https://doi.org/10.31891/2219-9365-2024-77-11
UDK 246

DYCHKA Ivan
National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”
https://orcid.org/0000-0002-3446-3076
e-mail: dychka@pzks.fpm.kpi.ua
POTAPOVA Kateryna
National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”
https://orcid.org/0000-0002-3347-6350
e-mail: avatarkina-avatarochka@ukr.net
VOVK Liliya
National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”
https://orcid.org/0000-0002-3098-8078
e-mail: lilyvovk@gmail.com
MELIUKH Vasyl
National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”
https://orcid.org/0009-0009-3783-9954
e-mail: vasylmeliukh430@gmail.com
VEDENIEIEVA Olga
Open International University of Human Development "Ukraine"
https://orcid.org/0009-0006-0941-165X
e-mail: vedenieieva@gmail.com

ADAPTIVE DOMAIN-SPECIFIC NAMED ENTITY RECOGNITION METHOD WITH LIMITED DATA

The ever-evolving volume of digital information requires the development of innovative search strategies aimed at obtaining the necessary data efficiently and economically feasible. The urgency of the problem is emphasized by the growing complexity of information landscapes and the need for fast data extraction methodologies. In the field of natural language processing, named entity recognition (NER) is an essential task for extracting useful information from unstructured text input for further classification into predefined categories. Nevertheless, conventional methods frequently encounter difficulties when confronted with a limited amount of labeled data, posing challenges in real-world scenarios where obtaining substantial annotated datasets is problematic or costly. In order to address the problem of domain-specific NER with limited data, this work investigates NER techniques that can overcome these constraints by continuously learning from newly collected information on pre-trained models. Several techniques are also used for making the greatest use of the limited labeled data, such as using active learning, exploiting unlabeled data, and integrating domain knowledge. Using domain-specific datasets with different levels of annotation scarcity, the fine-tuning process of pre-trained models, such as transformer-based models (TRF) and Toc2Vec (token-to-vector) models is investigated. The results show that, in general, expanding the volume of training data enhances most models' performance for NER, particularly for models with sufficient learning ability. Depending on the model architecture and the complexity of the entity label being learned, the effect of more data on the model's performance can change. After increasing the training data by 20%, the LT2V model shows the most balanced growth in accuracy overall by 11% recognizing 73% of entities and processing speed. Meanwhile, with consistent processing speed and the greatest F1-score, the Transformer-based model (TRF) shows promise for effective learning with less data, achieving 74% successful prediction and a 7% increase in performance after expanding the training data to 81%. Our results pave the way for the creation of more resilient and efficient NER systems suited to specialized domains and further the field of domain-specific NER with sparse data. We also shed light on the relative merits of various NER models and training strategies, and offer perspectives for future research.

Keywords: named entity recognition (NER), adaptive learning, domain-specific NER, information extraction, natural language processing (NLP), feature engineering.

ДИЧКА Іван, ПОТАПОВА Катерина, ВОВК Лілія, МЕЛЮХ Василь
Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського"
ВЕДЕНЄЄВА Ольга
Відкритий міжнародний університет розвитку людини "Україна"

МЕТОД АДАПТИВНОГО ВИЗНАЧЕННЯ ІМЕНОВАНИХ СУТНОСТЕЙ У СПЕЦІАЛІЗОВАНОМУ ДОМЕНІ З ОБМЕЖЕНИМИ ДАНИМИ

Постійно зростаючий обсяг цифрової інформації вимагає розробки інноваційних стратегій пошуку, спрямованого на ефективне та економічно доцільне отримання необхідних даних. Актуальність проблеми підкреслюється зростаючою складністю інформаційних ландшафтів і потребою в швидких методологіях вилучення даних. У галузі обробки природної мови розпізнавання іменованих об'єктів (NER) є важливим завданням для вилучення інформації з неструктурованих текстових даних для подальшої класифікації у наперед визначені категорії. Тим не менш, традиційні методи розпізнавання об'єктів часто стикаються з труднощами, маючи в розпорядженні обмежену кількістю анотованих даних необхідних для тренування моделі, створюючи проблеми в реальних сценаріях, де отримання обширного набору даних є проблематичним або дорогим. У цій роботі досліджуються методи NER, які можуть подолати ці обмеження шляхом адаптивного довчання на основі попередньо навчених моделей з можливістю ітеративного додавання нових даних. Також застосовуються декілька *технік для отримання найбільше користі від обмеженої кількості анотованих даних, таких як використання активного навчання, немаркованих даних та інтеграцію знань предметної області. Використовуючи предметні набори даних із різними рівнями розподілу між класами сутоностей, досліджується процес тонкого налаштування попередньо навчених моделей, таких як трансформаторні моделі (TRF) і моделі Toc2Vec (токен-вектор). Результати показують, що загалом збільшення обсягу навчальних даних підвищує продуктивність більшості моделей для NER, особливо для моделей з високою здатністю до навчання. Залежно від архітектури моделі та складності мітки сутності, що вивчається, вплив більшої кількості даних на продуктивність моделі може змінюватися. Після збільшення тренувальних даних на 20 % модель LT2V демонструє найбільш збалансоване зростання точності та швидкості обробки в загальному на 11%, розпізнаючи 73% сутностей. Водночас завдяки постійній швидкості обробки та найвищому показнику F1 модель на основі трансформатора (TRF) показує ефективне навчання з меншою кількістю даних, досягаючи 74% успішних передбачень й зростання продуктивності на 7% після розширення тренувальних даних до 81%. Наші результати прокладають шлях для створення більш стійких і ефективних систем NER, які підходять для спеціалізованих доменів, і розвивають галузь доменно-спеціального NER з обмеженими даними. Ми також проливаємо світло на відносні переваги різних моделей NER і стратегій навчання та пропонуємо перспективи майбутніх досліджень.*

Ключові слова: розпізнавання іменованих сутностей (NER), адаптивне навчання, доменно-залежний NER, витяг інформації, обробка природної мови (NLP), конструювання ознак.

Introduction

Named entity recognition (NER) is a core task in natural language processing (NLP) that entails locating and categorizing named entities in a text body, including individuals, locations, organizations, and other designated terms. Accurate performance of the NER is crucial for many downstream applications, including question answering, sentiment analysis, machine translation, and information retrieval [1]. Considering that the performance of complicated NLP tasks is greatly affected by the accuracy of NER systems, research and development in this vital area is imperative. Supervised learning approaches typically yield accurate results for large, annotated datasets. However, in numerous industries, gathering a substantial volume of labeled data is costly and time consuming [2]. Incremental NER techniques are a result of the development of more efficient methods, motivated by the requirement for fewer manual annotation efforts and the ability to learn and improve with the addition of new data [3].

Despite significant advances, NER continues to encounter major challenges in domain-specific settings. Because the subject of study is exceptionally specialized, obtaining a significant volume of manually annotated training data becomes an important roadblock. Training NER models that are reliable, precise, and tailored to the entity types and terminology of a specific domain becomes more challenging when there is insufficient data. General-purpose NER models perform adequately in high-representation domains; however, they lose their efficacy in specialized areas with limited labeled data. These circumstances usually result in the failure of conventional NER algorithms, underscoring the necessity of tailored solutions that can effectively handle data sparsity and adjust to domain-specific nuances [1-4].

Motivated by the demands of domain-specific NER with sparse data, this study presents a strategy centered on the principles of incremental learning. To improve NER performance in particular domains, incremental learning, a methodology in which models dynamically adapt to new data while conserving prior knowledge, offers a viable path. Our technique attempts to bridge the gap between traditional static models and the changing needs of real-world applications by gradually updating the NER model using domain-specific data as it becomes available.

The cornerstone of our research lies in an attempt to integrate domain knowledge by adding domain-specific knowledge resources, such as lexicons, gazetteers, and pre-established rules, and we hope to guide the model to find significant objects in a specialized field. We examined ways to make use of a large amount of publicly available unlabeled domain-specific text data. Additional training signals for the model can be obtained using methods such as retrieving correlated samples from unlabeled data and distant supervision. Another way to improve the adaptability of the model is to investigate active learning techniques to reduce the volume of human labor required for manual annotation. This involves focusing annotation efforts on the most important cases and selecting the most instructional data points for human annotation from the unlabeled pool to improve model performance.

Incremental learning is important in NER because it can reduce the constraints caused by a lack of labeled data. Incremental learning allows Natural English Recognition (NER) systems to adjust to volatile linguistic patterns and domain-specific variations by continuously introducing fresh data into the model. This flexibility is particularly helpful in dynamic contexts where access to annotated material is limited, or where the distribution of named entities may vary over time. We assessed the effectiveness of our proposed method in various domains and compared its results, demonstrating the performance boost of the presented models. We demonstrate the efficacy and robustness of our incremental method in tackling the difficulties associated with domain-specific NER tasks with sparse labeled data through empirical research and comparative studies.

Analysis of research and publications

Named Entity Recognition (NER) has been extensively studied in the field of Natural Language Processing (NLP), leading to the development of various techniques and methodologies, ranging from rule-based systems to machine learning approaches [1-2]. Traditional approaches to NER include rule-based systems, which rely on predefined patterns and heuristics to identify entities, and statistical models such as Hidden Markov Models (HMMs) and Conditional Random Fields (CRFs), which utilize probabilistic methods for sequence labeling tasks. While these methods have demonstrated effectiveness in generic NER tasks, their performance tends to degrade when confronted with domain-specific texts and scenarios characterized by sparse data. Recent research explored more sophisticated approaches to NER domain adaptation that learn a domain-independent base model which can then be adapted to specific domains. However, most such work focuses on machine learning techniques rather than rule-based systems. [5]. The development of various neural-based approaches helped increase the performance of NER systems on historical materials with F-scores going from 60-70% on average for rule based and traditional ML systems to, for the best neural systems, 80% [6].

The present approaches to incremental NER can be roughly divided into two strategies: semi-supervised learning, which uses both labeled and unlabeled data to enhance model performance, and active learning, which selects the most instructive cases for labeling. Although these strategies have demonstrated promising results in specific situations, they frequently have scaling problems or require the successful use of deep domain knowledge. Improper handling of poorly labeled data can negatively affect the model performance. Semi-supervised baselines perform better than supervised and weakly supervised baselines, suggesting that weak labels, if not managed appropriately, are much more detrimental than fake labels produced by model prediction [7].

The experiment shows that compared with traditional active selection strategies, an uncertainty-based active learning strategy called Lowest Token Probability (LTP), which combines the input and output of a Conditional Random Field (CRF) to select informative instances, has better performance, but lower annotation cost [8]. Rule-based systems rely on handcrafted patterns and linguistic rules to identify named entities, whereas machine learning methods leverage annotated data to learn patterns and generalize them to unseen instances. The findings indicate that a handwritten, rule-based approach outperforms transfer learning and machine learning systems, but these machine learning approaches could be valuable in scenarios where experts are not readily available or an efficient system needs to be developed quickly [9].

Due to demonstrated efficacy for Named Entity Recognition (NER) tasks, conditional random fields (CRFs), recurrent neural networks (RNNs), long short-term memory networks (LSTMs), and transformer-based models received significant attention in the field of machine learning techniques. Deep learning models achieve remarkable performance in Named Entity Recognition (NER) tasks on large datasets by using the hierarchical representations of text data to understand complex patterns and connections [1]. Contextual information in textual data can be effectively captured by using Conditional Random Fields (CRFs), which describe the connections between consecutive tokens in a sequence. Long short-term memory (LSTM) networks in particular are excellent at representing sequential dependencies, and they have shown promising results in NER assignments when used as recurrent neural networks (RNNs). By utilizing substantial pretraining on textual corpora, transformer-based models, such as BERT (Bidirectional Encoder Representations from Transformers), have demonstrated superior performance standards [10].

The Transformer architecture, which relies solely on attention mechanisms rather than recurrent or convolutional layers, achieves state-of-the-art results on machine translation tasks with improved training speed compared to prior approaches. Larger Transformer models provide clear gains over smaller ones, corroborating the benefits of increased model size for such tasks. The Transformer approach holds promise for scaling to longer sequences and being applied to other modalities beyond text [11]. The amount of labeled training data can be drastically reduced when deep learning is combined with active learning. The lightweight architecture for NER with the CNN-CNN-LSTM model consisting of convolutional character and word encoders and a long short-term memory (LSTM) tag decoder achieves nearly state-of-the-art performance on standard datasets for the task while being computationally much more efficient than best performing models with just 25% of the original training data [4, 12].

Despite the effectiveness of existing NER techniques, their performance is often hindered in scenarios where labeled data is limited or expensive to acquire. In many real-world applications, especially in specialized domains or low-resource languages, obtaining sufficient annotated data for training robust NER models is challenging. Limited labeled data can lead to overfitting, poor generalization, and suboptimal performance of NER systems, thereby impeding their practical utility. Data sparsity arises when the distribution of named entities is skewed or when certain entity types are underrepresented in the training data. Domain mismatch occurs when the linguistic characteristics of the training data differ from those of the target domain, leading to a drop in performance when deploying NER models in real-world applications. Annotation cost refers to the expenses associated with manually labeling large volumes of text, which can be prohibitive for resource-constrained organizations or projects, leading to poor performance and low recall rates, particularly for rare or unseen entities.

The performance of NER can be boosted by encompassing a small amount of labeled data with an oversized collection of unlabeled data. For the expectation-maximization algorithm, very few labeled examples are not sufficient to generate parameters for recognition of named entities; therefore, they cannot perform as well as the graph-based label propagation algorithm because of the scarcity of adequate parameters for recognition [13]. The empirical analysis of the unlabeled entity problem in NER showed that models severely suffer from incomplete annotated data. Performance degradation is caused by the reduction in annotated entities and the treatment of unlabeled entities as training negatives, which can be mitigated using pre-trained language models for the poorly annotated corpus and applying negative sampling instead of training negatives [14]. It has been shown that focusing on the fractional corpus containing domain-specialized entities and utilizing a more challenging pre-training strategy in domain-adaptive pre-training is beneficial for NER domain adaptation [15].

Considering the identified limitations, the primary research problem addressed in this study is the development of an incremental approach to domain-specific NER in sparse data environments. While several approaches have been proposed to address NER with limited labeled data, significant gaps remain in the existing methods. A substantial amount of annotated data must be collected for model training in several existing methods that rely on conventional supervised learning paradigms. While active and semi-supervised learning techniques have been investigated to reduce dependence on labeled data, they might not fully utilize the potential of incremental learning or run into scaling problems. Furthermore, existing methods may lack adaptability to evolving linguistic patterns, or may not generalize well across different domains or languages. NER techniques that can effectively generalize a variety of textual inputs, adapt to changing situations, and learn gradually from limited labeled data are in demand. Overcoming these shortcomings will help to advance the state-of-the-art NER state of the art and make it less challenging to design feasible solutions for practical applications with limited labeled data resources.

The goal of the suggested methodology is to guarantee that as new examples are continuously added, the Named Entity Recognition (NER) model may adapt and enhance its representations in response to evolving patterns in data and domain-specific attributes. A progressive NER system strives to improve its efficacy in domain-specific scenarios and mitigate the effects of data scarcity by iterative learning from incoming data streams. To improve research in the field of Natural Language Processing (NLP), this study attempts to advance NER approaches by addressing issues associated with restricted data availability and domain expertise.

The purpose of the study

The overarching goal of this research is to develop an incremental approach to domain-specific Named Entity Recognition (NER) that effectively addresses the challenges posed by sparse data environments. We seek to address the challenge of training robust NER models with sparse annotations by proposing a novel approach that leverages incremental learning techniques. The primary objective is to design a methodology that enables NER systems to iteratively improve their performance over time by adaptively incorporating new labeled or unlabeled data.

Large manually annotated datasets are costly and time-consuming to acquire, particularly for emerging categories of entities or specialized fields. Traditional NER models perform poorly with little labeled data, which results in errors and a decreased capacity to accommodate new entities. Maintaining efficiency while learning from fresh data must be balanced in incremental NER approaches, particularly when working with continuous data streams. Effective integration of new data ought to be possible without the need for expensive retraining on the complete dataset. With minimal data, traditional NER models find it difficult to recognize unknown entity types. With few labeled instances, incremental NER approaches should be flexible enough to learn and include new entity types. By conducting a thorough assessment and comparing it with current NER techniques, this study seeks to create a common paradigm for assessing incremental NER systems and to reveal their advantages, disadvantages, and possible improvements.

Results of the study

Incremental Learning Framework for NER

Large volumes of labeled data are typically required for traditional Named Entity Recognition (NER) algorithms to operate at their best. Nonetheless, acquiring such large-scale labeled datasets may be difficult in many domain-specific applications because of expenses, time constraints, or the fragile nature of the domain. Herein, lies the real value of Incremental Learning (IL) approaches for NER. Adapting and improving over time through the gradual incorporation of new data is the goal of the machine learning paradigm known as incremental learning, which is sometimes referred to as ongoing learning or lifelong learning. Regarding named entity recognition (NER), incremental learning is a viable method to tackle the difficulty of developing strong NER models with a small amount of labeled data that can continuously enhance their performance and update their knowledge base.

The fundamental principle of incremental learning in NER is to use newly annotated or unlabeled data to iteratively update the model parameters while preserving the knowledge gained from prior rounds. Through a series of iterative training cycles, the NER system is able to adjust to shifting data distributions, linguistic patterns, and domain-specific features. Incremental learning can improve a model's accuracy, generalization skills, and resilience to changes in the input data by iteratively improving the model with new knowledge.

The diagram in Figure 1 illustrates the standard pipeline employed for iterative training by utilizing pre-trained models and applying fine-tuning based on subsequent principles:

1. Gradual Model Adjustment: With the help of the framework's dynamic updating mechanism, the NER model can progressively include new data instances without requiring that the entire model be retrained. The

present settings can be adjusted, or additional layers or modules can be trained, to do this. This feature thus enables the model to quickly adapt to changes in the distribution of data and gradually improve performance over time.

2. Domain-specific Data Integration: Mechanisms for integrating domain knowledge into the NER model are incorporated into the framework to handle domain-specific data. To improve the model's capacity to correctly identify entities, this could involve pre-trained embeddings or domain-specific lexicons that capture the special traits and jargon of the target domain.

3. Limited Data Handling Strategies: The framework utilizes multiple ways to optimize the functional value of existing data in settings using sparse data, where annotated data is scarce. Techniques include semi-supervised learning, in which the model incorporates both labeled and unlabeled data to increase performance, and active learning, in which the model chooses informative instances for annotation to augment its training set.

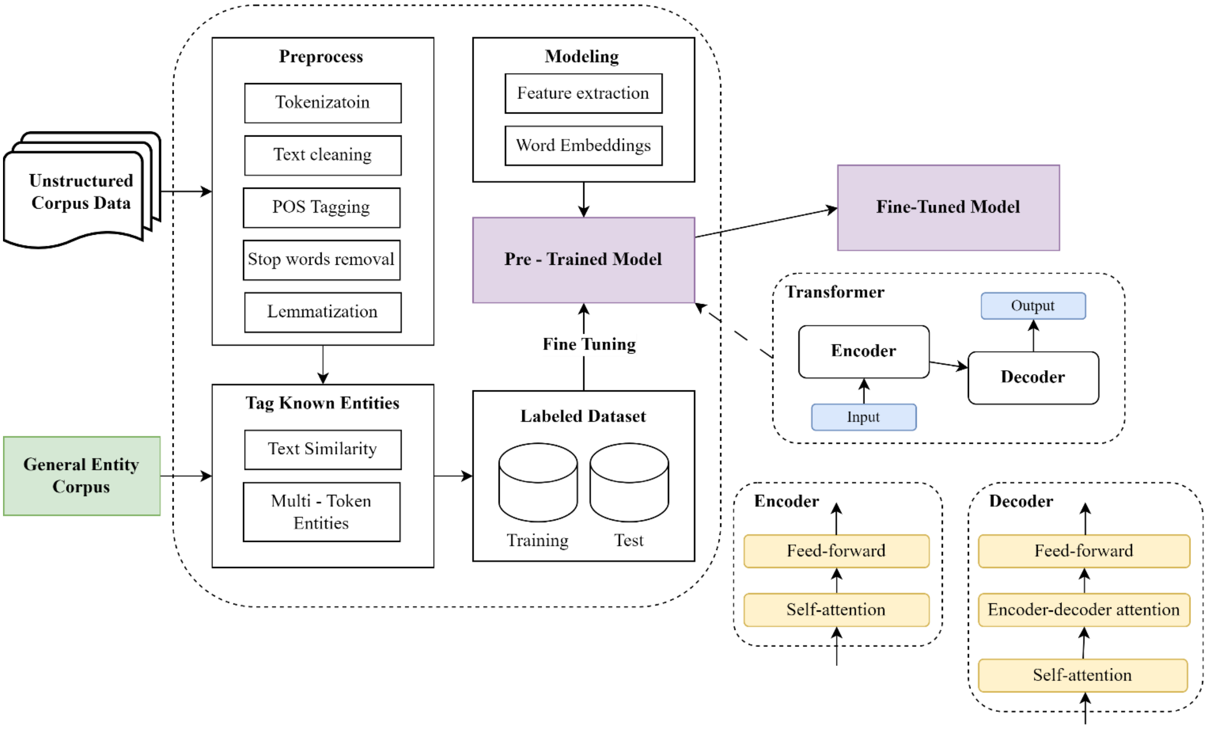

Fig. 1. Adaptive Domain-Specific NER Model Training Pipeline

By introducing domain knowledge into the NER model, the suggested framework is made to handle data that is unique to a certain area. This is accomplished by using lexicons or domain-specific embeddings, which capture the linguistic patterns and semantic similarities unique to the target domain. By including this information, the model performs better overall since it can identify items unique to the domain with more accuracy.

Incremental learning approaches can successfully tackle the difficulties posed by a lack of labeled data by including these tactics into the NER pipeline. This allows for the ongoing refinement and modification of NER models for practical use. These ideas are used to accomplish robust and adaptive named entity recognition in scenarios with limited labeled data resources. We discuss the specific technique and implementation specifics of our suggested incremental NER method in the following sections.

Data Collection and Preparation

We created a unique set of entity labels that correspond to the field of science and the focus of our research. These authorized labels consist of:

- ✓ CHEMICAL: This label is applied to denote references to chemical substances, including both molecular structures and specific elements.
- ✓ BIOLOGICAL: Names of species, biological phenomena, proteins, enzymes, and genes are examples of biological components that fall under this category.
- ✓ METHOD: This label is used to differentiate between references to research approaches and methodologies, including the instruments and processes that are commonly used in the field of scientific inquiry.
- ✓ UNIT: This category designates mathematical symbols, measuring units, and physical constants that are pertinent to the scientific realm.
- ✓ MATERIAL: This class comprises a variety of materials and chemical compositions used or discussed in scientific contexts, such as graphene, silicon, polymers, and nanoparticles.

We employed a pre-trained model known as the Generative Pre-training Transformer to generate sentences specific to certain domains, containing the aforementioned types of labels. By fine-tuning the GPT model on a corpus of scientific literature, we may tell it to produce sentences that are especially likely to incorporate the identified entity categories. This technique makes it easier to create large amounts of synthetic training data that are supplemented with specific terms and concepts relevant to the chosen entities and the subject of science.

Errors and mistakes are common in the sentences that are automatically generated. Enforcing strict manual classification and annotation protocols was necessary to ensure the data's integrity. Human annotators carefully examined the generated phrases, identifying instances of the designated entities inside the article. Following that, a label was assigned to each occurrence of an entity that corresponded to its predefined entity type (CHEMICAL, BIOLOGICAL, METHOD, UNIT, or MATERIAL).

As shown in Figure 2, with 1484 annotated named entities overall, the dataset consists of 1112 records, each of which represents a unique instance of text including mentions of domain-specific entities. These sentences had been carefully selected to guarantee a broad and diverse selection of expressions from the scientific field. The dataset provides extensive coverage of the entity classes under examination, encompassing a wide range of themes and events frequently found in scientific literature. Despite the difficulties presented by limited data, the inclusion enables thorough model training and assessment, guaranteeing strong performance in domain-specific Named Entity Recognition tasks. Image (2) displays the distribution of things by their representative classes.

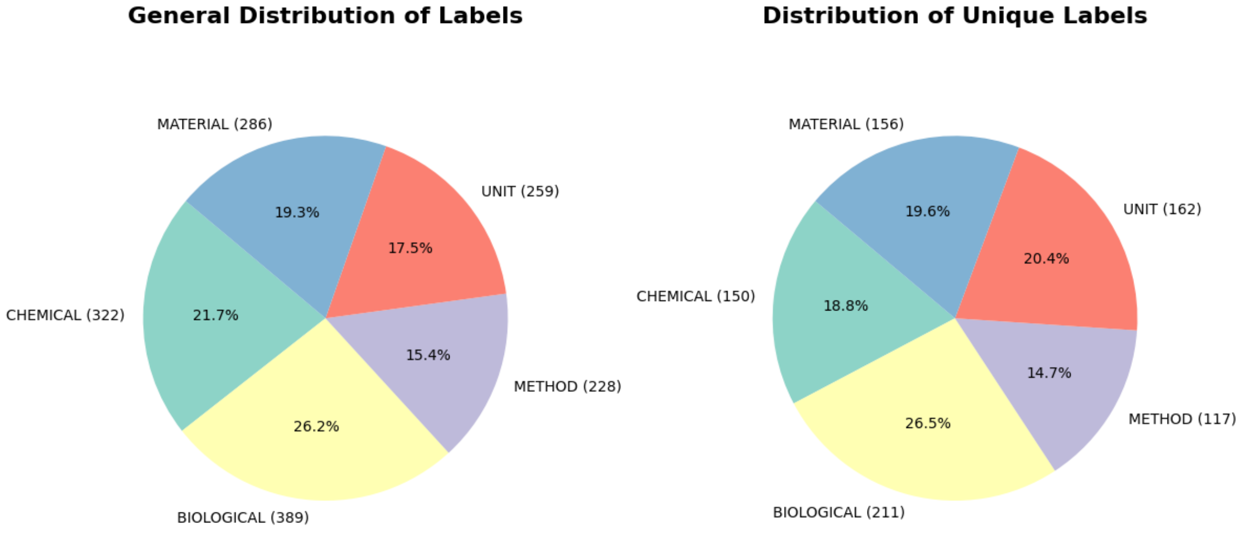

Fig. 2. Distribution of Entity Labels sorted by Total Count and Unique and Unique Occurences

An extensive variety of annotated text documents make up each of the training, validation, and test sets that collectively make up the dataset. The validation set is utilized for model selection and hyperparameter adjustments, while the training set is used to train the NER model. Only data that has never been seen before will be included in the test set to evaluate how well the trained model performs.

Before the NER model can be trained, the dataset is preprocessed to normalize the text and facilitate feature extraction. Text documents are converted to lowercase and are stripped of any extraneous punctuation or whitespace in order to preserve uniformity throughout the entire set of documents. Text documents are subsequently broken down into individual words or sub-word units and tokenized using a tokenization approach appropriate for the language of the dataset. Assigning every token a POS tag can assist in obtaining grammatical information that is vital to NER, such as discriminating between references to potential entities and common phrases with identical spellings. To facilitate model training, named entity labels are represented numerically using encoding. A distinct number identity is assigned so that the model can anticipate the entity labels during training and inference. To efficiently train the model using stochastic gradient descent (SGD) or other optimization algorithms, the training data is further divided into mini-batches.

Transparency and reproducibility in our experimental setup are ensured by providing details about the preprocessing processes performed to the data as well as the dataset used for training and evaluation. The preprocessed dataset is used as the basis for this paper's succeeding parts' training and assessment of the suggested incremental NER approach.

Model Selection and Architecture

Achieving efficient Named Entity Recognition (NER) performance in the presence of sparse data requires careful consideration of model architecture selection. This section explains why we selected spaCy for our domain-specific NER job and illustrates how we fine-tuned a pre-trained spaCy model.

Strong functionalities for Natural Language Processing (NLP) activities are provided by the open-source spaCy library. For several languages, including English, spaCy offers a range of pre-trained statistical and neural network models. Many types of general domain text data are used to train these models, such as "en_core_web_sm", "en_core_web_lg", and the transformer-based model "en_core_web_trf". These models can then be adjusted for particular tasks and domains. Customization of the NER pipeline is possible with spaCy. Using our domain-specific data and the target entity types (CHEMICAL, BIOLOGICAL, METHOD, UNIT, and MATERIAL), we can start with pre-trained models and refine them further. Because of is well-known for efficient performance, spaCy can handle big dataset processing even with constrained computing power.

The following models are used for fine-tuning; each is designed to handle particular subtleties in Named Entity Recognition (NER) tasks:

1. Small toc2vec (ST2V) model: This setup describes the architecture of a spaCy NER model, which feeds a transition-based NER model (ner) with a token vectorization component (tok2vec). Compared to other models, this one may capture less fine-grained information because of its smaller embedding width. The model is designed to be optimized for English text and is meant to be used with a customized NER dataset that contains particular kinds of entities.
2. Large toc2vec (ST2V) model: With word embeddings, this configuration makes use of a pre-trained spaCy model (en_core_web_lg), which may improve the model's recognition performance for entities in the domain. The token vectorization part is improved by using both character-level embeddings and pre-trained word vectors, and by increasing the embedding width.
3. Transformer (TRF) model: Tokenization and embedding are carried out by a pre-trained transformer model (RoBERTa-base), which uses a parser based on transitions for the NER task. It combines data from transformer outputs for NER predictions using a mean pooling layer, suited for training with a GPU, includes gradient accumulation techniques and warmup learning rate scheduling for effective training.

The main distinction between the models is that while both ST2V and LT2V rely on character-level embeddings and offer the option of using pre-trained word vectors, LT2V has a wider embedding width than ST2V. For word embedding, the TRF model uses a transformer that was previously trained. Although the TRF model employs distinct hyperparameters (hidden layer size, usage of uppercase information), all models incorporate the use of a transition-based parser. A distinct batching technique and training setup tailored for GPUs is implemented by the TRF model.

Several essential elements constitute the SpaCy model's architecture, which is used for fine-tuning:

1. Word Embeddings: SpaCy represents words in a continuous vector space by using pre-trained word embeddings. These embeddings help models to build on contextually rich representations by capturing semantic similarities between words.
2. Convolutional Neural Network (CNN) Layers: One or more convolutional layers process the token embeddings in order to extract local characteristics from the input text.
3. Pooling Layers: The convolutional layers' output is combined to gather data from different sections of the input text.
4. Feedforward Layers: One or more feedforward layers receive the pooled features and learn to predict the likelihood that each token would belong to a specific entity type.
5. Named Entity Recognition Layer: Tokens are given entity labels in the last layer, which carries out entity recognition by predicting token probabilities.
6. Fine-tuning Process: Using annotated samples from the target domain, the NER component was adjusted to better fit the pre-trained SpaCy model to domain-specific data. In this procedure, the pre-trained model's knowledge was retained while the model's parameters were updated based on domain-specific data.

We attempted to take advantage of the capabilities of the current model and adapt it to better fit the needs of our NER work inside the designated domain by fine-tuning a pre-trained SpaCy model on our domain-specific data.

Training and Evaluation

The annotated dataset was split into training and evaluation sets to facilitate model training and performance assessment. The majority of the data was allocated for training (~80%) and the remaining portion was reserved for evaluation (~20%). However, in scenarios with sparse data a stratified splitting approach was employed to ensure both training and evaluation sets maintain a similar distribution of entity types (CHEMICAL, BIOLOGICAL, METHOD, UNIT, and MATERIAL) in the original dataset.

A pre-trained SpaCy model was adjusted to the domain-specific data as part of the fine-tuning procedure. After loading the pre-trained SpaCy model, the training set was used to further train the model on the annotated dataset unique to the target domain. The model's parameters, such as the entity recognition rules and neural network layer weights, were changed during the fine-tuning process to maximize performance for the domain-specific NER task. This is a summary of the process of fine-tuning:

1. Freezing Early Layers: Fine-tuning allows for the freezing of the pre-trained model's first layers, which represent general language comprehension. As a result, the model can take advantage of its prior training and avoid overfitting to the particular training set.
2. Training on Labeled Data: The modifiable layers of the selected model are updated through backpropagation using the labeled training data. Within the domain-specific context, the model learns to identify the target entity types (CHEMICAL, BIOLOGICAL, METHOD, UNIT, and MATERIAL).
3. Monitoring Performance: Throughout training, the model's performance is tracked on the held-out evaluation set. In doing so, overfitting is less likely to occur and the number of training iterations through the training data is optimally identified, as well as the model's optimum performance on unknown data.

The performance of the fine-tuned SpaCy model was evaluated using standard NER evaluation metrics. These metrics provide insights into the model's accuracy, precision, recall, and F1-score in identifying named entities within the text:

1. Precision: Precision measures the proportion of correctly predicted entities out of all entities predicted by the model. It is calculated as the ratio of true positives to the sum of true positives and false positives.
2. Recall: Recall measures the proportion of correctly predicted entities out of all true entities in the dataset. It is calculated as the ratio of true positives to the sum of true positives and false negatives.
3. F1-score: The F1-score is the harmonic mean of precision and recall, providing a balanced measure of the model's performance. It is calculated as the harmonic mean of precision and recall, giving equal weight to both metrics.
4. Speed: Throughput of the model prediction speed in terms of words processed per second (WPS).

These measures were computed to evaluate the model's capability to identify each distinct entity type in the domain-specific text data: CHEMICAL, BIOLOGICAL, METHOD, UNIT, and MATERIAL. In addition, measures were employed to assess the overall effectiveness of every type of entities. The fine-tuned spaCy model for domain-specific NER with sparse data can be used to acquire insights into its strengths and flaws by assessing these metrics on the evaluation set. To enhance performance in upcoming iterations, this evaluation enables additional optimization of the model architecture, training procedure, or data gathering tactics.

Discussion of the research results of the on the efficiency of fine-tuned domain-specific NER models

The research on Domain-Specific Named Entity Recognition (NER) with Sparse Data has produced important findings about the performance and effectiveness of domain-specifically optimized NER models. The effectiveness of several refined models, such as the Transformer (TRF) model, the Large toc2vec (ST2V), and the Small toc2vec (ST2V), was evaluated by carefully testing and analyzing their results.

Results in the Table 1 display promising outcomes in terms of entity recognition accuracy and generalization to domain-specific data were found during the evaluation of the refined NER models. Comparing all models to those trained with fewer training data, significant gains in precision, recall, and F1-score were discovered. This demonstrates how effective fine-tuning methods are at improving the model's capacity to recognize named entities in the target domain.

With an F1-score of 0.61, the ST2V model manages to strike a respectable balance between recall and precision. This suggests that the model performs well in terms of recall—capturing a sizable percentage of all relevant named items in the dataset—and precision—identifying relevant named entities. Furthermore, the model exhibits remarkable speed, with the ability to process a substantial quantity of words per second.

While the Small tok2vec model (ST2V: F1-score: 0.61) has a faster prediction speed (1665 vs. 18590 words/second), the Large tok2vec model (LT2V) performs somewhat better in the NER domain (F1-score: 0.62). This shows that, in comparison to ST2V, LT2V's greater embedding width caught more information and produced a better balance between accuracy and speed.

In terms of precision, recall, and F1-score, the TRF model performs better than the ST2V and LT2V models. With the greatest F1-score (0.74), the Transformer-based model (TRF) is the most successful in recognizing true positive entities. However, it processes 152 words per second, which is much fewer than the other models, and it does it at a far slower pace.

Table 1.

General Performance Metrics and Processing Speed Comparison for Different Models after First Iteration of Training

Model	NER P	NER R	NER F	SPEED
ST2V	0.67	0.55	0.61	18590
LT2V	0.69	0.56	0.62	1665
TRF	0.82	0.68	0.74	152

After being trained on more data, as shown in Table 2, all three models show gains in F1-score, precision, and recall. This demonstrates how adding more training data can improve the models' capacity to identify items unique to a given domain. The ST2V model retains a comparable processing speed while exhibiting a minor rise in F1-score (0.02). This implies that it might benefit somewhat from more information. In contrast, the F1-score (0.11) and processing speed (although slightly) show a more notable improvement in the LT2V model. This suggests that it efficiently and successfully utilizes the additional info. Additionally, there is a noticeable increase in speed, as the model can now analyze more words per second.

In the second iteration, the TRF model earns the highest overall F1-score (0.81) with a slight improvement (0.07) over the first iteration. It's interesting to see that it keeps up the same processing pace. This implies that the original data may have taught the TRF model a lot already, and that further data allows for further refining without compromising performance. TRF continuously has the slowest prediction speed even though it provides the best NER performance overall. Accuracy and speed are well-balanced by LT2V, especially with more training data.

Table 2.

General Performance Metrics and Processing Speed Comparison for Different Models after Second Iteration of Training with 20% of Training Data

Model	NER P	NER R	NER F	SPEED
ST2V	0.67	0.59	0.63	18652
LT2V	0.77	0.69	0.73	2527
TRF	0.83	0.78	0.81	156

The Table 3 illustrates the evolution of the three models' performance (ST2V, LT2V, and TRF) with respect to distinct entity labels (CHEMICAL, BIOLOGICAL, METHOD, UNIT, and MATERIAL) following additional data training. For some labels (CHEMICAL, METHOD), the F1-score of the ST2V model indicates small gains; however, for other labels (BIOLOGICAL, UNIT, MATERIAL), there is either stagnation or even a reduction. This indicates that ST2V would not benefit much from more data, maybe because of its model's ability to handle complex learning from a bigger dataset. The F1-score of the LT2V model consistently improves for the majority of labels, especially for UNIT and MATERIAL. This shows how well it uses additional data for different kinds of entities. In the second iteration, the TRF model performs worse for some labels (METHOD) but better for others (CHEMICAL and BIOLOGICAL). Overfitting or restrictions in the training data for particular labels may be the cause of this. It could take more investigation to comprehend this behavior.

Label-wise performance:

✓ CHEMICAL: All models achieved better performance on this label in both iterations, suggesting it might be easier to identify or have more training examples available.

✓ BIOLOGICAL: TRF shows the most significant improvement with more data, while ST2V and LT2V exhibit smaller gains. This suggests the benefit of model architecture (Transformer) for complex labels like biological entities.

✓ METHOD: ST2V and TRF show a decrease in performance, while LT2V improves. This highlights the challenge of this label and the importance of sufficient training data or specific techniques for methods.

✓ UNIT: LT2V shows the most significant improvement, potentially due to better learning patterns for units with more data.

✓ MATERIAL: All models maintain good performance on this label, with some slight variations. More data might not have yielded significant improvement for this category.

For the majority of entity labels, there is a positive association between the quantity of training data and the overall performance of all models (ST2V, LT2V, and TRF). This is demonstrated by the fact that in the second iteration with more data, the F1-scores for LT2V and, to a lesser extent, ST2V, improved. For some models (TRF), labels with more distinct entities (CHEMICAL, BIOLOGICAL) typically exhibit greater improvements with additional data. This implies that the models learn more effectively when there are more variations of an entity. Some labels' (UNIT) inherent qualities, which facilitate learning from more data, may be the cause of performance improvement.

Many difficulties were found when evaluating our domain-specific named entity recognition (NER) models on sparse data; these difficulties may have an impact on the generalizability and performance of the model. To guarantee the NER system's resilience and efficacy, these difficulties and constraints need to be properly handled. The training data's intrinsic sparsity proved a major obstacle. This made it more difficult to determine with certainty how well the models performed, especially when it came to less common entity kinds. To lessen this, strategies like data augmentation or transfer learning from similar fields should be investigated.

There is an imbalance in the number of entities per label (e.g., CHEMICAL vs. METHOD) due to the allocation of training data among labels. Particularly when sparse data is present, some entity labels may be inherently ambiguous. As a result, throughout training, models may prioritize assigning labels with richer presence in the dataset, leading to increase number of false positives. To overcome this, strategies like oversampling or undersampling could be used, as well as loss functions that take class imbalance into consideration.

Table 3.
Results of Models Performance on Different Entity Labels after First and Second (20% training data increase) iterations

Model	Labels														
	CHEMICAL			BIOLOGICAL			METHOD			UNIT			MATERIAL		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
ST2V1	0.75.	0.75	0.72	0.69	0.48	0.56	0.75	0.63	0.68	0.82	0.47	0.60	0.50	0.46	0.48
ST2V2	0.79	0.59	0.64	0.67	0.53	0.59	0.79	0.71	0.75	0.78	0.74	0.76	0.45	0.44	0.44
LT2V1	0.79	0.73	0.76	0.68	0.44	0.53	0.67	0.69	0.68	0.61	0.37	0.46	0.66	0.58	0.62
LT2V2	0.75	0.68	0.71	0.78	0.67	0.72	0.90	0.77	0.83	0.83	0.76	0.79	0.64	0.60	0.62
TRF1	0.94	0.80	0.87	0.87	0.56	0.68	0.86	0.77	0.81	0.67	0.63	0.65	0.71	0.64	0.67
TRF2	0.79	0.89	0.75	0.83	0.68	0.75	0.89	0.83	0.86	0.83	0.82	0.83	0.83	0.70	0.76

In order to tackle these obstacles and constraints, there exist other possible directions for further investigation. Initially, it is worthwhile to look into how well data augmentation methods work to extract more training instances from the available data, especially for less common entity kinds. More research might be done on active learning techniques, in which the model can query for useful data points to enhance its performance iteratively. Finally, improving the interpretability of NER models through the use of methods such as saliency mapping, attention mechanisms, or feature visualization can help with error analysis and debugging as well as improve comprehension of model predictions.

Conclusions

In order to address the difficulties in detecting named entities in specialized fields where annotated material is scarce, this work examined the domain-specific named entity recognition (NER) task in the context of sparse data. The ability of three models (ST2V, LT2V, and TRF) to distinguish between various entity categories (CHEMICAL, BIOLOGICAL, METHOD, UNIT, and MATERIAL) from a small training dataset was assessed.

First, we found that, in comparison to training from scratch, fine-tuning pre-trained models—like spaCy's ST2V, LT2V, and TRF—on domain-specific data can greatly enhance NER performance. The refinement of model performance was aided by the inclusion of more training data in later iterations, which resulted in significant improvements in model precision, recall, and F1-score. Nevertheless, the effect varies with the complexity of the label to be learned and the model design. With more data, ST2V performs somewhat better or even worse for some labels, indicating that its model capacity might not be enough for complicated learning from a bigger dataset. With more data, LT2V shows the most balanced growth in accuracy and processing performance, making efficient use of the new data for entity types.

Additionally, our investigation showed that transformer-based models—specifically, TRF—performed better on a range of evaluation measures, indicating their ability to capture the intricate linguistic patterns and contextual information present in text relevant to a given domain. However, overfitting or restrictions in the training data for particular labels can affect the Transformer-based model (TRF).

These findings demonstrate how crucial model architecture and training data size are to NER performance in sparse data environments. The evaluation method experienced various problems and restrictions, such as data sparsity, label imbalance, and model complexity, despite the encouraging results. These difficulties highlight the need for additional study to create NER systems that are more reliable and efficient. Prospective research avenues to enhance model performance with minimal data comprise investigating data augmentation approaches, active learning tactics, and domain adaption methods. Furthermore, researching explainability strategies can help direct future development and offer insights into the behavior of the model.

All things considered, this work advances the area of domain-specific NER by proving that deep learning models may be successful in information extraction tasks even in the presence of sparse labeled data. It highlights how crucial it is to take into account elements like training data characteristics and model architecture when developing NER systems for particular domains.

References

1. Li J, Sun A, Han J, Li C. A survey on deep learning for named entity recognition. IEEE transactions on knowledge and data engineering. 2020 Mar 17;34(1):50-70.
2. Lample G, Ballesteros M, Subramanian S, Kawakami K, Dyer C. Neural architectures for named entity recognition. arXiv preprint arXiv:1603.01360. 2016 Mar 4.
3. Wang R, Yu T, Zhao H, Kim S, Mitra S, Zhang R, Henao R. Few-shot class-incremental learning for named entity recognition. InProceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) 2022 May (pp. 571-582).
4. Chiu JP, Nichols E. Named entity recognition with bidirectional LSTM-CNNs. Transactions of the association for computational linguistics. 2016 Jul 1;4:357-70.

5. Chiticariu L, Krishnamurthy R, Li Y, Reiss F, Vaithyanathan S. Domain adaptation of rule-based annotators for named-entity recognition tasks. InProceedings of the 2010 conference on empirical methods in natural language processing 2010 Oct (pp. 1002-1012).

6. Ehrmann M, Hamdi A, Pontes EL, Romanello M, Doucet A. Named entity recognition and classification in historical documents: A survey. ACM Computing Surveys. 2023 Sep 14;56(2):1-47.

7. Jiang H, Zhang D, Cao T, Yin B, Zhao T. Named entity recognition with small strongly labeled and large weakly labeled data. arXiv preprint arXiv:2106.08977. 2021 Jun 16.

8. Liu M, Tu Z, Zhang T, Su T, Xu X, Wang Z. LTP: a new active learning strategy for CRF-based named entity recognition. Neural Processing Letters. 2022 Jun;54(3):2433-54.

9. Gorinski PJ, Wu H, Grover C, Tobin R, Talbot C, Whalley H, Sudlow C, Whiteley W, Alex B. Named entity recognition for electronic health records: a comparison of rule-based and machine learning approaches. arXiv preprint arXiv:1903.03985. 2019 Mar 10.

10. Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. 2018 Oct 11.

11. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. Advances in neural information processing systems. 2017;30.

12. Shen Y, Yun H, Lipton ZC, Kronrod Y, Anandkumar A. Deep active learning for named entity recognition. arXiv preprint arXiv:1707.05928. 2017 Jul 19.

13. Sintayehu H, Lehal GS. Named entity recognition: a semi-supervised learning approach. International Journal of Information Technology. 2021 Aug;13:1659-65.

14. Li Y, Liu L, Shi S. Empirical analysis of unlabeled entity problem in named entity recognition. arXiv preprint arXiv:2012.05426. 2020 Dec 10.

15. Liu Z, Xu Y, Yu T, Dai W, Ji Z, Cahyawijaya S, Madotto A, Fung P. Crossner: Evaluating cross-domain named entity recognition. InProceedings of the AAAI Conference on Artificial Intelligence 2021 May 18 (Vol. 35, No. 15, pp. 13452-13460).