXiv:2302.10205*.

[40] T. Xie, Q. Li, J. Zhang, Y. Zhang, Z. Liu, and H. Wang, "Empirical study of zero-shot NER with ChatGPT," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, H. Bouamor, J. Pino, and K. Bali, Eds., Singapore: Association for Computational Linguistics, 2023, pp. 7935–7956.

[41] T. Xie, Q. Li, Y. Zhang, Z. Liu, and H. Wang, "Self-improving for zero-shot named entity recognition with large language models," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol.*, K. Duh, H. Gomez, and S. Bethard, Eds., Mexico City, Mexico: Association for Computational Linguistics, 2024, pp. 583–593. [Online]. Available: https://aclanthology.org/2024.naacl-short.49

[42] OpenAI. (2022). *ChatGPT—Release Notes | OpenAI Help Center*. [Online]. Available: https://help.openai.com/en/articles/6825453-chatgpt-release-notes

[43] X. Wang, W. Zhou, C. Zu, H. Xia, T. Chen, Y. Zhang, R. Zheng, J. Ye, Q. Zhang, T. Gui, J. Kang, J. Yang, S. Li, and C. Du, "InstructUIE: Multi-task instruction tuning for unified information extraction," 2023, *arXiv:2304.08085*.

[44] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, and J. E. Gonzalez, (Mar. 2023). *Vicuna: An Open-Source ChatBOT Impressing GPT-4 With 90% ChatGPT Quality*. [Online]. Available: https://lmsys.org/blog/2023-03-30-vicuna

[45] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. D. L. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. Le Scao, T. Lavril, T. Wang, T. Lacroix, and W. El Sayed, "Mistral 7B," 2023, *arXiv:2310.06825*.

[46] L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima, S. Presser, and C. Leahy, "The pile: An 800GB dataset of diverse text for language modeling," 2021, *arXiv:2101.00027*.

[47] B. Rozière et al., "Code llama: Open foundation models for code," 2023, *arXiv:2308.12950*.

[48] Z. Li, Y. Zeng, Y. Zuo, W. Ren, W. Liu, M. Su, Y. Guo, Y. Liu, X. Li, Z. Hu, L. Bai, W. Li, Y. Liu, P. Yang, X. Jin, J. Guo, and X. Cheng, "KnowCoder: Coding structured knowledge into LLMs for universal information extraction," 2024, *arXiv:2403.07969*.

[49] Y. Ding, J. Li, P. Wang, Z. Tang, B. Yan, and M. Zhang, "Rethinking negative instances for generative named entity recognition," 2024, *arXiv:2402.16602*.

[50] P. He, J. Gao, and W. Chen, "DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing," in *Proc. 11th Int. Conf. Learn. Represent.*, Kigali, Rwanda, 2021. [Online]. Available: https://openreview.net/forum?id=sE7-XhLxHA

[51] H. W. Chung et al., "Scaling instruction-finetuned language models," *J. Mach. Learn. Res.*, vol. 25, no. 70, pp. 1–53, Jan. 2022. [Online]. Available: http://jmlr.org/papers/v25/23-0870.html

[52] Y. Sun, S. Wang, Y. Li, S. Feng, X. Chen, H. Zhang, X. Tian, D. Zhu, H. Tian, and H. Wu, "ERNIE: Enhanced representation through knowledge integration," 2019, *arXiv:1904.09223*.

[53] L. Baldini Soares, N. FitzGerald, J. Ling, and T. Kwiatkowski, "Matching the blanks: Distributional similarity for relation learning," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*. Florence, Italy: Association for Computational Linguistics, 2019, pp. 2895–2905. [Online]. Available: https://aclanthology.org/P19-1279

[54] Y. Wang, C. Sun, Y. Wu, H. Zhou, L. Li, and J. Yan, "ENPAR: Enhancing entity and entity pair representations for joint entity relation extraction," in *Proc. 16th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2021, pp. 2877–2887. [Online]. Available: https://aclanthology.org/2021.eacl-main.251

[55] S. Bogdanov, A. Constantin, T. Bernard, B. Crabbé, and E. Bernard, "NuNER: Entity recognition encoder pre-training via LLM-annotated data," 2024, *arXiv:2402.15343*.

[56] L. Peng, Z. Wang, F. Yao, Z. Wang, and J. Shang, "MetaIE: Distilling a meta model from LLM for all kinds of information extraction tasks," 2024, *arXiv:2404.00457*.

[57] J. Lou, Y. Lu, D. Dai, W. Jia, H. Lin, X. Han, L. Sun, and H. Wu, "Universal information extraction as unified semantic matching," in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2023, vol. 37, no. 11, pp. 13318–13326. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/3

[58] A. Cucchiarelli and P. Velardi, "Unsupervised named entity recognition using syntactic and semantic contextual evidence," *Comput. Linguistics*, vol. 27, no. 1, pp. 123–131, Mar. 2001, doi: 10.1162/089120101300346822.

[59] D. Nadeau, P. Turney, and S. Matwin, "Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity," in *Advances in Artificial Intelligence* (Lecture Notes in Computer Science). Berlin, Germany: Springer, 2006, pp. 266–277.

[60] C. Jia, X. Liang, and Y. Zhang, "Cross-domain NER using cross-domain language modeling," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*. Florence, Italy: Association for Computational Linguistics, 2019, pp. 2464–2474. [Online]. Available: https://aclanthology.org/P19-1236

[61] Z. Liu, G. I. Winata, and P. Fung, "Zero-resource cross-domain named entity recognition," in *Proc. 5th Workshop Represent. Learn.*, S. Gella, J. Welbl, M. Rei, F. Petroni, P. Lewis, E. Strubell, M. Seo, and H. Hajishirzi, Eds., 2020, pp. 1–6. [Online]. Available: https://aclanthology.org/2020.repl4nlp-1.1

[62] A. Iovine, A. Fang, B. Fetahu, O. Rokhlenko, and S. Malmasi, "CycleNER: An unsupervised training approach for named entity recognition," in *Proc. ACM Web Conf.* New York, NY, USA: Association for Computing Machinery, Apr. 2022, pp. 2916–2924, doi: 10.1145/3485447.3512012.

[63] E. Perez, D. Kiela, and K. Cho, "True few-shot learning with language models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34. Red Hook, NY, USA: Curran Associates, Jan. 2021, pp. 11054–11070. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/5c04925674920eb58467fb52ce4ef728-Abstract.html

[64] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, CA, CA, USA, Jun. 2005, pp. 539–546.

[65] L. Cui, Y. Wu, J. Liu, S. Yang, and Y. Zhang, "Template-based named entity recognition using BART," in *Proc. Findings Assoc. Comput. Linguistics*, 2021, pp. 1835–1845. [Online]. Available: https://aclanthology.org/2021.findings-acl.161

[66] T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace, and S. Singh, "AutoPrompt: Eliciting knowledge from language models with automatically generated prompts," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 4222–4235.

[67] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982, doi: 10.1109/TIT.1982.1056489.

[68] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, vol. 1. Berkeley, CA, USA: Univ. California Press, Jan. 1967, pp. 281–297.

[69] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. Hoboken, NJ, USA: Wiley, 2000.

[70] R. J. G. B. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," in *Advances in Knowledge Discovery and Data Mining*, J. Pei, V. S. Tseng, L. Cao, H. Motoda, and G. Xu, Eds., Berlin, Germany: Springer, 2013, pp. 160–172.

[71] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: Ordering points to identify the clustering structure," *ACM SIGMOD Rec.*, vol. 28, no. 2, pp. 49–60, Jun. 1999, doi: 10.1145/304181.304187.

[72] A. J. Onumanyi, D. N. Molokomme, S. J. Isaac, and A. M. Abu-Mahfouz, "AutoElbow: An automatic elbow detection method for estimating the number of clusters in a dataset," *Appl. Sci.*, vol. 12, no. 15, p. 7515, Jul. 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/15/7515

[73] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, "Large scale online learning of image similarity through ranking," *J. Mach. Learn. Res.*, vol. 11, pp. 1109–1135, Mar. 2010. [Online]. Available: https://www.jmlr.org/papers/volume11/chechik10a/chechik10a.pdf

[74] C. Chen, J. Zhang, Y. Xu, L. Chen, J. Duan, Y. Chen, S. Tran, B. Zeng, and T. Chilimbi, "Why do we need large batchsizes in contrastive learning? A gradient-bias perspective," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, Dec. 2022, pp. 33860–33875.

[75] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, pp. 1–67, Jan. 2019. [Online]. Available: http://jmlr.org/papers/v21/20-074.html

[76] J. Liu, P. Pasupat, S. Cyphers, and J. Glass, "Asgard: A portable architecture for multilingual dialogue systems," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 8386–8390. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6639301

[77] A. Kumar and B. Starly, "'FabNER': Information extraction from manufacturing process science domain literature using named entity recognition," *J. Intell. Manuf.*, vol. 33, no. 8, pp. 2393–2407, Dec. 2022, doi: 10.1007/s10845-021-01807-x.

[78] J.-D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii, "GENIA corpus—A semantically annotated corpus for bio-textmining," *Bioinformatics*, vol. 19, pp. 180–182, Jul. 2003.

[79] A. Stubbs and Ö. Uzuner, "Annotating longitudinal clinical narratives for de-identification: The 2014 i2b2/UTHealth corpus," *J. Biomed. Informat.*, vol. 58, pp. 20–29, Dec. 2015.

[80] T. Aoyama, S. Behzad, L. Gessler, L. Levine, J. Lin, Y. J. Liu, S. Peng, Y. Zhu, and A. Zeldes, "GENTLE: A genre-diverse multilayer challenge set for English NLP and linguistic evaluation," in *Proc. 17th Linguistic Annotation Workshop (LAW-XVII)*. Toronto, ONT, Canada: Association for Computational Linguistics, 2023, pp. 166–178. [Online]. Available: https://aclanthology.org/2023.law-1.17

[81] A. Zeldes, "The GUM corpus: Creating multilayer resources in the classroom," *Lang. Resour. Eval.*, vol. 51, no. 3, pp. 581–612, Sep. 2017, doi: 10.1007/s10579-016-9343-x.

[82] L. Derczynski, E. Nichols, M. van Erp, and N. Limsopatham, "Results of the WNUT2017 shared task on novel and emerging entity recognition," in *Proc. 3rd Workshop Noisy User-Generated Text*, L. Derczynski, W. Xu, A. Ritter, and T. Baldwin, Eds., Copenhagen, Denmark: Association for Computational Linguistics, 2017, pp. 140–147. [Online]. Available: https://aclanthology.org/W17-4418

[83] L. Hubert and P. Arabie, "Comparing partitions," *J. Classification*, vol. 2, no. 1, pp. 193–218, Dec. 1985. [Online]. Available: https://link.springer.com/article/10.1007/BF01908075

[84] D. Steinley, "Properties of the Hubert-Arabie adjusted Rand index," *Psychol. Methods*, vol. 9, no. 3, pp. 386–396, Sep. 2004. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/15355155/

[85] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul. 1948. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6773024

[86] A. Rosenberg and J. Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure," in *Proc. Joint Conf. Empirical Methods Natural Language Process. Comput. Natural Language Learn.* Prague, Czech Republic: Association for Computational Linguistics, Jun. 2007, pp. 410–420.

[87] A. Bagga and B. Baldwin, "Algorithms for scoring coreference chain," in *Proc. 1st Int. Conf. Language Resources Eval. Workshop Linguistics Conf.* Granada, Spain: European Language Resources Association, 1998, pp. 563–566.

[88] S. Romano, N. X. Vinh, J. Bailey, and K. Verspoor, "Adjusting for chance clustering comparison measures," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 4635–4666, Jan. 2016, doi: 10.5555/2946645.3007087.

[89] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Is a correction for chance necessary?" in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, Jun. 2009, pp. 1073–1080.

[90] P. K. Pushp and M. M. Srivastava, "Train once, test anywhere: Zero-shot learning for text classification," 2017, *arXiv:1712.05972*.

[91] P. He, X. Liu, J. Gao, and W. Chen, "DeBERTa: Decoding-enhanced BERT with disentangled attention," in *Proc. 9th Int. Conf. Learn. Represent.*, Jan. 2020. [Online]. Available: https://openreview.net/forum?id=XPZIaotutsD

[92] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, San Diego, CA, USA, 2015.

[93] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.

[94] D. Shen, J. Zhang, J. Su, G. Zhou, and C.-L. Tan, "Multi-criteria-based active learning for named entity recognition," in *Proc. 42nd Annu. Meeting Assoc. Comput. Linguistics*, 2004, pp. 589–596.

[95] B. Duan, S. Wang, X. Liu, and Y. Xu, "Cluster-aware pseudo-labeling for supervised open relation extraction," in *Proc. 29th Int. Conf. Comput. Linguistics*, N. Calzolari, C.-R. Huang, H. Kim, J. Pustejovsky, L. Wanner, K.-S. Choi, P.-M. Ryu, H.-H. Chen, L. Donatelli, H. Ji, S. Kurohashi, P. Paggio, N. Xue, S. Kim, Y. Hahm, Z. He, T. K. Lee, E. Santus, F. Bond, and S.-H. Na, Eds., Gyeongju, Republic of Korea: International Committee on Computational Linguistics, Oct. 2022, pp. 1834–1841. [Online]. Available: https://aclanthology.org/2022.coling-1.158

[96] J. Zhao, Y. Zhang, Q. Zhang, T. Gui, Z. Wei, M. Peng, and M. Sun, "Actively supervised clustering for open relation extraction," in *Proc. 61st Annu. Meeting Assoc. Comput. Linguistics*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds., Toronto, ONT, Canada: Association for Computational Linguistics, 2023, pp. 4985–4997. [Online]. Available: https://aclanthology.org/2023.acl-long.273

[97] L. Ramshaw and M. Marcus, "Text chunking using transformation-based learning," in *Natural Language Processing Using Very Large Corpora* (Text, Speech and Language Technology), S. Armstrong, K. Church, P. Isabelle, S. Manzi, E. Tzoukermann, and D. Yarowsky, Eds., Dordrecht, The Netherlands: Springer, 1999, pp. 157–176, doi: 10.1007/978-94-017-2390-9_10.

[98] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.

[99] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "ELECTRA: Pre-training text encoders as discriminators rather than generators," in *Proc. 7th Int. Conf. Learn. Represent.*, New Orleans, LA, USA, Sep. 2019. [Online]. Available: https://openreview.net/forum?id=r1xMH1BtvB

[100] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowledge Discovery Data Mining*, Portland, OR, USA, 1996, pp. 226–231. [Online]. Available: www.aaai.org

[101] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, Feb. 2007, doi: 10.1126/science.1136800.

[102] R. Sibson, "SLINK: An optimally efficient algorithm for the single-link cluster method," *Comput. J.*, vol. 16, no. 1, pp. 30–34, Jan. 1973. [Online]. Available: https://academic.oup.com/comjnl/article/16/1/30/434805

[103] K. Pearson and J. Blakeman, *On the Theory of Contingency and Its Relation to Association and Normal Correlation* (Mathematical Contributions to the Theory of Evolution), vol. 13. London, U.K.: Dulau, 1904.

[104] R. Jonker and T. Volgenant, "A shortest augmenting path algorithm for dense and sparse linear assignment problems," in *Proc. 16th Annu. Meeting DGOR Cooperation NSOR/Vorträge der Jahrestagung der DGOR Zusammen Mit der NSOR*. Berlin, Germany: Springer, 1988, p. 622.

[105] D. F. Crouse, "On implementing 2D rectangular assignment algorithms," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, no. 4, pp. 1679–1696, Aug. 2016. [Online]. Available: https://ieeexplore.ieee.org/document/7738348

[106] E. Cuthill and J. McKee, "Reducing the bandwidth of sparse symmetric matrices," in *Proc. 24th Nat. Conf.* New York, NY, USA: Association for Computing Machinery, 1969, pp. 157–172, doi: 10.1145/800195.805928.
[107] A. George and J. W. Liu, *Computer Solution of Large Sparse Positive Definite*. Upper Saddle River, NJ, USA: Prentice-Hall, 1981.

PIERRE-YVES GENEST was born in France. He received the M.Sc. degree in engineering and computer science from the Computer Science Department, Institut National des Sciences Appliquées of Lyon (INSA Lyon), France, in 2021, where he is currently pursuing the Ph.D. degree in natural language processing with the LIRIS Laboratory.

He is a Data Scientist with Alteca, Villeurbanne, France, where he leads research and development and innovation projects in NLP and generative AI. He has published papers in prominent conferences, such as CIKM and SIGIR. His research interests include information extraction from documents, structured information extraction, knowledge graphs, generative AI, and natural language processing.

PIERRE-EDOUARD PORTIER received the Ph.D. degree in computer science from INSA Lyon, France, in 2010.

He has been the Lead Data Scientist with Caisse d'Epargne Rhône Alpes, France, since September 2023. From 2011 to 2023, he was an Associate Professor with INSA Lyon, within the Computer Science and Information Technologies Department and the LIRIS Laboratory. His research focuses on machine learning and soft computing techniques applied to big data analytics, such as anomaly detection, XAI for crash prediction, and traffic forecasting. Additionally, he works on natural language processing, including relation extraction, knowledge graph completion, semantic web search, and document engineering. His research interests include discovering, modeling, and representing knowledge to be integrated into a data analytics process.

ELŐD EGYED-ZSIGMOND received the master's degree in computer science engineering and the Ph.D. degree in computer science from the National Institute of Applied Sciences of Lyon, INSA Lyon, Lyon, France, in1999 and 2003, respectively.

He is currently a habilitated Associate Professor with the Computer Science Department, INSA Lyon, and a Researcher with LIRIS, UMR 5202, CNRS Research Laboratory. His research interests include structured information extraction from documents, information modeling, and natural language processing.

MARTINO LOVISETTO was born in Italy. He received the bachelor's degree (Hons.) in physics from the University of Turin, Italy, in 2016, the master's degree (Hons.) in physics of complex systems, a joint program between Sorbonne Université, Paris, France, and Politecnico di Torino, Italy, in 2018, and the Ph.D. degree in applied mathematics from Université Côte d'Azur, Nice, France, in 2022.

He is currently the Head of the Research and Development and Innovation, Alteca, Lyon, France, where he leads initiatives in natural language processing, generative, and multimodal AI. His research during his Ph.D. focused on computational modeling, quantum physics, and advanced differential equation solvers. His current research interests include large language models, retrieval-augmented generation, and optimization in AI applications.