applied sciences

MDPI

Review

Biomedical Flat and Nested Named Entity Recognition: Methods, Challenges, and Advances

Yesol Park [1], Gyujin Son [2] and Mina Rho [1,2,3,*]

[1] Department of Computer Science, Hanyang University, Seoul 04763, Republic of Korea; yesolpark@hanyang.ac.kr
[2] Department of Artificial Intelligence, Hanyang University, Seoul 04763, Republic of Korea; cnbe5494@hanyang.ac.kr
[3] Department of Biomedical Informatics, Hanyang University, Seoul 04763, Republic of Korea
* Correspondence: minarho@hanyang.ac.kr

Abstract: Biomedical named entity recognition (BioNER) aims to identify and classify biomedical entities (i.e., diseases, chemicals, and genes) from text into predefined classes. This process serves as an important initial step in extracting biomedical information from textual sources. Considering the structure of the entities it addresses, BioNER tasks are divided into two categories: flat NER, where entities are non-overlapping, and nested NER, which identifies entities embedded within another. While early studies primarily addressed flat NER, recent advances in neural models have enabled more sophisticated approaches to nested NER, gaining increasing relevance in the biomedical field, where entity relationships are often complex and hierarchically structured. This review, thus, focuses on the latest progress in large-scale pre-trained language model-based approaches, which have shown the significantly improved performance of NER. The state-of-the-art flat NER models have achieved average F1-scores of 84% on BC2GM, 89% on NCBI Disease, and 92% on BC4CHEM, while nested NER models have reached 80% on the GENIA dataset, indicating room for enhancement. In addition, we discuss persistent challenges, including inconsistencies of named entities annotated across different corpora and the limited availability of named entities of various entity types, particularly for multi-type or nested NER. To the best of our knowledge, this paper is the first comprehensive review of pre-trained language model-based flat and nested BioNER models, providing a categorical analysis among the methods and related challenges for future research and development in the field.

Keywords: named entity recognition; biomedical named entity recognition; flat named entity recognition; nested named entity recognition; flat and nested named entity recognition; natural language processing

Citation: Park, Y.; Son, G.; Rho, M. Biomedical Flat and Nested Named Entity Recognition: Methods, Challenges, and Advances. *Appl. Sci.* **2024**, *14*, 9302. https://doi.org/10.3390/app14209302

Academic Editors: Lykourgos Magafas and Rui Araújo

Received: 20 August 2024
Revised: 3 October 2024
Accepted: 5 October 2024
Published: 12 October 2024

1. Introduction

Biomedical articles serve as invaluable resources of knowledge and insights in medicine and life sciences. However, the manual extraction of comprehensive information from the rapidly amassing literature often comes with prohibitive costs. Consequently, there has been an increase in interest in developing automated and intelligent systems for extracting information from biomedical articles, including tasks such as text summarization [1,2], relation extraction [1,3], and knowledge graph construction [4]. Named entity recognition (NER) plays a crucial role in biomedical information extraction because it serves as the foundation for various downstream tasks. For instance, in relation extraction, the entities recognized through NER are used to identify and extract meaningful relationships between them [5]. Similarly, in knowledge graph construction, NER results form the basis for the graph, where the recognized entities become the nodes and the extracted relationships form the edges, allowing for a structured representation of knowledge [6].

The NER task aims to identify and classify terms into predefined categories, such as diseases, genes, or chemicals [7,8]. The NER is categorized into flat and nested NERs

based on the structure of the target entities. The flat NER task recognizes entities that are a singular span of consecutive words without overlap or nesting with others. On the other hand, the nested NER recognizes all named entities (NEs) that either overlap or nest with others, considering the hierarchical properties of entities. The properties provide a more comprehensive understanding of themselves and their interrelationships.

Biomedical NER (BioNER) poses significant challenges for several reasons [9–11]. One of the challenges is that a single term may encompass multiple types of entities depending on the context, introducing potential ambiguity. Moreover, biomedical terminologies are highly specialized and often differ from general ones, leading to suboptimal results when applied to embedding spaces designed for general corpora. Additionally, the continuous creation of new biomedical terms presents a challenge for BioNER to keep up with the latest vocabulary. In addition, biomedical NEs can cascade, introducing confusion about entity boundaries. For example, the DNA region "HIV-2 enhancer" contains the virus "HIV-2". Furthermore, biomedical entities often have descriptive names conveying important information about their functions, properties, or diagnostic criteria. Examples include "clear-cell metastatic renal cell carcinoma" and "autosomal dominant nocturnal frontal lobe epilepsy". These complications have highlighted the importance of nested NER to obtain more correct NEs.

Early BioNER systems relied on predefined dictionaries or rules to identify NEs [12,13], having limitations in recognizing NEs beyond the coverage of their predefined repertoires. Traditional machine learning (ML) approaches, on the other hand, learn patterns in features associated with NEs, such as capitalization, prefixes, suffixes, and part-of-speech tags, allowing them to identify entities with homologous meaning [14–16]. While these approaches can predict previously unseen NEs, they have the drawback of relying on manually designed features, which limits scalability and hinders the discovery of innovative patterns within the data.

In contrast, deep learning (DL) approaches automatically extract features using learning algorithms [17–19]. This allows them to adapt to diverse and heterogeneous entities and generalize better to unseen data. In particular, the advent of large-scale pre-trained language models (PLMs) such as BERT [20], XLNet [21], and GPT [22] has driven significant advancements in natural language processing (NLP) in recent years. Transfer learning models with PLMs have achieved state-of-the-art (SOTA) performance on various NLP tasks [23,24].

These PLM-based models have several limitations. First, PLMs such as BERT are constrained by a maximum input length, making it challenging to capture context in longer texts. Second, large-scale language models such as BERT, which has about 110 million parameters, and XLNet, which has about 117 million parameters, require substantial computational resources, making deployment difficult in resource-limited environments. To address these issues, models such as Longformer [25] extend the input length, while lightweight models such as DistilBERT [26] and BioMobileBERT [27] offer reduced computational overhead without sacrificing performance.

Many studies have adopted PLM in recent years. As shown in Supplementary Table S1, these methods have achieved significant improvements compared to existing machine learning and deep learning models, with performance increases of 3–7% across datasets such as BC2GM, NCBI disease, BC5CDR disease, and BC4CHMED. In line with this trend, this study provides a comprehensive review of the recent advancements in both flat and nested BioNER, with a focus on PLM-based approaches. This presents relevant resources and categorizes models from a problem-solving perspective. The review encompasses a wide range of approaches, from conventional methods to the latest innovations in the field. In particular, the flat NER section covers sequence labeling, the machine reading comprehension (MRC)-based approach, and multi-task learning for the multi-type NER model. Additionally, the nested NER section presents layer-based, span labeling approaches, sequence-to-set, and dependency parsing-based approaches. We also provide an analysis covering various aspects useful for future research, including the similarities

between different entity types and the contribution of encoding and decoding layers in PLM-based NER models. Finally, we discuss persistent issues, such as inconsistencies in entity type definitions across datasets and the limited availability of annotated corpora, especially for multi-type or nested NER tasks. To the best of our knowledge, this paper serves as the first comprehensive review of PLM-based flat and nested BioNER models.

2. Background

2.1. Overview of Flat NER

The flat NER identifies NEs defined by a single span of consecutive words without overlapping and nesting with other entities [15,28]. For example, as shown in Figure 1a,b, the flat NER model recognizes four entities in the given sentence: "Tetracycline resistance" as a gene; "*Clostridium perfringens*" as a species; "cattle" as a species; and "malignant edema" as a disease. A primary approach used in flat NER is sequence labeling, which generates a sequence of labels for an input sentence, creating one-to-one input-output pairs. The labeling scheme for this approach includes IO, BIO, BIOES, and BILOU, according to the combination of labels [29–31]. For example, the BIO scheme consists of three labels: B, I, and O. The label B indicates the beginning of the NE (i.e., the first word), and the label I indicates all other words in the span of the NE. The label O is used for words that are not part of any NE. For instance, when annotating a gene "Tetracycline resistance genes" using the BIO scheme, "Tetracycline" would be labeled as "B-gene", and both "resistance" and "genes" would be labeled as "I-gene". Additionally, the BIOES scheme extends the BIO scheme by including the label S for a single-word NE and the label E for the last word of the NE.

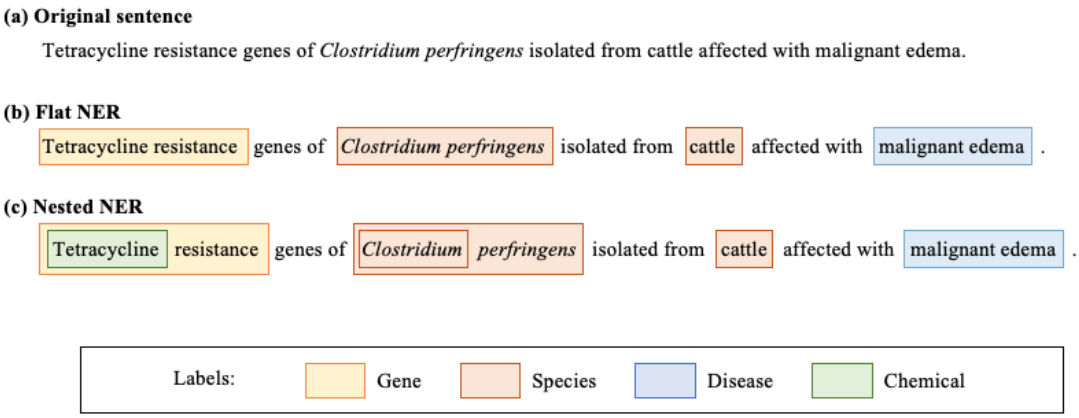

Figure 1. Example of flat and nested NER results. (**a**) Original sentence. (**b**) Result from flat NER. (**c**) Result from nested NER. The yellow rectangle indicates a gene; the orange rectangle indicates a species; the blue rectangle indicates a disease; and the green rectangle indicates a chemical as labels.

2.2. Overview of Nested NER

The nested NER aims to recognize entities at all hierarchical levels, including those embedded within other entities [15,32]. For instance, as shown in Figure 1c, the nested NER model additionally identifies "Tetracycline" as a chemical and "*Clostridium*" as a species alongside the entities detected by the flat NER model (Figure 1b). While the sequence labeling approach, which is typically used in the flat NER, can handle nested structures, it necessitates a more complex labeling scheme [33]. For this reason, nested NER is commonly approached using a tuple scheme. Formally, given an input sentence $X = \{x_1, x_2, \ldots, x_n\}$, nested NER outputs all entities in the input sentence as $Y = \left\{\left(s_1^{head}, s_1^{tail}, t_1\right), \left(s_2^{head}, s_2^{tail}, t_2\right), \ldots, \left(s_m^{head}, s_m^{tail}, t_m\right)\right\}$. Here, n is the number of words in the sentence, and m is the number of NEs. In the output, s_i^{head} and s_i^{tail} represent indices of the head and tail words of the i-th NE, respectively, and t_i represents the corresponding type of NE. For example, when annotating the nested NEs in Figure 1c, the

tuples would be represented as follows: {(1, 1, chemical), (1, 3, gene), (5, 5, species), (5, 6, species), (9, 9, species), (12, 13, disease)}.

2.3. Evaluation Metrics for NER

Three metrics are commonly used to evaluate performance in NER: precision, recall, and F1-score [28,34]. Precision is the proportion of correctly identified entities among the NEs predicted by the model, which is calculated as

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{1}$$

and recall is the proportion of NEs correctly predicted by the model among the actual NEs, which is calculated as

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

Here, TP refers to the number of NEs that are correctly predicted by the model. FP refers to the number of NEs incorrectly predicted. FN refers to the number of actual NEs that the model fails to predict. In the NER, a predicted NE is considered correct if it matches the actual NE in both its boundaries and type annotations.

The F1-score is a comprehensive metric that balances precision and recall. It is calculated as:

$$\text{F1-score} = \frac{2}{\text{precision}^{-1} + \text{recall}^{-1}} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \tag{3}$$

The highest achievable value for the F1-score is 1.0, with higher F1-scores indicating better overall efficiency of the model.

2.4. Text Representations

Text representation is a primary component of NLP models. Well-trained text representation positively impacts NLP model performance. Low-dimensional word representations, commonly called word embeddings [35,36], have been demonstrated to be effective across a variety of NLP tasks. However, word-level representations suffer from a chronic out-of-vocabulary (OOV) problem. Alternative approaches, such as fastText [37] and character embedding [38], have been developed to address this issue. FastText learns representations for n-gram characters and represents a word by aggregating the vectors of its constituent n-grams. The character embedding model learns to represent each character in a word as a vector and then combines these character vectors to obtain a representation for the entire word. Both fastText and character embedding mitigate the OOV problem by capturing the morphological aspects of words.

These representations are typically pre-computed and remain static during the training and inference of the NLP model. As a result, these static representations treat different homonyms as the same vector, ignoring their contextual differences. In contrast, contextual language models, such as BERT [20], ELMo [39], and GPT [22], dynamically capture word semantics in a context-dependent manner using DL. Specifically, they assign different vectors to the same word depending on the context, thereby solving the problem of homonyms. Moreover, these contextual language models can be transfer-learned for downstream tasks. The transfer learning models based on the contextual language model have achieved SOTA performance in a variety of domains. In the biomedical field, biomedical PLMs such as ClinicalBERT [40], PubmedBERT [41], and Bioformer [42] are widely utilized.

2.5. Biomedical NER Datasets

Table 1 presents benchmark datasets comprising 26 and 2 for flat and nested NE datasets. The table also provides whether NE normalization labels are included in each dataset. The normalization task maps NEs to unique, standardized identifiers to ensure that similar or identical entities are recognized consistently. This task is an essential post-process for NER as it enhances the clarity and usability of NEs.

Table 1. List of datasets for flat and nested named entity recognition.

Dataset	Norm [+]	URL	Ref.
Flat NER			
Multi-Type			
BioRED	✓	https://ftp.ncbi.nlm.nih.gov/pub/lu/BioRED/	[43]
MedMention	✓	https://github.com/chanzuckerberg/MedMentions	[44]
CRAFT	✓	http://bionlp-corpora.sourceforge.net/CRAFT/	[45]
JNLPBA		https://github.com/openbiocorpora/jnlpba *	[46]
CellFinder		https://github.com/openbiocorpora/cellfinder *	[47]
miRNA		https://www.scai.fraunhofer.de/mirna-corpora.html	[48]
Nagel		https://sourceforge.net/projects/bionlp-corpora/files/ProteinResidue/	[49]
GREC		http://www.nactem.ac.uk/GREC/	[50]
BC5CDR	✓	https://github.com/openbiocorpora/biocreative-v-cdr *	[51]
DrugProt		https://zenodo.org/records/5119892	[52]
tmVar 3.0	✓	https://github.com/ncbi/tmVar3?tab=readme-ov-file	[53]
Gene/Protein			
BC2GM	✓	https://github.com/openbiocorpora/biocreative-ii-gm *	[54]
DECA	✓	http://www.nactem.ac.uk/deca/	[55]
GETM		https://getm-project.sourceforge.net/	[56]
LocText	✓	https://github.com/Rostlab/LocText	[57]
FSU-PRGE		https://julielab.de/Resources/FSU_PRGE.html	[58]
NLM-Gene	✓	https://ftp.ncbi.nlm.nih.gov/pub/lu/NLMGene/	[59]
Chemical			
SCAI Chem	✓	https://www.scai.fraunhofer.de/chem-corpora.html	[60]
DDI		https://github.com/isegura/DDICorpus	[61]
BC4CHEMD	✓	https://github.com/bionlp-hzau/BioNLP-Corpus *	[62]
NLM-Chem	✓	https://ftp.ncbi.nlm.nih.gov/pub/lu/NLMChem/	[63]
Disease			
SCAI disease		https://www.scai.fraunhofer.de/disease-ae-corpus.html	[64]
NCBI disease	✓	http://www.ncbi.nlm.nih.gov/CBBresearch/Dogan/DISEASE/	[65]
Species			
LINNAEUS	✓	http://linnaeus.sourceforge.net/	[66]
Species-1000	✓	https://jensenlab.org/resources/s1000/	[67]
Variant			
SNP corpus	✓	https://www.scai.fraunhofer.de/snp-normalization-corpus.html	[68]
Nested NER			
GENIA		https://github.com/openbiocorpora/genia-term *	[69]
Bacteria biotopes	✓	https://sites.google.com/view/bb-2019/dataset?authuser=0	[70]

[+] Norm indicates NE normalization. * These URLs are not from official sources; they have been shared by independent users. URLs accessed on 14 December 2023.

Among datasets, 11 flat and 2 nested NE datasets involve multiple entity types. The multi-type datasets embody realistic complexity and diversity, thereby enhancing the capabilities of NER models. Moreover, GENIA and bacteria biotopes (BB19) provide nested entities in approximately 22% and 11% of sentences in each set, respectively. Such nested NE datasets allow NER models to learn hierarchical information representing the complicated relationships among biomedical entities.

3. Deep Learning Methods for Flat NER

This section categorizes flat BioNER models into three main approaches: sequence labeling, MRC, and multi-task learning. Sequence labeling models are favored for their simplicity and efficiency, making them straightforward to implement and fast. MRC models enhance performance by integrating external knowledge and framing NER tasks as question-answering problems. Multi-task learning models predict multiple types of entities simultaneously by leveraging shared knowledge across tasks. Table 2 reports the performance metrics of each model on biomedical corpora, with the metrics sourced from

the respective method papers. While the recent models achieve high F1-scores (90–93%) for NCBI disease, BC5CDR chemical, BC4CHEMD, and LINNAEUS, they have relatively weak performance on BC2GM (84.4 ± 1.2%) and BC5CDR disease (87.7 ± 0.5%). The gene entities in BC2GM include complex variations such as abbreviations, numbers, and special characters (e.g., pepX, AB004534, and SNAP-23), adding challenges for models. Similarly, BC5CDR disease shows weaker performance than NCBI disease, likely due to differences in disease scope and low overlap in linguistic content between training and test data [71].

Hereafter, models are introduced based on a two-step structure consisting of encoding and decoding layers, as illustrated in Figure 2. The encoding layer includes a text vectorization layer and a context layer. The text vectorization layer converts words or tokens into numerical vectors. The context layer enhances the ability of the model to understand the broader linguistic context by employing neural networks, such as long short-term memory (LSTM) or transformer-based language models. Lastly, the decoding layer aims to predict NEs based on context information from the encoding layer. The configuration of this layer varies depending on the specific approach to handling NEs.

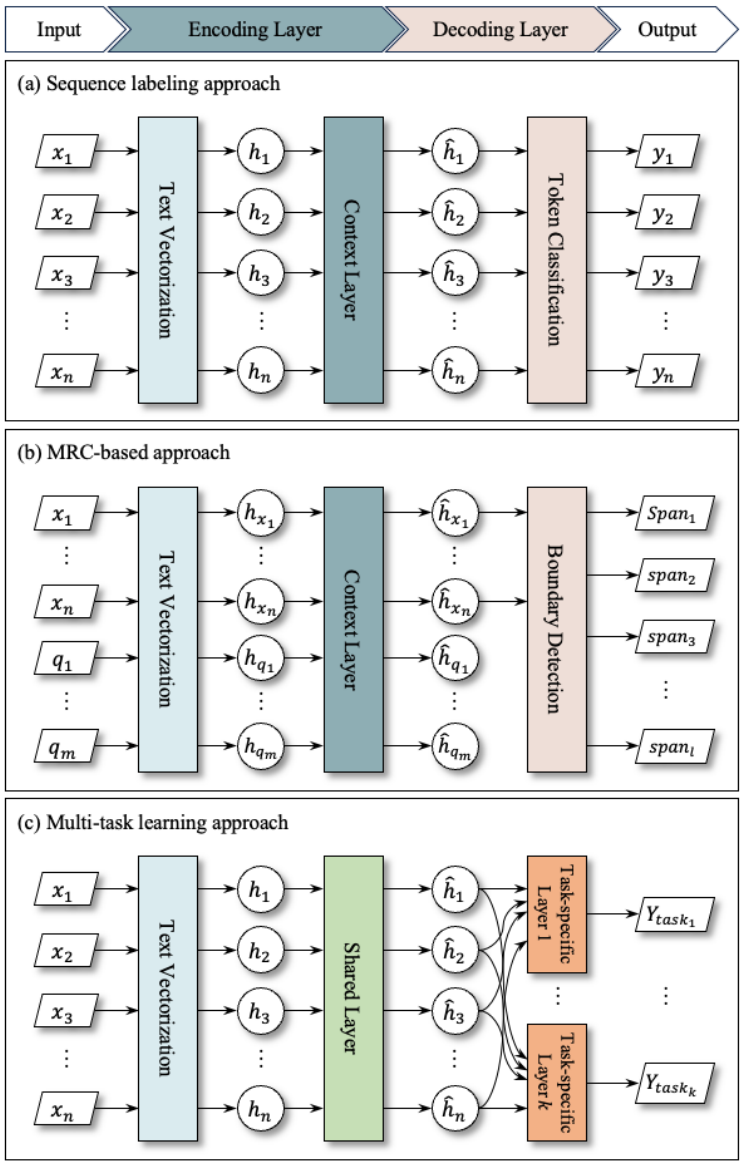

Figure 2. Overview of three flat NER approaches. For each approach, the diagram shows how an input sequence $\{x_1, \ldots, x_n\}$, where x_i indicates the i-th word in an input sentence, is transformed

through encoding and decoding layers to predict flat NEs. The encoding layer consists of a text vectorization, which converts words or tokens into numerical vectors, and a context layer that uses neural networks such as LSTM or transformer to capture broader linguistic context. The decoding layer then predicts NEs based on the context from the encoding layer. (**a**) Sequence labeling approach. h_i and $\hat{h}_i$ refer to the initial and latent embeddings for x_i, respectively. y_i is a label assigned to x_i. (**b**) MRC-based approach. This approach includes an additional input in the form of a query: q_i represents the i-th word in the query. h_{q_i} indicates initial embedding for i-th word in the query, while $\hat{h}_{q_i}$ represents its latent embedding. $Span_i$ refers to i-th entity generated through boundary detection. (**c**) Multi-task learning approach. Y_{task_i} is the output for the i-th task.

Table 2. A comprehensive list of methods for flat BioNER.

Approach Type	Model	Performance (F1-Score)						Code	Ref.
		Gene/Protein	Disease		Chemical		Species		
		BC2GM	NCBI Disease	BC5CDR Disease	BC5CDR Chemical	BC4CHEMD	LINNAEUS		
Sequence labeling	BioBERT	84.72	89.71	87.15	93.47	92.36	88.24	Yes	[23]
	Naseem et al.	86.05	91.23	88.34	94.24	92.28	-	No	[24]
MRC	BioBERT-MRC	85.48	90.04	87.83	94.19	92.92	-	Yes	[72]
Multi-task learning	MT-BioNER	83.01	88.10	-	89.50	-	-	No	[73]
	MTL-LS	82.92	89.25	87.28	93.83	92.42	86.37	No	[74]
	BERT-CNN	83.47	89.72	-	-	92.39	92.63	Yes	[75]
	AIONER	-	89.59	87.89	92.84	-	90.63	Yes	[76]
	TaughtNet	84.84	89.20	-	93.95	-	-	Yes	[77]

3.1. Sequence Labeling

The sequence labeling approach is the most commonly used method in flat NER (Figure 2a). In this approach, a sentence is split into individual tokens, which are taken as input and processed through an encoding layer. The encoding layer enables the model to interpret each token within the context of the entire sentence. Following this, a token classification layer is applied to assign a label to each token based on the contextual information learned from the previous layer.

Lee et al. [23] utilize a transformer model, trained on the biomedical text and named BioBERT, in the encoding layer. They also employ a feed-forward neural network (FFNN) and *softmax* function as the token classification layer. BioBERT is initialized with the weights of BERT [20] and retrained on the biomedical domain corpora, including PubMed abstracts and PMC full-text articles. It outperforms BERT on most BioNER datasets, proving the effect of the domain-specific representation model.

A model developed by Naseem et al. [24] employs a fused representation by concatenating four representations: word-level representation [78], character-level representation [31], BioELMo [79], and BioBERT [23]. They apply an attention-based bi-directional LSTM (Bi-LSTM) as the context layer and a conditional random field (CRF) as the token classification layer. The fused representation outperforms the one using BioBERT alone. However, the complex representation requires substantial computational resources and memory, which may hinder its efficiency in real-time processing and resource-constrained environments.

3.2. Machine Reading Comprehension-Based Approach

MRC is an automated system designed to comprehend a passage and respond to questions. This approach has been applied in various NLP tasks. Figure 2b illustrates the process of an MRC-based approach designed for the NER task. This approach takes a sentence along with a query as input. The queries include information about the NEs,

such as entity type, definitions, and examples. Through the encoding layer, the sentence's embeddings incorporate the information from the query. Finally, in the boundary detection layer, the model identifies a set of spans corresponding to NEs.

BioBERT-MRC, as proposed by Sun et al., uses the MRC approach to address the flat NER problem [72]. Their input is defined as "[CLS] *Sentence* [SEP] *Query* [SEP]", where [CLS] and [SEP] are special tokens used to indicate the start of input and the separation between sentences, respectively. The query involves a request to detect NEs along with examples of such entities. The BioBERT is used for the encoding layer. In the boundary detection layer, two FFNNs are employed to predict the start and end indices of NEs. These two FFNNs operate sequentially, first predicting the start indices and then predicting the end indices with information on the start indices. Finally, the model generates final outputs by matching the nearest start and end indices. This model is limited in that it can only predict a single entity type per inference, even when using multi-type datasets. Moreover, the presence of queries may aggravate the inherent length limitations in PLM-based models. Nevertheless, the approach of incorporating external knowledge into the model through queries is interesting. The MRC system can also be utilized to solve nested or multi-type NEs.

3.3. Multi-Task Learning

A straightforward approach to enabling an NER model to predict diverse types of entities is to train it on a dataset that encompasses all those types of entities. However, multi-type datasets composed of the desired entity types are rare in the biomedical domain. To overcome such limitations, several studies have explored methods to train a model on a combination of multiple datasets. A representative approach is multi-task learning, which successfully achieves this goal by training a model to perform multiple tasks simultaneously.

Figure 2c illustrates the general multi-task learning model consisting of a single shared layer and several task-specific layers. The shared layer is used for all tasks and learns common knowledge across them. Subsequently, the latent embedding, processed through the share layer, is passed onto the task-specific layers, which are further refined and specialized for each task, ultimately producing task-specific outputs $\{Y_{task_1}, \ldots, Y_{task_n}\}$ tailored to each task. Given m tasks, for $i \in \{1, \ldots, m\}$, the total loss function L_{total} of the multi-task model can be expressed as [73]

$$L_{total} = \sum_{i=1}^{m} \lambda_i L_i(\theta_{shared}, \theta_i). \quad (4)$$

Here, the hyperparameter λ_i controls the contribution of the i-th task. The parameters in the shared layer and the i-th task-specific layer are represented by θ_{shared} and θ_i, respectively.

Khan et al. introduced MT-BioNER, which utilizes an encoding layer (i.e., BioBERT) as a shared layer and a decoding layer as a task-specific layer [73]. The task-specific layers are assigned to each dataset. They present two versions of the model: one trained on three datasets (BC2GM, BC5CDR chemical, and NCBI disease) and the other trained on four datasets, including JNLPBA. The latter model, with additional training data, performs worse overall than the former. This result suggests that merely augmenting the training data does not necessarily enhance performance. Similarly, Chai et al. proposed MTL-LS, which consists of a shared layer and multiple task-specific layers [74]. They apply hierarchical sharing on the encoding layer to improve the stability of the multi-task model. Specifically, they divide the encoding layer (i.e., XLNet [21]) into the lower and upper parts. The lower part serves as a shared layer, while the upper part, along with the decoding layer, is allocated per task. Note that MTL-LS was trained on multiple datasets by categorizing them based on entity types, with each category treated as a separate task.

Meanwhile, Banerjee et al. introduced a multi-task learning model using the MRC system, named BERT-CNN [75]. It is noted that this model receives information about tasks through queries. Therefore, unlike typical multi-task learning models, it does not

require task-specific layers and has the structure of the MRC-based model (Figure 2b). The model utilizes a combination of BERT and a convolutional neural network (CNN) as an encoding layer, which captures global and local information, respectively. Subsequently, the model utilizes a sequence labeling layer to determine the boundaries of certain types of NEs. The labeling scheme includes "B", "I", and "O" without specifying the type of entity. Similarly, all-in-one NER (AIONER), as developed by Luo and Wei, receives information about tasks through special tags instead of queries [76]. The special tags, unlike the queries used in BERT-CNN, do not incorporate detailed external information such as definitions or examples. The input for AIONER is structured as "<task> Sentence </task>", where the tag pair "<task></task>" specifies a particular entity type, such as "<disease></disease>". AIONER also employs a sequence labeling layer to predict the boundaries of NEs. For a particular task, its labeling scheme consists of "B-task", "I-task", and "O-task" labels (e.g., B-disease, I-disease, and O-disease). Unlike traditional BIO schemes, "O" labels are customized for each task to enhance flexibility and alleviate task conflicts. This model also supports the "<all></all>" tag pair, which collectively represents all entity types. However, this tag requires a dataset that includes all entity types.

Moscato et al. [77] proposed TaughtNet, a novel method of multi-task learning that applies knowledge distillation. Knowledge distillation involves training a compact student model to emulate a larger teacher model. They first train single-type NER models to serve as teachers on distinct datasets. In this study, three teacher models are trained separately on the NCBI disease, BC5DER chemical, and BC2GM. Subsequently, they integrate the probability distributions from teacher models into a unified distribution that the student model can emulate. In the training phase, the student model aims to minimize the discrepancy between its own distribution and the unified distribution. Simultaneously, it also minimizes the discrepancy with ground truth, similar to existing sequence labeling models. The knowledge distillation approach enables multi-task learning on a model that is more compact than the single-teacher model, highlighting its usefulness under stringent computational and memory constraints.

The models under discussion differ in their approach to integrating task-specific information and in their architectural structure. MT-BioNER and MTL-LS employ a shared encoding layer with task-specific decoding layers for each dataset of entity types. This results in a more complex structure due to the necessity for discrete decoders. In contrast, BERT-CNN, AIONER, and TaughtNet eliminate the necessity for task-specific layers. BERT-CNN and AIONER acquire task-specific information through queries and tags, respectively, allowing them to process multiple tasks within a unified structure. Similarly, TaughtNet also bypasses the need for task-specific layers by training a compact student model to emulate several teacher models through knowledge distillation.

A multi-task learning approach is an effective approach for addressing the lack of multi-type datasets in the BioNER domain. Additionally, it reduces memory requirements by allowing multiple tasks to be processed with a single model. However, not all tasks positively influence each other, and conflicts between tasks may lead to performance degradation. Additionally, most multi-task models produce separate outputs for each entity type, necessitating a post-processing step to integrate these outputs for practical applications.

4. Deep Learning Methods for Nested NER

Nested NER is a more complex problem than flat NER because it recognizes entities that nest or overlap with other entities. Recent advances in DL have yielded promising results in handling nested NER, leading to an increased interest in it. This section provides an overview of various nested NER models, broadly classified into layer-based, span labeling, and other approaches, as outlined in Table 3. The layer-based models solve the problem of nested entities by using multiple layers to capture different levels of entity nesting. The span labeling approaches identify and classify candidate spans (that are likely to be entities) within the text. These approaches are classified into enumeration, boundary detection, and MRC, depending on the identification method of candidate spans. Finally,

the other approaches involve models that tackle the nested NER problem in unconventional ways, such as sequence-to-set and affine models.

Due to the limited number of biomedical nested NER datasets, Table 3 includes performance metrics not only for the GENIA [69] and BB19 [70] datasets but also for three additional datasets—ACE04 [80], ACE05 [81], and KBP17 [82]. The performance metrics were taken from the respective method papers. The recent nested NER models demonstrate an average F1-score of 80.8% (±1.6) on the GENIA dataset. On ACE04 and ACE 05, they achieve F1-scores of 87.2% (±1.1) and 86.4% (±2.1), respectively, and perform at an average of 83.7% (±1.6) on the KBP17 dataset. BB only has performance results reported by SpanMB, with an F1-score of 81.8%. Additionally, it can be observed that the models show overall lower performance on the GENIA corpus compared to the ACE04 and ACE05 datasets, which involve more general terms.

Table 3. A comprehensive list of methods for nested NER.

Approach Type		Method	Performance (F1-Score)					Code	Ref.
			GENIA	BB19	ACE04	ACE05	KBP17		
	Layer-based	Merge Label	- [+]	-	-	82.40	-	Yes	[83]
		Pyramid	79.19	-	86.28	84.66	-	Yes	[84]
Span labeling	Enumeration	PURE	-	-	88.10	88.70	-	Yes	[85]
		SpanMB	-	81.80	-	-	-	Yes	[86]
		PL-Marker	-	-	88.80	89.80	-	Yes	[87]
	Boundary	BENSC	78.30	-	85.30	83.90	-	No	[88]
		Locate-and-Label	80.54	-	87.41	86.67	84.05	Yes	[89]
	MRC	BERT-MRC	83.75	-	85.98	86.88	80.97	Yes	[90]
		PIQN	81.77	-	88.14	87.42	84.50	Yes	[91]
Others	Sequence-to-set	Sequence-to-Set	80.44	-	87.26	87.05	83.96	Yes	[92]
		PnRNet	81.85	-	88.12	87.63	85.27	Yes	[93]
	Affine	Biaffine	80.50	-	86.70	85.40	-	Yes	[94]
		Triaffine	81.23	-	87.40	86.82	-	Yes	[95]

[+]—means that the model does not report performance for the given dataset.

This section also introduces models based on a two-step structure consisting of an encoding and decoding layer, similar to those described in Section 3.

4.1. Layer-Based Approaches

The layer-based methods handle the hierarchy of nested entities by stacking multiple NER layers, designed to predict entities of a specific length or level (Figure 3a). The model progressively predicts entities by stacking layers, starting either from the layer that predicts the shortest (innermost) entities to the one that predicts the longest (outermost) entities, or vice versa. For instance, assume a model structured to predict entities from the innermost to the outermost levels. The NER layer 1 identifies entities that do not embed other entities within them. Following this, the NER layer 2 detects entities that embed those identified by the preceding layer. This process continues for subsequent layers, enabling the model to cover progressively broader spans of entities.

Fisher and Vlachos proposed Merge and Label, a layer-based model that identifies a set of entities by level [83]. The model determines the entity spans by assessing whether adjacent tokens and/or entities belong to the same entity, assigning a continuous real value to this prediction. The predicted real values are used as weights for the words and/or entities within the span, enabling a weighted sum calculation to derive the span embedding. These embeddings are then used to classify the spans. The model proceeds by passing tokens and/or spans to subsequent layers, thereby enabling the gradual formation of larger spans.

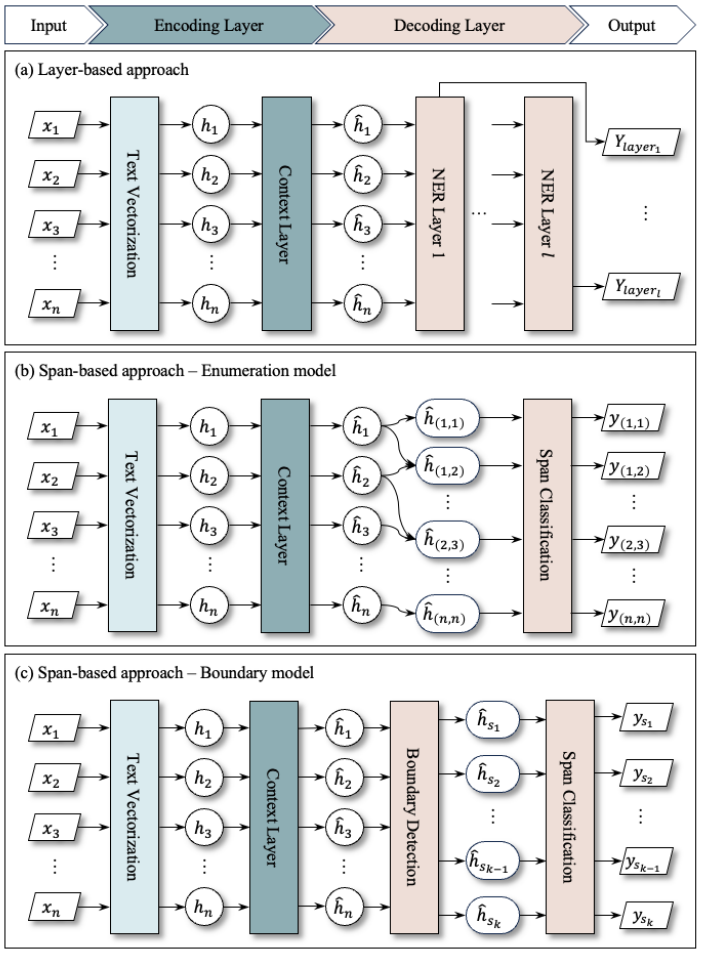

Figure 3. Overview of three approaches for nested NER. For each approach, the diagram shows how an input sequence $\{x_1, \ldots, x_n\}$, where x_i indicates the i-th word in an input sentence, is transformed through encoding and decoding layers to predict nested NEs. The encoding layer consists of a text vectorization, which converts words or tokens into numerical vectors, and a context layer that uses neural networks such as LSTM or transformer to capture broader linguistic context. The decoding layer then predicts NEs based on the context from the encoding layer. (**a**) Layer-based approach. h_i and $\hat{h}_i$ refer to the initial and latent embeddings for x_i, respectively. Y_{layer_i} denotes the output from i-th NER layer. (**b**) Enumeration model of span-based approach. $\hat{h}_{(i,j)}$ indicates the latent embedding for the span from x_i to x_j, and $y_{(i,j)}$ refers to a class assigned to that span. (**c**) Boundary model of span-based approach. $\hat{h}_{s_i}$ represents the latent embedding for i-th candidate span determined by the boundary detection layer. y_{s_i} denotes the label assigned to the i-th candidate span.

Wang et al. developed a layered model pyramid, which predicts complete entity mentions of the corresponding length in each layer [84]. The model consists of multiple interconnected layers and employs a convolutional network with two kernels between the layers to combine information from adjacent words and/or spans, forming a pyramid structure. Furthermore, the model operates in both forward and backward directions through an inverse pyramid, providing bidirectional interactions between adjacent layers. This model alleviates common issues of the general layer-based approaches, such as layer disorientation and error propagation. These studies demonstrate that the hierarchical structures of nested NEs can be handled in intuitive ways.

4.2. Span Labeling Approaches

The span labeling approach addresses the nested NER problem by assigning labels to all candidate spans likely to be NEs (Figure 3b,c). This approach detects all NEs simultaneously, mitigating the error propagation problem. In this section, we classify span labeling

approaches into three categories: enumeration, boundary, and MRC. The enumeration and boundary models are distinguished by how they generate candidate spans. The enumeration model sets all spans shorter than a certain length as candidates, whereas the boundary model uses neural networks to generate candidate spans. The MRC model differs from the other two by incorporating information about entity type using queries.

4.2.1. Enumeration Model

The enumeration model considers all spans shorter than a threshold as potential NE candidates (Figure 3b). All NE candidates are transformed into high-quality representations based on the span representation strategy and then classified as either specific entity types or non-entities through a span classification layer. The enumeration model is a straightforward approach that can consider all valid spans. Consequently, its performance heavily depends on the quality of the span representation.

Zhong and Chen define a span representation as a concatenation of the head and tail token representations, along with a learned representation of its width feature [85]. This concatenation strategy has the disadvantage of ignoring information from intermediate tokens within the span. To overcome the limitation, Zuo et al. employ a max-pooling representation [86]. The max-pooling representation is defined as collecting the maximum values for each dimension across the token representations within the span. It is then concatenated with the representations of the head token, the tail token, and the width feature to construct the span representation.

The concatenation and pooling representations are defined by combining token representations generated by the encoding layer. However, these strategies have limitations in accounting for dependencies among spans. Ye et al. emphasized the importance of these interrelationships and introduced a packed levitated marker (PL-Marker) [87]. In their model, each span is assigned two levitated markers that contain the position information of its head and tail tokens, respectively. These markers are paired and appended to the end of a sentence with other levitated marker pairs from adjacent spans together. The sentence, along with the group of levitated markers, is passed through the encoding layer. Subsequently, the representations of the encoded head and tail levitated markers are concatenated together and utilized as a span representation.

Enumeration approaches can consider as many spans as possible. However, this approach is inherently expensive due to the large number of spans processed. The representation strategies that account for dependencies between spans, such as PL-marker, further increase computational and time costs. Additionally, these approaches are constrained by their inability to predict long entities that exceed a certain threshold. While enumeration methods can intuitively handle nested NEs, their critical disadvantage lies in the substantial resource requirements, making them less efficient for time-sensitive applications.

4.2.2. Boundary Model

The boundary approach involves additional layers, called boundary detection, to generate NE candidates, as illustrated in Figure 3c. These candidates are then assigned labels through the span classification layer. In this section, we discuss the research by focusing on the additional component, the boundary detection layer.

Tan et al. introduced a boundary-enhanced neural span classification (BENSC) model that employs two token classifiers within the boundary detection layer [88]. The two token classification layers predict whether the token is a head or tail word of the entity, respectively. Then, all valid head-tail pairs, i.e., where the tail follows the head, are labeled through the span classification layer. This model has the advantage of being able to predict longer entities, as it does not impose a length limitation. However, it presents the issue of error propagation, where inaccuracies in the boundary detection layer carry over to the next stages.

In general, span-based models treat prediction results that do not exactly match the correct span boundaries as incorrect. However, Shen et al. argue that this rigid

consideration can introduce noise into the model [89]. To address this issue, they introduced boundary regression, inspired by object detection techniques in computer vision. Their model enumerates all spans that are shorter than a specific threshold and assigns them as candidate spans. The candidates go through two steps: filtration and refinement. In the initial step, the low-quality candidates that do not closely overlap with the actual NEs are filtered out using a binary classification layer. In the subsequent step, the boundaries of the remaining high-quality candidates are refined using a regression layer. The candidate spans adjusted in the two steps are finally classified through a span classification layer. This model is not free from the length limitation, as it employs the enumeration strategy for generating candidates in the initial stage. Although the excessive computational resource consumption associated with enumeration is mitigated through the filtration step, this introduces an error propagation issue. Nevertheless, the model remains notable for its use of approximate boundary refinement, allowing for adjustments in span boundaries even when initial predictions are not perfectly accurate.

The primary advantage of these boundary models lies in their ability to narrow the candidate pool, thereby reducing computational time complexity and resource consumption. Although its overall performance may be lower compared to the enumeration approach, the model's resource efficiency makes it well-suited for real-world applications where computation constraints are a key consideration.

4.2.3. MRC

The MRC-based approaches typically receive information about the entity type to be predicted in the form of a query along with the sentence and then predict the boundary of the corresponding type (Figure 2b).

BERT-MRC utilizes the queries in natural language format [90]. The model passes the concatenation of the sentence with the query through an encoding layer, allowing the information from the query to be integrated into the sentence representation. BERT-MRC employs three binary classifiers to handle nested NEs. The two classifiers are designed to identify candidates for head and tail tokens of NEs, respectively. Subsequently, the last classifier determines whether each head-tail pair is an appropriate target for the query. BERT-MRC generates each query for one specific entity type, encountering the limitation that it can handle only one entity type per inference. Therefore, it fails to consider interrelationships among entities of different types. Moreover, the model involves a manual process for crafting appropriate queries for each entity type, adding to its complexity.

To overcome these shortcomings, Shen et al. designed the parallel instance query network (PIQN) [91]. PIQN employs instance queries that are learnable vectors, diverging from the traditional natural language form. These instance queries are initialized with random values and autonomously learn their semantic significance during the training phase. The sentence is encoded independently, whereas the instance queries interact with the sentence's embedding through cross-attention mechanisms. This allows the queries to extract relevant contextual information without altering the original sentence encoding. PIQN utilizes query embeddings to identify and classify NEs instead of using sentence embeddings. For each query, two separate linear layers are used to predict the start and end positions of the span corresponding to the query. After determining the span boundaries, a span classification layer is applied to categorize the span into a specific entity type. In other words, each query is designed to identify only a single entity, ensuring that one query corresponds to one entity prediction. The process for multiple queries occurs in parallel, allowing the model to efficiently handle multiple entities within the same sentence.

This section introduces two models that utilize different query formats: traditional natural language queries and learnable vector formats. BERT-MRC, which employs traditional natural language queries, provides the model with precise, user-defined information. However, the manual crating of these queries requires domain-specific expertise, and the model's performance is highly dependent on the quality of the queries. In contrast, PIQN leverages learnable vector format queries that dynamically acquire information about

entities during training. This model enhances flexibility and adaptability across various entity types without the need for manual input. Furthermore, the parallel instance query mechanism in PIQN allows for the simultaneous prediction of multi-type NEs, capturing correlations between types and significantly reducing inference time.

4.3. Other Nested NER Approaches

This section introduces two atypical approaches for nested NER: the sequence-to-set and the dependency parsing-based approaches. Tan et al. [92] and Wu et al. [93] focused on being order-agnostic of NEs and accordingly introduced the sequence-to-set approach. Tan et al. identify in a single pass by aligning an input sentence with a set of learnable entity queries. The self-attention and cross-attention layers are applied for aligning, which capture the relationships between entity queries and between the relationships between the sentence and each entity query, respectively. For each query that learns context information through this process, the model identifies the head and tail indices and class. Wu et al. introduced the propose-and-refine network (PnRNet) that follows a similar process for aligning a sentence with a set of entity queries. PnRNet utilizes advanced representations of entity queries and a sentence. Concretely, the model enumerates all spans from 1 to l grams in a sentence and then selects the top k most probable ones as entity queries. In addition, PnRNet constructs the multi-scale sentence representation by concatenating representations of 1 to l grams. The k entity queries and the multi-scale sentence are processed through the attention layers with a process identical to that used by Tan et al. [92]. These sequence-to-set approaches take advantage of the fact that NEs are inherently order-independent, processing multiple entities in a single pass. These models are particularly suitable for learning dependencies among NEs.

A graph-based dependency parsing aims to analyze the syntactic structure of a sentence by predicting the head of each token and recognizing its relationship to the head. This task inspired the development of a nested NER model. Yu et al. [94] employed a biaffine mechanism developed by Dozat and Manning [96] for dependency parsing. The biaffine mechanism applied to a sentence produces a class score tensor for all word pairs that can form a NE, providing a global view of the sentence. However, this mechanism cannot consider additional information, such as the relationships among spans. To consider such heterogeneous factors, Yuan et al. [95] introduced a triaffine mechanism that advanced from a biaffine mechanism. Their model uses the triaffine mechanism to fuse various factors such as tokens, boundaries, labels, and related spans. These studies demonstrate that well-designed features and model structures are still helpful for complex tasks such as nested NER.

5. Analysis of Entity Types and Span Representation

This section performed three analyses to provide guidance for generating BioNER models. The first analysis examined the morphological similarities between different biomedical types in practical datasets. Recent studies have built multi-type NER models. However, biomedical entities often show close similarities, which is reflected in their word forms. We observed this phenomenon in practical datasets since it can confuse multi-type NER models. Second, we compared the performance of different span representation strategies. The span representation strategy significantly affects performance in nested NER models. Therefore, we compiled strategies from the previous studies and compared their performances. Finally, we evaluated the performance of re-context and decoding layers in PLM-based NER models. Since the advent of PLM-based transfer learning, this approach has become the standard for solving various NLP problems, and models that utilized PLM with an additional context layer (hereinafter referred to as the re-context layer) have been introduced in several studies. Nevertheless, there has been no study analyzing the performance of the re-context layer in PLM-based models. Therefore, we aimed to evaluate the impact of these re-context layers on the NER model. We also conducted

performance comparisons among various networks for the decoding layer, which is the component that produces the final output.

5.1. Similarities between Different Entity Types

Biomedical entities are often closely related across various types, and this characteristic is also reflected in word forms. These morphological similarities across different types may confuse the NER model. Therefore, this section observed the morphological similarities in practical datasets. For experiments, we integrated the seven widely available datasets: BioRED [43], JNLPBA [46], NCBI disease [65], BC2GM [54], BC4CHEMD [62], BC5CDR [51], and Linnaeus [66]. All NEs within the datasets underwent a series of preprocessing steps, which included lemmatization, decapitalization, and the removal of special characters. Consequently, their types were normalized into eight classes: gene/protein, DNA, RNA, cell, disease, chemical, species, and variant. We defined the similarity of set A to B with respect to entity type as follows [97]:

$$Similarity(A,\ B) = \frac{1}{|A|}\sum_{a\in A}\max_{b\in B} DSC(a,b), \tag{5}$$

$$DSC(a,\ b) = \frac{2|a\cap b|}{|a|+|b|}. \tag{6}$$

Here, a and b represent word collections of each entity. The similarity between word collections is calculated using the Dice-Sørensen coefficient (DSC). Note that this similarity measure is not symmetric for the entity types A and B.

Figure 4 illustrates the morphological textual similarities between biomedical entity types, revealing several noteworthy correlations. The strong correlations were observed among gene/protein, DNA, RNA, and cell entities. The pair of RNA and gene/protein exhibits the highest similarity score of 0.69. Pairs of DNA and gene/protein and RNA and DNA also showed high similarity with scores of 0.66 and 0.56, respectively. These outcomes are likely due to the functional and structural interconnectivity among them. For example, in 'myeloid cells (cell)', the 'Mcl-1 (gene/protein)' is regulated by its promoter 'Mcl-1 promoter (DNA)', which controls the transcription of 'Mcl-1 mRNA (RNA)'. These morphological similarities can potentially increase the risk of model boundary misclassification when processing entities of that type simultaneously.

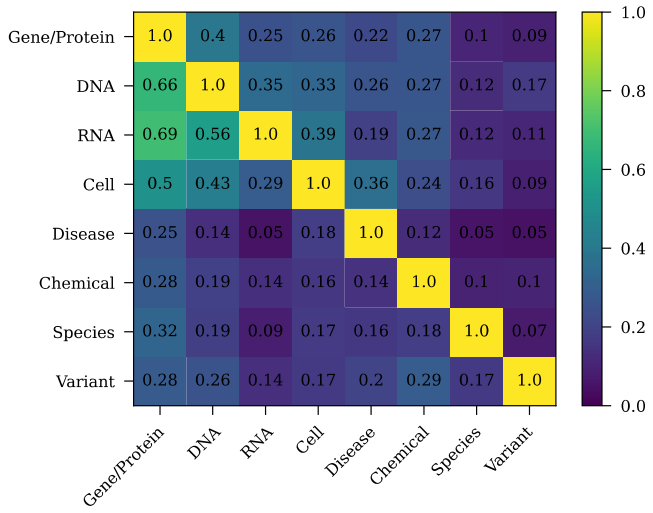

Figure 4. Morphological textual similarity between biomedical entity types. Strong correlations were observed between gene/protein, DNA, RNA, and cell types exhibiting high biological similarity. The highest similarity is observed between RNA and gene/protein (0.69), followed by DNA and gene/protein (0.66). RNA and DNA also exhibit significant similarity (0.56). Cell correlates notably with gene/protein (0.5) and DNA (0.43).

Another important consideration is that homonymous entities may be classified into different types depending on the dataset. For instance, 'angiotensin' is classified as a gene or gene product in BioRED and BC2GM but as a chemical in BC4CHEMD and BC5CDR. Additional examples are provided in Supplementary Table S2. Recently, many studies have focused on integrating multiple datasets to develop multi-type NER models [74,76]. However, our results reveal an inconsistency of type among the datasets, which inevitably hampers model performance. This discrepancy underscores the critical need for standardized annotation guidelines to ensure consistency across datasets. Additionally, post-processing techniques are necessary to maintain compatibility when integrating different existing datasets. Establishing these guidelines and addressing inconsistencies are essential for improving the accuracy and robustness of multi-type NER models.

5.2. Strategies for Obtaining Span Representations

The span representation strategy is a key factor in span-based models and significantly impacts model performance. This section evaluated and compared four span representation strategies, such as concatenation, max-pooling, mean-pooling, and PL-marker. Because previous studies have performed comparisons with subsets of these strategies, this experiment aims to provide a comprehensive evaluation of all four strategies.

The concatenation, max-pooling, and mean-pooling strategies generate span representations based on token representations. Formally, given a sentence of n tokens, an input is denoted as $X = \{x_1, x_2, \ldots, x_n\}$, and their latent embeddings are denoted as $H = \left\{\hat{h}_1, \hat{h}_2, \ldots, \hat{h}_n\right\}$. The candidate spans are defined based on the enumerate method as $span(X) = \left\{s_{x_1,x_1}, \ldots, s_{x_1,x_l}, \ldots, s_{x_{n-l+1},x_n}, \ldots, s_{x_n,x_n}\right\}$, where l is the maximum span length. Given s_{x_i,x_j}, the concatenation [85], max-pooling, and mean-pooling [98] methods are as follows:

$$Concat\left(s_{x_i,x_j}\right) = \left\{\hat{h}_i; \hat{h}_j\right\}, \tag{7}$$

$$Max\text{-}pooling\left(s_{x_i,\,x_j}\right) = \left\{\hat{h}_i; max\left(\hat{h}_i, \ldots, \hat{h}_j\right); \hat{h}_j\right\}, \tag{8}$$

$$Mean\text{-}pooling\left(s_{x_i,\,x_j}\right) = \left\{\hat{h}_i; mean\left(\hat{h}_i, \ldots, \hat{h}_j\right); \hat{h}_j\right\}. \tag{9}$$

Here, ; denotes the concatenation operation.

The PL-markers refer to individual spans and learn their contextual information [87]. Specifically, the span candidate s_{x_i,x_j} is assigned a pair of markers $m^{(s)}_{x_i,x_j}$ and $m^{(e)}_{x_i,x_j}$, respectively referring to the start and end tokens of the span. A set of these markers is provided as input to the model along with a sentence. The latent embeddings of the marker are then employed as the representation of the span. Consequently, the PL-marker of span s_{x_i,x_j} is defined as follows [87]:

$$PL\text{-}Marker\left(s_{x_i,x_j}\right) = \left\{\hat{h}^{(s)}_{m_{x_i,x_j}}; \hat{h}^{(e)}_{m_{x_i,x_j}}\right\}. \tag{10}$$

The experiment was performed on three datasets: GENIA [69], BB19 [70], and SciERC [99]. SciERC is not a biomedical dataset, but it is based on scientific corpora and contains nested NEs. The experiments were conducted on a single NVIDIA Tesla V100 DGXS 32GB GPU (produced by NVIDIA Corporation in Santa Clara, CA, USA) using bioformers/bioformer-16L model. A batch size of 8 and a learning rate of 3×10^{-5} were applied. The training was performed for 20 epochs for the GENIA dataset, while the BB19 and SciERC datasets were trained for 50 epochs. To ensure robustness, each experiment was repeated with five different seeds, and the average precision, recall, and F1-scores were used for performance comparison (Table 4). The concatenation strategy achieved an overall lower F1-score. Specifically, the differences from best strategies were substantial: -0.69 in GENIA, -1.96 in BB19, and -2.08 in SciERC. These results demonstrate the effectiveness of pooling and marker strategies. The strategy that demonstrated consistently high performance was mean-pooling, which achieved the highest F1-score in the BB19 dataset

and the second highest in both the GENIA and SciERC datasets. On the other hand, the PL-Marker strategy showed the best performance in the GENIA and SciERC datasets but ranked third in the BB19 dataset. These results indicate that the choice of strategy should consider the characteristics of the data. Additionally, although the PL-Marker strategy requires additional computational resources because it allocates two markers per span, it proved its worth by outperforming the second-best strategies by 0.58 in GENIA and 0.88 in SciERC. Therefore, future work should balance performance and computational cost when selecting a span representation strategy.

Table 4. Comparison of span classification performance according to span representation strategies.

Span Repr. Strategy	GENIA			BB19			SciERC		
	P	R	F	P	R	F	P	R	F
Concatenation	78.07	76.96	77.47	78.31	72.63	75.35	65.78	67.91	66.78
Max-pooling	78.11	76.73	77.40	79.52	73.80	76.55	**67.57**	68.09	67.81
Mean-pooling	**78.84**	76.37	77.58	**79.92**	**74.89**	**77.31**	67.52	68.48	67.98
PL-Marker	78.34	**78.01**	**78.16**	79.17	73.25	76.09	67.53	**70.27**	**68.86**

Bold indicates the best scores, and underline indicates second-best scores.

5.3. Contribution of Re-Context and Decoding Layers

Most recent NLP models adopt transfer learning based on PLM. For instance, as illustrated in Figure 2a, the PLM is applied to both the text vectorization and context layer. Several studies implement a re-context layer on top of PLM to capture additional contextual information. This layer may include a recurrent neural network (RNN) or CNN. Lastly, the decoding layer generates corresponding labels for downstream tasks based on the representation.

In this section, we introduced the widely used re-context and decoding layers in flat NER models and analyzed their individual impacts. The experiments were performed on three datasets: NCBI disease [65], BC2GM [54], and BC5CDR chemical [51]. Each model was trained and tested on five seeds, and the averaged precision, recall, and F1-score were used for comparison. The bold numbers indicate the best scores, and the underlined numbers represent the second-best scores. All experiments utilized a single NVIDIA Tesla V100 DGXS 32GB GPU for training the NER models. The bioformers/bioformer-16L model was used as the PLM. A batch size of 32 and a learning rate of 3×10^{-5} were employed. The training was performed for between 20 and 30 epochs, with early stopping triggered after 5 consecutive epochs without improvement.

5.3.1. Comparison of Re-Context Layers

Re-context layers have been employed to obtain additional contextual information; however, to the best of our knowledge, their effectiveness has not been comprehensively evaluated since the advent of PLMs. This section evaluates the effects of six re-context layers on NER tasks: Bi-LSTM, 1-dimension CNN (1dCNN), 2-dimension CNN (2dCNN), BiLSTM-attention (BiLSTM-attn), 1dCNN-attn, and 2dCNN-attn. Bi-LSTM, 1dCNN, and attention mechanisms have traditionally been applied to text data, whereas 2dCNN has primarily been employed in image-related tasks due to its ability to capture local patterns and features. However, since Banerjee et al. suggested that the localized perspective of 2dCNN can enhance the performance of NER tasks [75], our NER experiments included it as a re-context layer.

In our experimental setup, the input dimensions were in the shape of (sentence length, embedding size). The 1dCNN layer had a kernel size of 5, a padding size of 2, and an output channel size double that of the input size. The 2dCNN layer was designed with a kernel size of (5, 5), a stride size of (1, 2), and circular padding of (2, 256). The settings for the 2dCNN layer were referenced in a previous study [75]. The decoding layer utilized a *softmax* layer for sequence labeling, following the IOB tagging format.

Table 5 compares the performance of six models with re-context layers to the base model, which is a model without any re-context layer. The majority of models with the re-context layer generally outperformed the base model (except for 2dCNN and BiLSTM-attn on BC5CDR chemical); however, the differences from the base model were not significant. Specifically, the differences from the best model were −0.43 in NCBI disease, −0.44 in BC2GM, and −0.39 in BC5CDR chemical. We also observed that the performance differences between re-context layers were not substantial. These results suggest that the PLM already contains sufficient contextual information for NER tasks, so the additional re-context layers may offer only marginal improvements. Therefore, the focus should be on optimizing other components of the NER to achieve better performance.

Table 5. Comparison of the flat NER performance according to the re-contextualization layer.

Re-Context Layer	NCBI Disease			BC2GM			BC5CDR Chemical		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Base model	86.14	89.10	87.60	82.85	84.08	83.46	92.36	93.46	92.90
1dCNN	**86.93**	89.08	87.99	82.98	84.14	83.55	92.67	**93.89**	93.28
2dCNN	86.70	88.94	87.80	83.19	84.44	83.81	92.42	93.29	92.86
BiLSTM	86.91	89.19	**88.03**	83.05	84.17	83.60	92.72	93.38	93.05
1dCNN-attn	86.73	**89.31**	88.00	83.02	**84.64**	83.82	**92.99**	93.28	93.13
2dCNN-attn	86.33	88.96	87.63	83.02	84.30	83.65	92.84	93.75	**93.29**
BiLSTM-attn	86.52	89.21	87.84	**83.43**	84.37	**83.90**	92.41	93.28	92.84

Bold indicates the best scores, and underline indicates second-best scores.

5.3.2. Comparison of Decoder Layers

In this section, we trained and evaluated four decoding layers frequently used in NER models: *softmax*, CRF, RNN, and span labeling. *Softmax*, CRF, and RNN are employed for the sequence labeling approach, which is common for flat NER tasks. On the other hand, the span labeling layer is not as commonly used for flat NER. Nonetheless, we included it in the experiment to observe the adaptability of the span labeling approach to flat NER.

Softmax is favored for its computational efficiency and straightforward implementation. It is used to assign probabilities to each class for individual tokens independently. On the other hand, CRF and RNN capture the dependencies between tokens. CRF learns transition probabilities between adjacent tokens, capturing that certain tags are likelier to follow others. RNN maintains information about previous tokens when predicting the tag of the current token, capturing sequential dependencies. Both CRF and RNN require higher computational costs than *softmax*. On the other hand, the span labeling layer was adapted for flat NER tasks in this experiment. The model with a span labeling layer enumerated candidate spans of up to 8 words, generating span representations using a concatenation strategy. The model then classified these candidate spans based on their representations. To ensure a flat structure, nested entities were consolidated by prioritizing the later and longer spans that are classified as entities.

According to Table 6, the RNN and CRF exhibited high overall performance. Specifically, RNN performed best on the NCBI disease and BC5CDR chemical datasets, and CRF secured the second-best performance across all datasets. This indicates that dependency information between tokens is valuable in the NER model. Notably, the span labeling approach showed low performance on the NCBI disease and BC5CDR chemical datasets, achieving high performance only on BC2GM. There is even a +0.86% difference from *softmax*. These results are intriguing, considering the simplicity of the post-process of handling nested entities. However, it should be noted that the span labeling layer incurs a higher computational cost than CRF and RNN due to the generation and classification of numerous candidate spans. Therefore, CRF and RNN can be partial and effective choices when considering both computational cost and performance.

Table 6. Comparison of the flat NER performance according to the decoding layer.

Decoding Layer	NCBI Disease			BC2GM			BC5CDR Chemical		
	P	R	F	P	R	F	P	R	F
SoftMax	86.14	89.10	87.60	82.85	84.08	83.46	92.36	93.46	92.90
CRF	86.04	<u>89.27</u>	<u>87.62</u>	83.49	**84.77**	<u>84.13</u>	<u>92.74</u>	<u>93.68</u>	<u>93.20</u>
RNN	**87.32**	**89.40**	**88.35**	<u>83.61</u>	<u>84.45</u>	84.02	92.49	**93.96**	**93.22**
Span labeling	<u>87.06</u>	87.48	87.27	**85.36**	83.31	**84.32**	**93.38**	92.43	92.91

Bold indicates the best scores, and <u>underline</u> indicates second-best scores.

6. Discussion

After the emergence of LLMs, BioNERs have shown significant improvements, with flat NER models achieving around 90% accuracy for most datasets and nested NER models achieving around 80%. However, there are still considerable limitations that hinder their application in real-world environments, particularly in terms of the lack of diverse datasets and generalization across datasets.

First, lack of data remains a key issue. The scarcity of domain-specific datasets, especially for nested NER tasks, presents a significant challenge. There are only two available corpora for biomedical nested NER, GENIA and BB19, which hinder generalizing the models across different data. While flat NER has more datasets, there is a lack of multi-type corpora. Manually creating large-scale datasets with annotation is labor-intensive and expensive. To overcome these challenges, data augmentation and meta-learning models are actively researched. Data augmentation systems attempt to generate synthetic examples or transform existing data, while the meta-learning models enable effective performance with limited labeled examples. Despite the progress in these areas, the performance of these methods has not yet reached a sufficient level for practical use in real-world applications [100–102].

Secondly, generalization across datasets presents another challenge. A model that performs well on one dataset often sees its performance decline when applied to other datasets within the same domain (e.g., NCBI disease to BC5CDR disease) [71]. This decline is often due to differences in annotation schemes, boundary definitions, and entity types between datasets. Models that struggle to maintain consistent performance across multiple datasets are less reliable for real-world applications. To address this, there is a need for models that can generalize effectively across various datasets and handle these discrepancies.

7. Conclusions

This review comprehensively analyzes flat and nested NER tasks in the biomedical domain, underscoring recent advancements and persisting challenges. While LLMs have markedly improved the performance of flat NER tasks, substantial challenges remain to address the complexities of nested NER. Our analysis highlighted the limitations of nested entity processing strategies that require excessive resources and utilize span representation methods that lack contextual considerations. Additionally, the limited availability of annotated datasets containing diverse entity types is a significant bottleneck for scaling NER tasks. Furthermore, annotation inconsistencies across different corpora hinder models from achieving reliable performance on datasets of similar types. Future research efforts should focus on developing robust models that can maintain consistent performance across different datasets and achieve high-quality learning with limited resources. Addressing these challenges is essential for building reliable BioNER systems that can be effectively used in real-world biomedical applications.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/app14209302/s1, Table S1: Performance evaluation of traditional approaches and LLM-based models on the flat NER datasets; Table S2: Examples of the type discrepancy among biomedical datasets.

Author Contributions: Conceptualization, Y.P., G.S. and M.R.; investigation, Y.P. and G.S.; writing—original draft preparation, Y.P. and G.S.; writing—review and editing, Y.P. and M.R.; supervision, M.R.; funding acquisition, M.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partly supported by Korea Institute of Marine Science & Technology Promotion (KIMST) funded by the Ministry of Oceans and Fisheries, Korea (20220517) and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2020-II201373, Artificial Intelligence Graduate School Program (Hanyang University)).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yamada, K.; Miwa, M.; Sasaki, Y. Biomedical Relation Extraction with Entity Type Markers and Relation-specific Question Answering. In Proceedings of the 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks, Toronto, ON, Canada, 13–14 July 2023; pp. 377–384.
2. Frisoni, G.; Italiani, P.; Moro, G.; Bartolini, I.; Boschetti, M.A.; Carbonaro, A. Graph-Enhanced Biomedical Abstractive Summarization via Factual Evidence Extraction. *SN Comput. Sci.* **2023**, *4*, 500. [CrossRef]
3. Lai, P.-T.; Wei, C.-H.; Luo, L.; Chen, Q.; Lu, Z. BioREx: Improving biomedical relation extraction by leveraging heterogeneous datasets. *J. Biomed. Inform.* **2023**, *146*, 104487. [CrossRef] [PubMed]
4. Al-Moslmi, T.; Ocaña, M.G.; Opdahl, A.L.; Veres, C. Named entity extraction for knowledge graphs: A literature overview. *IEEE Access* **2020**, *8*, 32862–32881. [CrossRef]
5. Yang, S.; Yoo, S.; Jeong, O. DeNERT-KG: Named entity and relation extraction model using DQN, knowledge graph, and BERT. *Appl. Sci.* **2020**, *10*, 6429. [CrossRef]
6. Park, Y.; Lee, J.; Moon, H.; Choi, Y.S.; Rho, M. Discovering microbe-disease associations from the literature using a hierarchical long short-term memory network and an ensemble parser model. *Sci. Rep.* **2021**, *11*, 4490. [CrossRef]
7. Grishman, R.; Sundheim, B.M. Message understanding conference-6: A brief history. In Proceedings of the COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics, Copenhagen, Denmark, 5–9 August 1996.
8. Li, J.; Sun, A.; Han, J.; Li, C. A survey on deep learning for named entity recognition. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 50–70. [CrossRef]
9. Campos, D.; Matos, S.; Oliveira, J.L. Biomedical named entity recognition: A survey of machine-learning tools. *Theory Appl. Adv. Text Min.* **2012**, *11*, 175–195.
10. Wang, X.; Yang, C.; Guan, R. A comparative study for biomedical named entity recognition. *Int. J. Mach. Learn. Cybern.* **2018**, *9*, 373–382. [CrossRef]
11. Song, B.; Li, F.; Liu, Y.; Zeng, X. Deep learning methods for biomedical named entity recognition: A survey and qualitative comparison. *Brief. Bioinform.* **2021**, *22*, bbab282. [CrossRef]
12. Gaizauskas, R. Term recognition and classification in biological science journal articles. In Proceedings of the Workshop on Computational Terminology for Medical and Biological Applications, Patras, Greece, 2–4 June 2000.
13. Song, M.; Yu, H.; Han, W.-S. Developing a hybrid dictionary-based bio-entity recognition technique. *BMC Med. Inform. Decis. Mak.* **2015**, *15*, 1–8. [CrossRef]
14. Zhou, G. Recognizing names in biomedical texts using mutual information independence model and SVM plus sigmoid. *Int. J. Med. Inform.* **2006**, *75*, 456–467. [CrossRef] [PubMed]
15. Alex, B.; Haddow, B.; Grover, C. Recognising nested named entities in biomedical text. In Proceedings of the Biological, Translational, and Clinical Language Processing, Prague, Czech Republic, 29 June 2007; pp. 65–72.
16. Leaman, R.; Wei, C.-H.; Lu, Z. tmChem: A high performance approach for chemical named entity recognition and normalization. *J. Cheminform.* **2015**, *7*, S3. [CrossRef] [PubMed]
17. Gridach, M. Character-level neural network for biomedical named entity recognition. *J. Biomed. Inform.* **2017**, *70*, 85–91. [CrossRef] [PubMed]
18. Yoon, W.; So, C.H.; Lee, J.; Kang, J. Collabonet: Collaboration of deep neural networks for biomedical named entity recognition. *BMC Bioinform.* **2019**, *20*, 55–65. [CrossRef]
19. Cho, H.; Lee, H. Biomedical named entity recognition using deep neural networks with contextual information. *BMC Bioinform.* **2019**, *20*, 735. [CrossRef]
20. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
21. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.R.; Le, Q.V. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv* **2019**, arXiv:1906.08237.
22. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.

23. Lee, J.; Yoon, W.; Kim, S.; Kim, D.; Kim, S.; So, C.H.; Kang, J. BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* **2020**, *36*, 1234–1240. [CrossRef]
24. Naseem, U.; Musial, K.; Eklund, P.; Prasad, M. Biomedical named-entity recognition by hierarchically fusing biobert representations and deep contextual-level word-embedding. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–8.
25. Beltagy, I.; Peters, M.E.; Cohan, A. Longformer: The long-document transformer. *arXiv* **2020**, arXiv:2004.05150.
26. Sanh, V. DistilBERT, A Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. *arXiv* **2019**, arXiv:1910.01108.
27. Rohanian, O.; Nouriborji, M.; Kouchaki, S.; Clifton, D.A. On the effectiveness of compact biomedical transformers. *Bioinformatics* **2023**, *39*, btad103. [CrossRef] [PubMed]
28. Sang, E.F.; De Meulder, F. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. *arXiv* **2003**, arXiv:cs/0306050.
29. Ratinov, L.; Roth, D. Design challenges and misconceptions in named entity recognition. In Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009), Boulder, CO, USA, 4–5 June 2009; pp. 147–155.
30. Ramshaw, L.A.; Marcus, M.P. Text chunking using transformation-based learning. In *Natural Language Processing Using Very Large Corpora*; Springer: Berlin/Heidelberg, Germany, 1999; pp. 157–176.
31. Lample, G. Neural architectures for named entity recognition. *arXiv* **2016**, arXiv:1603.01360.
32. Finkel, J.R.; Manning, C.D. Nested named entity recognition. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Singapore, 6–7 August 2009; pp. 141–150.
33. Wang, Y.; Tong, H.; Zhu, Z.; Li, Y. Nested named entity recognition: A survey. *ACM Trans. Knowl. Discov. Data (TKDD)* **2022**, *16*, 108. [CrossRef]
34. Olson, D.L.; Delen, D. *Advanced Data Mining Techniques*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2008.
35. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013, Lake Tahoe, NV, USA, 5–10 December 2013.
36. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
37. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [CrossRef]
38. Chen, X.; Xu, L.; Liu, Z.; Sun, M.; Luan, H. Joint learning of character and word embeddings. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015.
39. Peters, M.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep contextualized word representations. *arXiv* **2018**, arXiv:1802.05365.
40. Huang, K.; Altosaar, J.; Ranganath, R. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv* **2019**, arXiv:1904.05342.
41. Gu, Y.; Tinn, R.; Cheng, H.; Lucas, M.; Usuyama, N.; Liu, X.; Naumann, T.; Gao, J.; Poon, H. Domain-specific language model pretraining for biomedical natural language processing. *ACM Trans. Comput. Healthc. (HEALTH)* **2021**, *3*, 2. [CrossRef]
42. Fang, L.; Chen, Q.; Wei, C.-H.; Lu, Z.; Wang, K. Bioformer: An efficient transformer language model for biomedical text mining. *arXiv* **2023**, arXiv:2302.01588.
43. Luo, L.; Lai, P.T.; Wei, C.H.; Arighi, C.N.; Lu, Z. BioRED: A rich biomedical relation extraction dataset. *Brief. Bioinform.* **2022**, *23*, bbac282. [CrossRef] [PubMed]
44. Mohan, S.; Li, D. Medmentions: A large biomedical corpus annotated with umls concepts. *arXiv* **2019**, arXiv:1902.09476.
45. Bada, M.; Eckert, M.; Evans, D.; Garcia, K.; Shipley, K.; Sitnikov, D.; Baumgartner, W.A., Jr.; Cohen, K.B.; Verspoor, K.; Blake, J.A.; et al. Concept annotation in the CRAFT corpus. *BMC Bioinform.* **2012**, *13*, 161. [CrossRef] [PubMed]
46. Kim, J.-D.; Ohta, T.; Tsuruoka, Y.; Tateisi, Y.; Collier, N. Introduction to the bio-entity recognition task at JNLPBA. In Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and Its Applications, Geneva, Switzerland, 28–29 August 2004; pp. 70–75.
47. Neves, M.; Damaschun, A.; Kurtz, A.; Leser, U. Annotating and evaluating text for stem cell research. In Proceedings of the Third Workshop on Building and Evaluation Resources for Biomedical Text Mining (BioTxtM 2012) at Language Resources and Evaluation (LREC), Istanbul, Turkey, 21–27 May 2012; pp. 16–23.
48. Bagewadi, S.; Bobić, T.; Hofmann-Apitius, M.; Fluck, J.; Klinger, R. Detecting miRNA mentions and relations in biomedical literature. *F1000Research* **2014**, *3*, 205. [CrossRef] [PubMed]
49. Nagel, K.; Jimeno-Yepes, A.; Rebholz-Schuhmann, D. Annotation of protein residues based on a literature analysis: Cross-validation against UniProtKb. *BMC Bioinform.* **2009**, *10*, S4. [CrossRef]
50. Thompson, P.; Iqbal, S.A.; McNaught, J.; Ananiadou, S. Construction of an annotated corpus to support biomedical information extraction. *BMC Bioinform.* **2009**, *10*, 349. [CrossRef]
51. Li, J.; Sun, Y.; Johnson, R.J.; Sciaky, D.; Wei, C.-H.; Leaman, R.; Davis, A.P.; Mattingly, C.J.; Wiegers, T.C.; Lu, Z. BioCreative V CDR task corpus: A resource for chemical disease relation extraction. *Database* **2016**, *2016*, baw068. [CrossRef]

52. Miranda, A.; Mehryary, F.; Luoma, J.; Pyysalo, S.; Valencia, A.; Krallinger, M. Overview of DrugProt BioCreative VII track: Quality evaluation and large scale text mining of drug-gene/protein relations. In Proceedings of the Seventh BioCreative Challenge Evaluation Workshop, Virtual Event, 8–10 November 2021; pp. 11–21.
53. Wei, C.-H.; Allot, A.; Riehle, K.; Milosavljevic, A.; Lu, Z. tmVar 3.0: An improved variant concept recognition and normalization tool. *Bioinformatics* **2022**, *38*, 4449–4451. [CrossRef]
54. Smith, L.; Tanabe, L.K.; Kuo, C.-J.; Chung, I.; Hsu, C.-N.; Lin, Y.-S.; Klinger, R.; Friedrich, C.M.; Ganchev, K.; Torii, M. Overview of BioCreative II gene mention recognition. *Genome Biol.* **2008**, *9*, S2. [CrossRef]
55. Wang, X.; Tsujii, J.i.; Ananiadou, S. Disambiguating the species of biomedical named entities using natural language parsers. *Bioinformatics* **2010**, *26*, 661–667. [CrossRef]
56. Gerner, M.; Nenadic, G.; Bergman, C.M. An exploration of mining gene expression mentions and their anatomical locations from biomedical text. In Proceedings of the 2010 Workshop on Biomedical Natural Language Processing, Uppsala, Sweden, 15 July 2010; pp. 72–80.
57. Cejuela, J.M.; Vinchurkar, S.; Goldberg, T.; Prabhu Shankar, M.S.; Baghudana, A.; Bojchevski, A.; Uhlig, C.; Ofner, A.; Raharja-Liu, P.; Jensen, L.J. LocText: Relation extraction of protein localizations to assist database curation. *BMC Bioinform.* **2018**, *19*, 15. [CrossRef] [PubMed]
58. Faessler, E.; Modersohn, L.; Lohr, C.; Hahn, U. ProGene—A large-scale, high-quality protein-gene annotated benchmark corpus. In Proceedings of the 12th Language Resources and Evaluation Conference, Marseille, France, 11–16 May 2020; pp. 4585–4596.
59. Islamaj, R.; Wei, C.-H.; Cissel, D.; Miliaras, N.; Printseva, O.; Rodionov, O.; Sekiya, K.; Ward, J.; Lu, Z. NLM-Gene, a richly annotated gold standard dataset for gene entities that addresses ambiguity and multi-species gene recognition. *J. Biomed. Inform.* **2021**, *118*, 103779. [CrossRef] [PubMed]
60. Kolárik, C.; Klinger, R.; Friedrich, C.M.; Hofmann-Apitius, M.; Fluck, J. Chemical names: Terminological resources and corpora annotation. In Proceedings of the Workshop on Building and Evaluating Resources for Biomedical Text Mining (6th Edition of the Language Resources and Evaluation Conference), Osaka, Japan, 11–16 December 2008.
61. Herrero-Zazo, M.; Segura-Bedmar, I.; Martínez, P.; Declerck, T. The DDI corpus: An annotated corpus with pharmacological substances and drug–drug interactions. *J. Biomed. Inform.* **2013**, *46*, 914–920. [CrossRef]
62. Krallinger, M.; Rabal, O.; Leitner, F.; Vazquez, M.; Salgado, D.; Lu, Z.; Leaman, R.; Lu, Y.; Ji, D.; Lowe, D.M. The CHEMDNER corpus of chemicals and drugs and its annotation principles. *J. Cheminform.* **2015**, *7*, S2. [CrossRef] [PubMed]
63. Islamaj, R.; Leaman, R.; Kim, S.; Kwon, D.; Wei, C.-H.; Comeau, D.C.; Peng, Y.; Cissel, D.; Coss, C.; Fisher, C. NLM-Chem, a new resource for chemical entity recognition in PubMed full text literature. *Sci. Data* **2021**, *8*, 91. [CrossRef] [PubMed]
64. Gurulingappa, H.; Klinger, R.; Hofmann-Apitius, M.; Fluck, J. An empirical evaluation of resources for the identification of diseases and adverse effects in biomedical literature. In Proceedings of the 2nd Workshop on Building and Evaluating Resources for Biomedical Text Mining (7th Edition of the Language Resources and Evaluation Conference), Valetta, Malta, 18 March 2010; pp. 15–22.
65. Doğan, R.I.; Leaman, R.; Lu, Z. NCBI disease corpus: A resource for disease name recognition and concept normalization. *J. Biomed. Inform.* **2014**, *47*, 1–10. [CrossRef] [PubMed]
66. Gerner, M.; Nenadic, G.; Bergman, C.M. LINNAEUS: A species name identification system for biomedical literature. *BMC Bioinform.* **2010**, *11*, 85. [CrossRef] [PubMed]
67. Luoma, J.; Nastou, K.; Ohta, T.; Toivonen, H.; Pafilis, E.; Jensen, L.J.; Pyysalo, S. S1000: A better taxonomic name corpus for biomedical information extraction. *Bioinformatics* **2023**, *39*, btad369. [CrossRef]
68. Thomas, P.E.; Klinger, R.; Furlong, L.I.; Hofmann-Apitius, M.; Friedrich, C.M. Challenges in the association of human single nucleotide polymorphism mentions with unique database identifiers. *BMC Bioinform.* **2011**, *12*, S4. [CrossRef]
69. Kim, J.-D.; Ohta, T.; Tateisi, Y.; Tsujii, J.i. GENIA corpus—A semantically annotated corpus for bio-textmining. *Bioinformatics* **2003**, *19*, i180–i182. [CrossRef]
70. Bossy, R.; Deléger, L.; Chaix, E.; Ba, M.; Nédellec, C. Bacteria biotope at BioNLP open shared tasks 2019. In Proceedings of the 5th Workshop on BioNLP Open Shared Tasks, Hong Kong, China, 4 November 2019; pp. 121–131.
71. Kühnel, L.; Fluck, J. We are not ready yet: Limitations of state-of-the-art disease named entity recognizers. *J. Biomed. Semant.* **2022**, *13*, 26. [CrossRef] [PubMed]
72. Sun, C.; Yang, Z.; Wang, L.; Zhang, Y.; Lin, H.; Wang, J. Biomedical named entity recognition using BERT in the machine reading comprehension framework. *J. Biomed. Inform.* **2021**, *118*, 103799. [CrossRef] [PubMed]
73. Khan, M.R.; Ziyadi, M.; AbdelHady, M. Mt-bioner: Multi-task learning for biomedical named entity recognition using deep bidirectional transformers. *arXiv* **2020**, arXiv:2001.08904.
74. Chai, Z.; Jin, H.; Shi, S.; Zhan, S.; Zhuo, L.; Yang, Y. Hierarchical shared transfer learning for biomedical named entity recognition. *BMC Bioinform.* **2022**, *23*, 8. [CrossRef] [PubMed]
75. Banerjee, P.; Pal, K.K.; Devarakonda, M.; Baral, C. Biomedical named entity recognition via knowledge guidance and question answering. *ACM Trans. Comput. Healthc.* **2021**, *2*, 33. [CrossRef]
76. Luo, L.; Wei, C.-H.; Lai, P.-T.; Leaman, R.; Chen, Q.; Lu, Z. AIONER: All-in-one scheme-based biomedical named entity recognition using deep learning. *Bioinformatics* **2023**, *39*, btad310. [CrossRef]
77. Moscato, V.; Postiglione, M.; Sansone, C.; Sperli, G. Taughtnet: Learning multi-task biomedical named entity recognition from single-task teachers. *IEEE J. Biomed. Health Inform.* **2023**, *27*, 2512–2523. [CrossRef]

78. Moen, S.; Ananiadou, T.S.S. Distributional semantics resources for biomedical text processing. In Proceedings of the LBM 2013, Tokyo, Japan, 12–13 December 2013; pp. 39–44.
79. Jin, Q.; Dhingra, B.; Cohen, W.W.; Lu, X. Probing biomedical embeddings from language models. *arXiv* **2019**, arXiv:1904.02181.
80. Doddington, G.R.; Mitchell, A.; Przybocki, M.A.; Ramshaw, L.A.; Strassel, S.M.; Weischedel, R.M. The automatic content extraction (ace) program-tasks, data, and evaluation. In Proceedings of the LREC, Lisbon, Portugal, 26–28 May 2004; pp. 837–840.
81. Walker, C.; Consortium, L.D. *ACE 2005 Multilingual Training Corpus*; Linguistic Data Consortium: Philadelphia, PA, USA, 2005.
82. Getman, J.; Ellis, J.; Song, Z.; Tracey, J.; Strassel, S.M. Overview of Linguistic Resources for the TAC KBP 2017 Evaluations: Methodologies and Results. In Proceedings of the TAC, Gaithersburg, MD, USA, 13–14 November 2017.
83. Fisher, J.; Vlachos, A. Merge and Label: A novel neural network architecture for nested NER. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019), Florence, Italy, 28 July–2 August 2019; pp. 5840–5850.
84. Wang, J.; Shou, L.; Chen, K.; Chen, G. Pyramid: A layered model for nested named entity recognition. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 5918–5928.
85. Zhong, Z.; Chen, D. A Frustratingly Easy Approach for Entity and Relation Extraction. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online, 6–11 June 2021; pp. 50–61.
86. Zuo, M.; Zhang, Y. A span-based joint model for extracting entities and relations of bacteria biotopes. *Bioinformatics* **2022**, *38*, 220–227. [CrossRef]
87. Ye, D.; Lin, Y.; Li, P.; Sun, M. Packed Levitated Marker for Entity and Relation Extraction. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, 22–27 May 2022; pp. 4904–4917.
88. Tan, C.A.Q.; Qiu, W.; Chen, M.S.; Wang, R.; Huang, F. Boundary Enhanced Neural Span Classification for Nested Named Entity Recognition. *AAAI Conf. Artif. Intell.* **2020**, *34*, 9016–9023. [CrossRef]
89. Shen, Y.; Ma, X.; Tan, Z.; Zhang, S.; Wang, W.; Lu, W. Locate and Label: A Two-stage Identifier for Nested Named Entity Recognition. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Virtual Event, 1–6 August 2021; pp. 2782–2794.
90. Li, X.; Feng, J.; Meng, Y.; Han, Q.; Wu, F.; Li, J. A Unified MRC Framework for Named Entity Recognition. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 5849–5859.
91. Shen, Y.; Wang, X.; Tan, Z.; Xu, G.; Xie, P.; Huang, F.; Lu, W.; Zhuang, Y. Parallel Instance Query Network for Named Entity Recognition. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, 22–27 May 2022; pp. 947–961.
92. Tan, Z.; Shen, Y.; Zhang, S.; Lu, W.; Zhuang, Y. A sequence-to-set network for nested named entity recognition. *arXiv* **2021**, arXiv:2105.08901.
93. Wu, S.; Shen, Y.; Tan, Z.; Lu, W. Propose-and-refine: A two-stage set prediction network for nested named entity recognition. *arXiv* **2022**, arXiv:2204.12732.
94. Yu, J.; Bohnet, B.; Poesio, M. Named Entity Recognition as Dependency Parsing. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 6470–6476.
95. Yuan, Z.; Tan, C.; Huang, S.; Huang, F. Fusing Heterogeneous Factors with Triaffine Mechanism for Nested Named Entity Recognition. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, 22–27 May 2022; pp. 3174–3186.
96. Dozat, T.; Manning, C.D. Deep biaffine attention for neural dependency parsing. *arXiv* **2016**, arXiv:1611.01734.
97. Dice, L.R. Measures of the amount of ecologic association between species. *Ecology* **1945**, *26*, 297–302. [CrossRef]
98. Yu, J.; Ji, B.; Li, S.; Ma, J.; Liu, H.; Xu, H. S-NER: A Concise and Efficient Span-Based Model for Named Entity Recognition. *Sensors* **2022**, *22*, 2852. [CrossRef]
99. Luan, Y.; He, L.; Ostendorf, M.; Hajishirzi, H. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. *arXiv* **2018**, arXiv:1808.09602.
100. Chen, P.; Wang, J.; Lin, H.; Zhao, D.; Yang, Z. Few-shot biomedical named entity recognition via knowledge-guided instance generation and prompt contrastive learning. *Bioinformatics* **2023**, *39*, btad496. [CrossRef]
101. Zhou, R.; Li, X.; He, R.; Bing, L.; Cambria, E.; Si, L.; Miao, C. MELM: Data augmentation with masked entity language modeling for low-resource NER. *arXiv* **2021**, arXiv:2108.13655.
102. Wang, S.; Sun, X.; Li, X.; Ouyang, R.; Wu, F.; Zhang, T.; Li, J.; Wang, G. Gpt-ner: Named entity recognition via large language models. *arXiv* **2023**, arXiv:2304.10428.