Effective Multi-Task Learning for Biomedical Named Entity Recognition

João Ruano, Gonçalo M. Correia, Leonor Barreiros and Afonso Mendes
Priberam Labs, Alameda D. Afonso Henriques, 41, 2º, 1000-123 Lisboa, Portugal
{joao.ruano,goncalo.correia,leonor.barreiros,amm}@priberam.pt

Abstract

Biomedical Named Entity Recognition presents significant challenges due to the complexity of biomedical terminology and inconsistencies in annotation across datasets. This paper introduces SRU-NER (Slot-based Recurrent Unit NER), a novel approach designed to handle nested named entities while integrating multiple datasets through an effective multi-task learning strategy. SRU-NER mitigates annotation gaps by dynamically adjusting loss computation to avoid penalizing predictions of entity types absent in a given dataset.[1] Through extensive experiments, including a cross-corpus evaluation and human assessment of the model's predictions, SRU-NER achieves competitive performance in biomedical and general-domain NER tasks, while improving cross-domain generalization.

1 Introduction

Named entity recognition (**NER**) is a crucial step in several natural language processing pipelines, such as information extraction, information retrieval, machine translation, and question-answering systems (Sharma et al., 2022). Given unstructured text, the task of NER is to identify and classify text spans according to categories of interest. These categories are defined depending on the downstream application and can range from general (*people*, *locations*, *organizations*) to specific domains such as biomedical entities (*genes*, *diseases*, *chemicals*).

In particular, Biomedical Named Entity Recognition (**BioNER**) is challenging due to the complexity of biomedical nomenclature. Morphologically, these entities can contain Greek letters, digits, punctuation (α-*tubulin*, *IL-6*), form variations (*inhibitor* vs. *inhibitory*), and compound terms (*tumor necrosis factor-alpha* vs. *TNF-*α). Semantically, polysemy (e.g., *p53* referring to a gene, protein, or condition) adds ambiguity. These challenges make human annotation costly, leading to BioNER datasets that are smaller and often focus on a limited number of entity types (Greenberg et al., 2018).

One approach to addressing data scarcity while building a BioNER model is to leverage multiple datasets, each annotated with a specific subset of entities. However, simply training a single model on the union of all available datasets assumes that every entity type is consistently annotated across all training instances, which is not the case. This leads to a high prevalence of false negatives, as entities that are labeled in one dataset may be entirely ignored in another. On the other hand, training separate models for each dataset fails to exploit shared statistical patterns across datasets and introduces the challenge of resolving conflicting predictions at inference time (Greenberg et al., 2018). Therefore, an effective strategy must balance learning from multiple sources while accounting for missing annotations and inconsistencies in labeling schemes.

Our contributions are three-fold: (i) we introduce **SRU-NER** (**S**lot-based **R**ecurrent **U**nit NER), a model which is able to solve nested NER through generating a sequence of actions; (ii) we propose an effective multi-task training strategy to handle the complex challenges of leveraging multiple NER datasets in a single model; and (iii) we show how the SRU-NER can handle multiple datasets on a single shared network through multiple experiments, including cross-corpus evaluations and a human evaluation on corpora of disjoint entity sets.

2 Related work

Named entity recognition has evolved significantly in the last decades. Early systems relied on rule-based methods, which were interpretable but lacked flexibility. The introduction of machine learning enabled more adaptable approaches, further enhanced by deep learning techniques that captured complex

[1] Code is publicly available at https://github.com/Priberam/sru-ner.

Proceedings of the 24th Workshop on Biomedical Language Processing (BioNLP 2025), pages 225–239

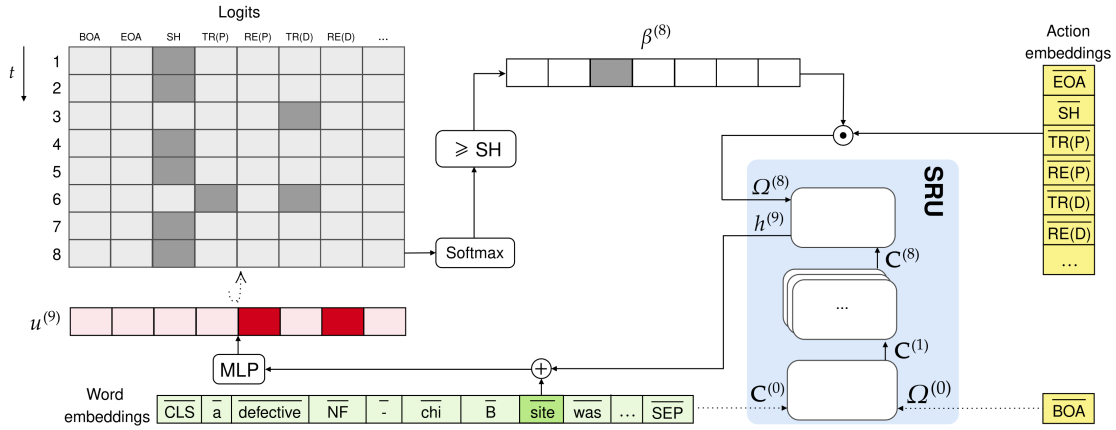

Figure 1: Action selection process for the sentence given in section 3.1, at time step $t = 9$. The gold nested mentions are "NF - chi B site", "chi B", of type *DNA* (*D*), and "NF - chi B" of type *Protein* (*P*). To compute the logits $u^{(9)}$, the model leverages the logits of the previous time steps, action embeddings and word embeddings.

linguistic patterns. Recently, Transformer-based architectures have set new benchmarks, driving significant advancements in NER performance (Li et al., 2022; Keraghel et al., 2024). In the CoNLL-2003 dataset (Tjong Kim Sang and De Meulder, 2003), a benchmark for NER tasks, performance has improved substantially, with F1 scores that have soared above 94% (Wang et al., 2021). The same phenomenon is seen for the GENIA corpus (Kim et al., 2003), a nested BioNER dataset, with test F1 scores exceeding 80% (Yu et al., 2020; Tan et al., 2021; Shen et al., 2021, 2022).

To tackle the proliferation of BioNER datasets, several studies have turned to multi-task learning (**MTL**; Park et al., 2024). Traditional deep learning NER models trained on a single dataset are referred to as *single-task* models, as they specialize in identifying mention spans for the specific entity types annotated within their training data. Single-task models often underperform on out-of-domain settings. In contrast, MTL frameworks leverage multiple datasets, each corresponding to a different *task*, allowing the model to learn from diverse sources. The fundamental premise is that different datasets share information which can be jointly leveraged to encourage the learning of more generalized representations, hence improving a model's robustness (Mehmood et al., 2019; Li et al., 2022).

MTL learning frameworks can be categorized into two types, depending on which modules are shared across tasks: (i) those that share the encoding layers while maintaining task-specific decoding layers (Crichton et al., 2017; Wang et al., 2018; Khan et al., 2020), and (ii) those that share *all* layers (Greenberg et al., 2018; Huang et al., 2019; Banerjee et al., 2021; Luo et al., 2023; Moscato et al., 2023). SRU-NER resembles models of type (ii), which share its decoding layers across all tasks. Typically, these models have a natural problem with false negatives, as the unified decoder may struggle to distinguish task-specific entity boundaries and labels, leading to the omission of valid entities. Our approach avoids this issue through an effective multi-task learning strategy.

3 Effective Multi-Task Learning for Named Entity Recognition

The proposed model, SRU-NER, solves the task of nested named entity recognition similar to that of a transition-based parser (Dyer et al., 2015; Marinho et al., 2019). Given a sequence of words $S = [w_1, w_2, \ldots, w_N]$, the model generates a sequence of *actions*. At each time step, the actions are chosen depending on the words of the sentence and on the previously chosen actions. At the end of the parsing procedure, the complete sequence of actions is decoded into mentions.

3.1 Action encoding

Consider the system is trained to recognize mentions of entity types belonging to $\mathbb{E} = \{e_1, e_2, \ldots, e_M\}$. Let $\mathcal{A}_\mathbb{E}$ stand for the system's $2M + 2$ possible *actions*: two special tokens (`SH` and `EOA`) and, for each entity type e_i, a pair of actions denoted `TR`(e_i) and `RE`(e_i). `TR`(e_i), short for "transition to entity e_i", indicates the start of a mention of type e_i; one says that this action *opened* a mention of type e_i. `RE`(e_i), short

for "reduce of entity e_i", indicates the end of the mention of type e_i that was opened more recently; one says that a mention was *closed* by this action. SH, short for "shift", indicates that the input pointer should move to the next token; therefore, there is one SH for each word in the sentence. Finally, EOA is the end action.

These actions encode nested mentions effectively through the order in which they are chosen. If a mention of type e_k starts at the word w_i and ends at the word w_j, TR(e_k) appears before the SH representing the i-th word, and RE(e_j) appears after the SH representing the k-th word; if two mentions start at the same word, the TR() of the longest mention appears first; conversely, if two mentions end at the same word, the RE() of the shortest mention appears first. Consider the following sentence from the GENIA dataset (Kim et al., 2003):

a defective NF - chi B site was completely ...

This sentence has nested mentions, *e.g.* the mention "NF - chi B" of type *Protein* is contained in the mention "NF - chi B site" of type *DNA*. The action encoding of the sentence with its mentions is: SH → SH → TR(*DNA*) → TR(*Protein*) → SH → SH → TR(*DNA*) → SH → SH → RE(*DNA*) → RE(*Protein*) → SH → RE(*DNA*) → SH → SH → ... → EOA.

3.2 Overall architecture

Using the previous notation, suppose one wants to detect mentions of $\mathbb{E}$ in the sentence S. The model consists of three consecutive steps: the encoding of S into a dense contextual embedding matrix $\mathbf{S}$, the iterative action generation procedure, and the decoding of the chosen actions into the mentions present in the sentence.

Contextual embeddings For the first step, S is passed through a BERT-like encoder to generate a matrix of contextual embeddings. For each word w_i, its dense embedding, denoted by $\overline{w_i}$, is obtained by max-pooling across the embeddings of its subwords. In this way, the encoded sentence $\mathbf{S}$ is a tensor of size $(N+2, d_{\text{enc}})$, $\mathbf{S} = \left[\overline{\text{CLS}}, \overline{w_1}, \overline{w_2}, \ldots, \overline{w_N}, \overline{\text{SEP}}\right]$, where d_{enc} is the encoder embedding dimension, $\overline{\text{CLS}}$ (respectively $\overline{\text{SEP}}$) is the embedding of the classification (respectively, separator) token of the encoder.

Action generation Given $\mathbf{S}$, the model enters an iterative action selection process, where at each time step t, logits are computed for each possible action in $\mathcal{A}_{\mathbb{E}}$.[2] Figure 1 shows a schematic representation of a time step of the cycle.

More concretely, define $u_{a_i}^{(t)}$ to be the logit value of action $a_i \in \mathcal{A}_{\mathbb{E}}$ for time step t. Suppose the system has already computed these values for the first $T \geq 1$ time steps, and is therefore about to compute them for time step $t = T+1$. According to the last section, the SH action corresponds to advancing a token in the sentence S. Hence, define

$$p^{(t)} = \sum_{t_0 \leq t} \mathbb{1}\left(\arg\max_{a_i \in \mathcal{A}_{\mathbb{E}}} \left(u_{a_i}^{(t_0)}\right) = \text{SH}\right), \quad (1)$$

where $\mathbb{1}$ stands for the indicator function. $p^{(t)}$ is therefore the number of tokens that have already been parsed at a previous time step t, for $1 \leq t \leq T$. Lastly, define, for each $1 \leq t \leq T$,

$$\Omega^{(t)} = \sum_{a_i \in \mathcal{A}_{\mathbb{E}}} \beta_{a_i}^{(t)}\ \overline{a_i}, \quad (2)$$

where $\overline{a_i}$ is a trained embedding of size d_{enc} and

$$\beta_{a_i}^{(t)} = \begin{cases} u_{a_i}^{(t)} & \text{if } u_{a_i}^{(t)} \geq u_{\text{SH}}^{(t)} \\ 0 & \text{otherwise} \end{cases}$$

In other words, $\Omega^{(t)}$ is a weighted embedding of the actions chosen at time step t, where actions with logits lower than the logit of SH are excluded.

Let $\mathbf{u}^{(T+1)}$ be the vector of logits $u_{a_i}^{(T+1)}$ over $a_i \in \mathcal{A}_{\mathbb{E}}$. These are computed as

$$\mathbf{u}^{(T+1)} = \text{MLP}\left(f\left(p^{(T)}, \Omega^{(T)}\right)\right), \quad (3)$$

where the MLP is composed of a dropout layer, a fully-connected layer, a $\tanh$ activation, and a linear layer with output nodes corresponding to each action in $\mathcal{A}_{\mathbb{E}}$. The input of this MLP is

$$f\left(p^{(T)}, \Omega^{(T)}\right) = \mathbf{S}_{p^{(T)}+1} \oplus \text{SRU}\left(\Omega^{(T)}, p^{(T)}\right),$$

i.e. the concatenation of the embedding of the *next* token, $\mathbf{S}_{p^{(T)}+1}$, and an embedding of the last state

[2] Unlike token-based labeling approaches, the total number of time steps is not determined *a priori*, although always bounded below by N, the number of words in S.

of a "processed actions memory". This memory holds an action history and computes weighted embeddings at each call by leveraging a set of internal latent representations. This module is refered to as the **S**lot-based **R**ecurrent **U**nit (**SRU**), and is described in section 3.3.

In order to make the first prediction, $\mathbf{u}^{(1)}$, the system is initialized by setting $p^{(0)} = 0$, and $\Omega^{(0)}$ to be another trained embedding of size d_{enc}, denoted by $\overline{\texttt{BOA}}$.[3] The action generation cycle terminates when a time step $t = T_{\text{final}}$ is reached such that

$$\text{Sigmoid}\left(u_{\texttt{EOA}}^{(T_{\text{final}})}\right) > 0.5\,. \tag{4}$$

Decoding At the end of the action generation cycle, the output logits from all time steps are passed through a sigmoid function. This produces a set of independent probability scores for each action in $\mathcal{A}_{\mathbb{E}}$, from which mention spans are extracted. The decoder module maintains separate stacks of open spans for each entity type in $\mathbb{E}$, allowing spans of different types to overlap.

The decoding process iterates through the list of probability scores until reaching a time step where the highest-scoring action is `EOA`[4]. Before such a time step is reached, the decoder proceeds following two rules: (i) if the highest-scoring action is `SH`, a pointer that counts the number of parsed words is incremented; and (ii) if the highest-scoring action is a `TR()` or a `RE()`, the entity mention stacks are updated. In the latter case, only actions with probability scores above 0.5 are considered. Transition actions open new spans, while reduce actions close the most recent span of the corresponding entity type, as discussed in section 3.1.

3.3 Slot-based Recurrent Unit

The Slot-based Recurrent Unit (SRU) is a stateful function that, at each time step, takes a pair of inputs, updates its internal state, and produces an output embedding.

At each time step t, the SRU updates its internal state according to

$$\mathbf{C}^{(t+1)} = m\left(\mathbf{C}^{(t)}, \Omega^{(t)}, p^{(t)}\right)\,,$$

where $\mathbf{C}^{(t)} \in \mathbb{R}^{Q\times d}$ is the SRU's internal state matrix, $\Omega^{(t)} \in \mathbb{R}^{d}$ is an input vector, and

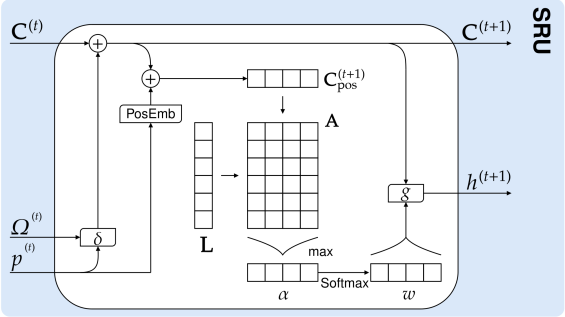

Figure 2: SRU unit at time step t. Its internal state is updated depending on its current state $\mathbf{C}^{(t)}$ and the weighted action embeddings $\Omega^{(t)}$. This stateful function also leverages a set of latent representations. It produces an output embedding $h^{(t+1)}$ by applying an attention mechanism to the updated state.

$p^{(t)} \in \{0, 1, \ldots, Q-1\}$ is an input integer. It also produces an output embedding $h^{(t+1)} \in \mathbb{R}^d$ via

$$h^{(t+1)} = g\left(\mathbf{C}^{(t+1)}, p^{(t)}\right)\,.$$

A schematic representation is present in Figure 2. Q and d refer to the number of rows (or *slots*) in the internal state matrix and the hidden dimension of the input and output embeddings, respectively.

The function m updates $\mathbf{C}^{(t)}$ by summing the input vector $\Omega^{(t)}$ to its $p^{(t)}$-th row, *i.e.*

$$m\left(\mathbf{C}^{(t)}, \Omega^{(t)}, p^{(t)}\right) \coloneqq \mathbf{C}^{(t)} + \delta_{p^{(t)}} \left(\Omega^{(t)}\right)^T$$

where $\delta_{p^{(t)}} \in \mathbb{R}^Q$ is a one-hot vector with 1 in its $p^{(t)}$-th coordinate.

The output embedding $h^{(t)} \in \mathbb{R}^d$ is obtained via the function g, defined as

$$g\left(\mathbf{C}^{(t+1)}, p^{(t)}\right) \coloneqq \mathbf{w}^T \left(\mathbf{C}^{(t+1)}\,\mathbf{D}_1\right)$$

where $\mathbf{D}_1$ is a trained diagonal matrix of size d and $\mathbf{w} \in \mathbb{R}^Q$ are weights computed via an attention mechanism inspired by Ganea and Hofmann, 2017, detailed as follows. First, $\mathbf{C}^{(t+1)}$ is enhanced by adding positional information,

$$\mathbf{C}_{\text{pos}}^{(t+1)} = \alpha\,\mathbf{C}^{(t+1)} + \text{Dropout}\left(\mathbf{P}\left(p^{(t)}\right)\right) \tag{5}$$

where α is a trained scaling parameter, and $\mathbf{P}\left(p^{(t)}\right) \in \mathbb{R}^{Q\times d}$ are positional embeddings.[5]

[3] In this text, a zero-indexing notation is adopted for tensors, and so $\mathbf{S}_{p_0+1} = \overline{w_1}$.

[4] This stopping condition was shown to provide better results empirically, despite being different to that of the action generation procedure, present in equation (4).

[5] These positional embeddings are *relative*, in the sense that each row of $\mathbf{P}\left(p^{(t)}\right)$ is selected from a table of trained embeddings based on its distance to the row with index $p^{(t)}$.

Next, a set of J trained latent embeddings of size d are used to compute an attention score for each row in $\mathbf{C}^{(t+1)}$. Defining $\mathbf{L} \in \mathbb{R}^{J \times d}$ to be the matrix of latent embeddings, an attention score matrix is computed by

$$\mathbf{A} = \text{Dropout}(\mathbf{L})\ \mathbf{D}_2\ \left(\mathbf{C}_{\text{pos}}^{(t+1)}\right)^T,$$

where $\mathbf{D}_2$ is a trained diagonal matrix of size d. An attention score for each slot is obtained by setting $\mathbf{s} = \max_j(A_{jq})$ for $q \in \{0, 1, \ldots, Q-1\}$. Finally, the scores $\mathbf{s}$ are normalized through a softmax to get the weights $w \in \mathbb{R}^Q$.

The SRU module is used at each action generation time step to compute an embedding that models the current state of a "processed actions memory" stack. For each time step t, the input integer $p^{(t)}$ is the one defined by equation (1), and the input vector $\Omega^{(t)}$ is the one defined by equation (2). Furthermore, d is set to be the encoder embedding dimension d_{enc}, the number of slots to be $Q = N+2$, and the number of latent variables J to be an integer multiple[6] of $|\mathcal{A}_{\mathbb{E}}| = 2M+2$. The internal state matrix is initialized by setting $\mathbf{C}^{(0)} = \mathbf{S}$. Taking this choice of initialization into account, and referring back to equation (3), for the computation of $h^{(T+1)} = \text{SRU}\left(\Omega^{(T)}, p^{(T)}\right)$, all the slots of the updated internal state matrix $\mathbf{C}^{(T+1)}$ are filled with the embeddings of the encoded sentence $\mathbf{S}$. In addition, a history of the previously chosen actions is present in $\mathbf{C}^{(T+1)}$ since, at each call of the SRU module in previous time steps $0 \leq t \leq T$, the weighted action embeddings $\Omega^{(t)}$ of equation (2) were summed to the slots pointed to by $p^{(t)}$.

4 Multi-task training strategy

Suppose the model is trained on an ensemble of K datasets $\mathcal{D} = \{D_i\}_{i=1}^K$, where each dataset D_i is annotated with spans of entity types $\mathbb{E}_i$. In order to account for differences in labeling schemes, during training, the entity types of distinct datasets are always considered to be distinct as well.[7] Therefore, the model is trained to recognize spans of entity types in the disjoint union set $\widehat{\mathbb{E}} = \bigsqcup_{i=1}^K \mathbb{E}_i$.

The training objective of the model is to minimize the mean loss of the samples in a batch. Each batch is constructed by randomly selecting samples from $\mathcal{D}$. To ensure a balanced contribution from all datasets, the probability of selecting a sample from a given dataset is inversely proportional to the total number of sentences in that dataset. The total number of samples per epoch is the average number of sentences in the datasets of $\mathcal{D}$.

Let S be a sentence in the batch, coming from dataset D_i, and thus annotated with gold spans of entity types $\mathbb{E}_i$. The output of the action generation cycle is a matrix

$$\mathbf{U} = \left(u_{a_i}^{(t)}\right)_{t=1,\ldots,T_{\text{final}};\, a_i \in \mathcal{A}_{\mathbb{E}}},$$

where each row $u_*^{(t)}$ contains the logits, for time step t, over *all* actions $\mathcal{A}_{\widehat{\mathbb{E}}}$ associated with the disjoint union set $\widehat{\mathbb{E}}$.[8] To compute a loss value for $\mathbf{U}$, the following constraints are enforced:

i) on one hand, the model *should* be penalized for failing to predict the `TR()` and `RE()` actions that correspond to the gold spans of the entity types $\mathbb{E}_i$, for which S is annotated; but

ii) on the other hand, the model *should not* be penalized for predicting `TR()` and `RE()` actions of entity types in $\widehat{\mathbb{E}} \setminus \mathbb{E}_i$, which are not annotated in S.

In practice, this strategy is applied as follows. The list of actions corresponding to the gold annotations of sentence S (constructed as detailed in section 3.1 and considering the disjoint entity type set $\widehat{\mathbb{E}}$) is augmented to a matrix $\mathbf{G} = \left(G_{a_i}^{(t)}\right) \in \mathbb{R}^{T_{\text{initial}} \times |\mathcal{A}_{\widehat{\mathbb{E}}}|}$ such that each row $G_*^{(t)}$ is a multi-hot vector representing a distinct timestep t, with 1's in the columns that correspond to the gold actions. This conversion is done such that the `SH` and `EOA` actions always occupy different time steps, but `TR()` and `RE()` actions of different entity types can coexist at the same time step. Then, $\mathbf{G}$ is changed during the action generation cycle by incorporating the probabilities of the model's decisions on `TR()` and `RE()` actions from other datasets. More concretely, at time step t of the cycle, for $a_i \in \mathcal{A}_{\widehat{\mathbb{E}}} \setminus \mathcal{A}_{\mathbb{E}_i}$, $G_{a_i}^{(t)}$ is set to be equal to $\sigma\left(u_{a_i}^{(t)}\right)$, where σ is the sigmoid function. In addition, when $G_{\texttt{SH}}^{(t)} = 1$ and $u_{a_i}^{(t)} > u_{\texttt{SH}}^{(t)}$ for some

[6] For the experiments conducted, it was set to 2 or 10 (see Table 12 in Appendix B).

[7] In practice, this is implemented by simply changing the name of an entity type $e \in \mathbb{E}_i$ belonging to D_i, to i_e in $\mathbb{E}$.

[8] At inference time, the action generation procedure halts when the probability of the `EOA` action exceeds a threshold, as described in section 3.2. However, during training, in order to guarantee that all gold actions are considered, the cycle halts only after all tokens have been parsed (*i.e.* shifted).

Dataset	SRU-NER *Merged*	SRU-NER *Disjoint*	Wang et al., 2018	Huang et al., 2019	Khan et al., 2020	Moscato et al., 2023
BC2GM	78.80	83.95	80.74 *	79.1	83.01 *	**84.84**
BC4CHEMD	90.42	**92.05**	89.37 *	87.3	—	—
BC5CDR	89.37	**90.26**	88.78 *	—	89.50 *	◇
JNLPBA	72.15	76.00	73.52 *	**83.8**	72.89 *	—
Linnaeus	**88.82**		—	83.9	—	—
NCBI Disease	87.32	88.71	86.14 *	84.0	88.10 *	**89.20**
Average	84.48	86.63				

Table 1: Micro-F1 scores of several multi-task models trained on subsets of an ensemble of six biomedical datasets. For SRU-NER, scores are reported by considering two evaluation scenarios, *Merged* and *Disjoint*, as explained in section 5.2. Best scores are **bold**, and second best scores are underlined. Symbol reference:
— : dataset was absent in training;
* : model was trained on both the training and development splits of the corpora;
◇ : model was trained using only the 'Chemical' annotations of BC5CDR, obtaining an F1 of 93.95; for the same tag, SRU-NER gets an F1 of 93.77 on the disjoint evaluation and 93.18 on the merged evaluation.

$a_i \in \mathcal{A}_{\widehat{\mathbb{E}}} \setminus \mathcal{A}_{\mathbb{E}_i}$, that is, when the model is trying to open/close a new span of an entity type of other dataset D_j ($j \neq i$), the value $G^{(t)}_{\mathsf{SH}}$ is changed to $\sigma\left(u^{(t)}_{\mathsf{SH}}\right)$. In this case, a one-hot vector is inserted in $\mathbf{G}$ after $G^{(t)}_{*}$, so that, on the next time step $t+1$, $G^{(t+1)}_{\mathsf{SH}} = 1$ and $G^{(t+1)}_{a_i} = 0$ for all $a_i \in \widehat{\mathbb{E}} \setminus \{\mathsf{SH}\}$. This procedure ensures that $\mathbf{G}$ still reflects the original gold annotations in the columns corresponding to `TR()` and `RE()` actions of entity types in the source dataset, but incorporates the model's probabilities for other actions. Then, by setting, for each $1 \leq t \leq T_{\text{final}}$,

$$L^{(t)} = -\frac{1}{|\mathcal{A}_{\widehat{\mathbb{E}}}|} \sum_{a_i \in \mathcal{A}_{\widehat{\mathbb{E}}}} \left(G^{(t)}_{a_i} \log\left(\sigma\left(u^{(t)}_{a_i}\right)\right) + \left(1 - G^{(t)}_{a_i}\right) \log\left(1 - \sigma\left(u^{(t)}_{a_i}\right)\right) \right)$$

the total loss of the sample is computed as

$$L = \frac{1}{T_{\text{final}}} \sum_{t=1}^{T_{\text{final}}} L^{(t)} .$$

Given how $\mathbf{G}$ is constructed, this ensures the aforementioned constraints i) and ii) on the loss function are satisfied.

5 Experiments and Results

To evaluate the performance of the proposed architecture for the NER task, single-task experiments were conducted on benchmarks datasets, specifically the English subset of CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003) and GENIA (Kim et al., 2003). The model's multi-task performance is also assessed by training it with an ensemble of six biomedical datasets that have been extensively used in previous research. In order to demonstrate the viability of SRU-NER for downstream applications, a model is evaluated in a cross-corpus setting by replicating the experimental setup of Sänger et al., 2024. Finally, two further experiments are conducted to quantify the reliability of the multi-task models' predictions for entity types that are not explicitly annotated in the test corpora, providing a more comprehensive assessment of their generalization capabilities.

The datasets used across the following sections and respective experimental setup are described in Appendix A. Training details can be found in Appendix B. For evaluation purposes, a predicted mention is considered a true positive if and only if both its span boundaries and entity type exactly match the gold annotation. Results are reported for each dataset using mention-level micro F1 scores.

5.1 Single-task performance

The results of the two single-task models are presented in Table 2. The proposed model achieves micro F1 scores of 94.48% on the CoNLL-2003 dataset, and 80.10% on the GENIA dataset. These results are very close to state-of-the-art (SOTA), demonstrating the competitiveness of SRU-NER in both flat and nested NER scenarios. Nonetheless, in contrast to our approach, the models presented as SOTA were trained using both the training and development splits of their respective datasets. This difference in training data availability may contribute to the observed performance gap, particu-

larly on GENIA, where additional annotated data could provide further benefits in capturing complex biomedical terminology.

Dataset	SRU-NER	SOTA
CoNLL	94.48	94.6*, (Wang et al., 2021)
GENIA	80.10	81.53*, (Shen et al., 2023)

Table 2: Micro-F1 scores of single-task models on benchmark datasets. The entity counts of the datasets can be found in Table 7. The * symbol indicates that a model was trained on both the training and development splits of the corpus.

Dataset	SRU-NER	SOTA
BC2GM	85.43	85.48* (Sun et al., 2021)
BC4CHEMD	92.64	92.92* (Sun et al., 2021)
BC5CDR	90.61	91.90 (Zhang et al., 2023)
JNLPBA	77.12	78.93* (Sun et al., 2021)
Linnaeus	89.62	94.13 (Habibi et al., 2017)
NCBI Disease	89.25	90.04* (Sun et al., 2021)
Average	87.45	

Table 3: Micro-F1 scores of single-task models trained on the datasets used for the multi-task model described in section 5.2. SOTA results are for single-task models. The * symbol indicates the model was trained on a larger training split.

5.2 Multi-task performance

In Table 1, we show the results of SRU-NER trained on an ensemble $\{D_i\}_{i=1}^{6}$ of six biomedical datasets, annotated for $|\cup_i \mathbb{E}_i| = 8$ entity types. Since there are entity types which are annotated on more than one dataset (*e.g.* BC4CHEMD and BC5CDR are both annotated with mentions of the Chemical type), two evaluation scenarios are considered, that differ in how these type labels are interpreted. Recalling that the model infers mentions with entity types in the disjoint union set $\widehat{\mathbb{E}} = \sqcup_i \mathbb{E}_i$, given a sentence coming from the test split of dataset D_i of the ensemble, in the case of:

i) *disjoint evaluation*, the predicted spans of types $\mathbb{E}_i \subset \widehat{\mathbb{E}}$ are compared against the gold ones, and any predicted span of type in $\widehat{\mathbb{E}} \setminus \mathbb{E}_i$ is discarded;

ii) *merged evaluation*, the entity types of predicted spans are mapped to $\cup_i \mathbb{E}_i$, and the spans whose mapped types do not also belong in $\mathbb{E}_i$ are discarded; the remaining spans are compared against the gold ones.

An example of the predictions of the model on a test sentence, together with which spans are used to compute the metrics on the two evaluation scenarios is present in Figure 3.

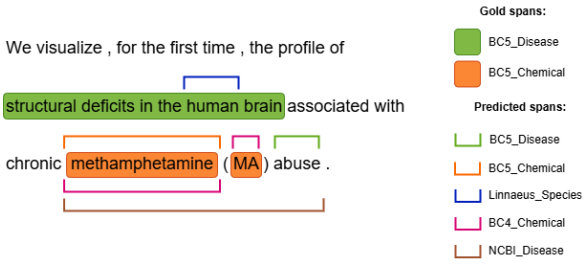

Figure 3: Example of a sentence from the test split of the BC5CDR corpus (Li et al., 2016), together with gold spans and predicted spans as annotated by the MTL model described in section 5.2. The model is trained on six datasets, covering eight entity types $\cup_i \mathbb{E}_i = \{\text{Chemical}, \text{Disease}, \ldots\}$. Notice that some of these types are common to multiple datasets (namely, 'Chemical', annotated on both the BC4CHEMD and BC5CDR datasets; and 'Disease', annotated on both the BC5CDR and NCBI datasets). SRU-NER tags spans with one of 11 possible types, built by adjoining the dataset name to the original type name, such that $\widehat{\mathbb{E}} = \{\text{BC4\_Chemical}, \text{BC5\_Chemical}, \ldots\}$. In the *disjoint* evaluation case, and since this sentence comes from the BC5CDR corpus, metrics are computed by considering only the spans whose types in $\widehat{\mathbb{E}}$ start with the BC5 shorthand, resulting in one true positive, one false positive and two false negatives. In the *merged* evaluation case, spans whose types in $\widehat{\mathbb{E}}$ do not end with 'Chemical' or 'Disease' are discarded, and the remaining spans have their types mapped to $\cup_i \mathbb{E}_i$ by removing the dataset identifier. With these spans, there are two true positives, two false positives and one false negative in the sentence.

Compared to previous MTL models, the proposed model achieves the best or second-best F1 scores in the disjoint evaluation setting. These results are obtained without relying on task-specific classification layers (Wang et al., 2018; Khan et al., 2020) or training multiple single-task teacher models followed by knowledge distillation into a student model (Moscato et al., 2023). Instead, a single unified model learns each task directly from its respective annotated dataset while preserving the performance of other tasks. This approach enables joint decoding, thereby eliminating the need for post-processing steps to resolve span conflicts.

Table 3 presents F1 scores for single-task models trained on each dataset used in the multi-task setting, alongside SOTA references. The results demonstrate that the proposed model remains com-

petitive in the single-task setting. The average F1 score of the six single-task SRU-NER models is 0.82 percentage points higher than the dataset-average F1 of the multi-task SRU-NER model under the disjoint evaluation setting. This aligns with previous findings, which suggest that while multi-task training improves model robustness across datasets, it may lead to lower in-corpus performance compared to single-task models (Yin et al., 2024). To further investigate the generalization capabilities of the model, the next section presents an evaluation in a cross-corpus setting.

Dataset	Entity type	SRU-NER	Baseline
BioID	Species	**62.41**	58.21
MedMentions	Chemical	**59.53**	58.40
	Disease	**62.48**	62.18
tmVar3	Gene	**90.38**	87.87
Average		**68.70**	66.67

Table 4: Mention-level F1 scores for the cross-corpus experiment. SRU-NER was trained on an emsemble of 8 biomedical datasets, and evaluated on 3 independent corpora. Baseline refers to the scores obtained by (Sänger et al., 2024). Best scores are in **bold**.

Training datasets	Chemical	Disease
Only BC5-Chemical	91.27	—
Only BC5-Disease	—	85.41
Both	**91.81**	**86.10**

Table 5: Global prediction F1 scores on the test split of BC5CDR of models trained on synthetic datasets. Best scores are **bold**.

5.3 Cross-corpus evaluation

Table 4 presents the results of the proposed model in a cross-corpus evaluation, replicating the experimental setup of Sänger et al., 2024. The model was trained on an ensemble of nine datasets covering five entity types and evaluated on three independent corpora annotated for four of these types. The results indicate that SRU-NER outperforms the baseline by an average of 2.03%, with notable improvements for the Species (4.2%) and Gene (2.51%) entity types. These findings underscore the robustness of the model and demonstrate its potential for downstream applications. For reference, in-corpus F1 scores are provided in Appendix C.

5.4 Evaluation of global predictions

The previous experiments evaluated the model's *local* prediction ability. Specifically, when the model is trained on a collection $\{D_i\}_{i=1}^{K}$, where each dataset D_i was annotated for entity types $\mathbb{E}_i$, its performance was assessed on a test dataset D_{test} annotated with entity types $\mathbb{E}_{\text{test}} \subseteq \mathbb{E}_j$ for some $j \in \{1, \ldots, K\}$. However, the model generates predictions for spans of all entity types in $\cup_i \mathbb{E}_i$ within D_{test}. To evaluate its global prediction ability, it is necessary to test the model on a dataset annotated with a superset of entity types spanning multiple training datasets.

First, following the approach of Huang et al., 2019, a synthetic dataset is constructed from the BC5CDR corpus. The original training set is randomly partitioned into two disjoint subsets: one containing only Chemical annotations (*BC5-Chemical*) and another containing only Disease annotations (*BC5-Disease*). Additional details on these synthetic datasets are provided in Appendix A. Two single-task models are trained separately on each subset, while a multi-task model is trained on both. All models are evaluated on the original test split of the BC5CDR corpus. The results, presented in Table 5, demonstrate that the training strategy outlined in section 4 effectively enables the model to make accurate global predictions across entity types from different training datasets.

Secondly, a multi-task model is trained on both the CoNLL-2003 dataset and the BC5CDR dataset. This approach results in a model capable of recognizing six entity types: four from the general domain (*LOC*, *MISC*, *ORG*, *PER*) and two from the biomedical domain (*Chemical*, *Disease*). To assess the model's ability to generalize across domains, its predictions of general-domain entity types in the test split of the BC5CDR dataset and, conversely, its predictions of biomedical entity types in the test split of the CoNLL dataset are evaluated. The results of the multi-task model can be found in Table 6 under the column SRU-NER-MTL. Since gold annotations for these cross-domain predictions are not available, the evaluation was conducted manually by two human annotators. Provided with definitions of the entity types, they independently assessed whether the model's predictions were correct. This human evaluation was also conducted for the predictions of two single-task models: one trained on CoNLL-2003 and evaluated on the BC5CDR test set (SRU-NER-CoNLL), and another trained on BC5CDR and evaluated on the CoNLL-2003 test set (SRU-NER-BC5). A comparison between the single-task and multi-task models reveals that multi-task SRU-NER is, on average,

Entity	SRU-NER-CoNLL			SRU-NER-BC5			SRU-NER-MTL		
	P	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*
Chemical	24.71	87.76	38.57	—	—	—	75.00	9.18	16.36
Disease	25.25	83.33	38.76	—	—	—	88.46	38.33	53.49
LOC	—	—	—	98.25	88.89	93.33	100.00	96.83	98.39
ORG	—	—	—	80.00	80.00	80.00	86.36	71.25	78.08
PER	—	—	—	94.44	94.44	94.44	100.00	22.22	36.36

Table 6: Human evaluation of the out-of-domain predictions made by three models. *P* stands for precision, *R* for the simulated recall, and *F1* for the F1 computed with the former two metrics. Details on how these metrics were computed can be found in Appendix D.

25.4% more precise in identifying out-of-domain spans. For instance, the single-task model trained on biomedical entity types incorrectly classified *lead* as a chemical in the CoNLL-2003 sentence: *"Indonesian keeper Hendro Kartiko produced a string of fine saves to prevent the Koreans increasing their lead."* In contrast, the multi-task model did not make this error. Further details on this experiment are provided in Appendix D.

6 Conclusion

This work presents SRU-NER, a novel architecture for Named Entity Recognition capable of handling nested entities through a transition-based parsing approach. The model integrates a Slot-based Recurrent Unit (SRU) to maintain an evolving representation of past actions, enabling effective entity extraction. Unlike traditional multi-task learning approaches that rely on separate models for different entity types, SRU-NER employs a unified learning strategy, allowing a single model to learn from multiple datasets. This approach improves adaptability to annotation inconsistencies and enhances generalization across domains.

Experimental results demonstrate that SRU-NER achieves strong performance in both single- and multi-task settings, with cross-corpus evaluations and human assessments confirming the robustness of its predictions. These findings highlight the advantages of training a single multi-task model for BioNER and suggest promising directions for future research, including advancements in nested entity recognition and domain adaptability.

Limitations

While the proposed SRU-NER architecture has demonstrated effectiveness for named entity recognition in general and biomedical domains, its performance in other domains, such as legal or financial, was not evaluated. Furthermore, the generalizability of the findings may be limited, as evaluations on community-available biomedical datasets may not fully capture the diversity of real-world biomedical text. Finally, the assessment of global prediction ability in a cross-domain scenario relied on human annotators, introducing a degree of subjectivity to the evaluation.

While the model achieves competitive results, we note that no extensive hyperparameter search was conducted. A more systematic tuning of hyperparameters could potentially yield further improvements. Additionally, the training strategy presents opportunities for refinement, notably in the sampling strategy utilized within the multi-task learning framework.

Acknowledgments

This research was supported by the Portuguese Recovery and Resilience Plan through project C645008882-00000055 (*i.e.*, the Center For Responsible AI).

References

Cecilia Arighi, Lynette Hirschman, Thomas Lemberger, Samuel Bayer, Robin Liechti, Donald Comeau, and Cathy Wu. 2017. Bio-id track overview. In *BioCreative VI Challenge Evaluation Workshop*, volume 482, page 376.

Pratyay Banerjee, Kuntal Kumar Pal, Murthy Devarakonda, and Chitta Baral. 2021. Biomedical Named Entity Recognition via Knowledge Guidance and Question Answering. *ACM Trans. Comput. Healthcare*, 2(4):33:1–33:24.

Nigel Collier, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Jin-Dong Kim. 2004. 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 (NLPBA/BioNLP)*, pages 73–78, Geneva, Switzerland. COLING.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Gamal Crichton, Sampo Pyysalo, Billy Chiu, and Anna Korhonen. 2017. A neural network multi-task learning approach to biomedical named entity recognition. *BMC Bioinformatics*, 18(1):368.

Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. 2014. NCBI disease corpus: a resource for disease name recognition and concept normalization. *J Biomed Inform*, 47:1–10.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 334–343, Beijing, China. Association for Computational Linguistics.

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep joint entity disambiguation with local neural attention. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2619–2629, Copenhagen, Denmark. Association for Computational Linguistics.

Martin Gerner, Goran Nenadic, and Casey M. Bergman. 2010. Linnaeus: A species name identification system for biomedical literature. *BMC Bioinformatics*, 11(1):85.

Nathan Greenberg, Trapit Bansal, Patrick Verga, and Andrew McCallum. 2018. Marginal likelihood training of BiLSTM-CRF for biomedical named entity recognition from disjoint label sets. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2824–2829, Brussels, Belgium. Association for Computational Linguistics.

Harsha Gurulingappa, Roman Klinger, Martin Hofmann-Apitius, and Juliane Fluck. 2010. An empirical evaluation of resources for the identification of diseases and adverse effects in biomedical literature. In *2nd Workshop on Building and evaluating resources for biomedical text mining (7th edition of the Language Resources and Evaluation Conference)*, Valetta, Malta.

Maryam Habibi, Leon Weber, Mariana Neves, David Luis Wiegandt, and Ulf Leser. 2017. Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*, 33(14):i37–i48.

Xiao Huang, Li Dong, Elizabeth Boschee, and Nanyun Peng. 2019. Learning a unified named entity tagger from multiple partially annotated corpora for efficient adaptation. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 515–527, Hong Kong, China. Association for Computational Linguistics.

Rezarta Islamaj, Robert Leaman, Sun Kim, Dongseop Kwon, Chih-Hsuan Wei, Donald C Comeau, Yifan Peng, David Cissel, Cathleen Coss, Carol Fisher, Rob Guzman, Preeti Gokal Kochar, Stella Koppel, Dorothy Trinh, Keiko Sekiya, Janice Ward, Deborah Whitman, Susan Schmidt, and Zhiyong Lu. 2021a. NLM-Chem, a new resource for chemical entity recognition in PubMed full text literature. *Sci Data*, 8(1):91.

Rezarta Islamaj, Chih-Hsuan Wei, David Cissel, Nicholas Miliaras, Olga Printseva, Oleg Rodionov, Keiko Sekiya, Janice Ward, and Zhiyong Lu. 2021b. NLM-Gene, a richly annotated gold standard dataset for gene entities that addresses ambiguity and multi-species gene recognition. *J Biomed Inform*, 118:103779.

Imed Keraghel, Stanislas Morbieu, and Mohamed Nadif. 2024. Recent advances in named entity recognition: A comprehensive survey and comparative study. *Preprint*, arXiv:2401.10825.

Muhammad Raza Khan, Morteza Ziyadi, and Mohamed AbdelHady. 2020. Mt-bioner: Multi-task learning for biomedical named entity recognition using deep bidirectional transformers. *Preprint*, arXiv:2001.08904.

J.-D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. 2003. Genia corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19:i180–i182.

Corinna Kolarik, Roman Klinger, Christoph M. Friedrich, Martin Hofmann-Apitius, and Juliane Fluck. 2008. Chemical Names: Terminological Resources and Corpora Annotation. In *Workshop on Building and evaluating resources for biomedical text mining (6th edition of the Language Resources and Evaluation Conference)*, pages 51–58, Marrakech, Morocco.

Martin Krallinger, Obdulia Rabal, Florian Leitner, Miguel Vazquez, David Salgado, Zhiyong Lu, Robert Leaman, Yanan Lu, Donghong Ji, Daniel M. Lowe, Roger A. Sayle, Riza Theresa Batista-Navarro, Rafal Rak, Torsten Huber, Tim Rocktäschel, Sérgio Matos, David Campos, Buzhou Tang, Hua Xu, Tsendsuren Munkhdalai, Keun Ho Ryu, S. V. Ramanan, Senthil Nathan, Slavko Žitnik, Marko Bajec, Lutz Weber, Matthias Irmer, Saber A. Akhondi, Jan A. Kors, Shuo Xu, Xin An, Utpal Kumar Sikdar, Asif Ekbal, Masaharu Yoshioka, Thaer M. Dieb, Miji Choi, Karin Verspoor, Madian Khabsa, C. Lee Giles, Hongfang Liu, Komandur Elayavilli Ravikumar, Andre Lamurias, Francisco M. Couto, Hong-Jie Dai, Richard Tzong-Han Tsai, Caglar Ata, Tolga Can, Anabel Usié, Rui Alves, Isabel Segura-Bedmar, Paloma Martínez,

Julen Oyarzabal, and Alfonso Valencia. 2015. The chemdner corpus of chemicals and drugs and its annotation principles. *Journal of Cheminformatics*, 7(1):S2.

Jiao Li, Yueping Sun, Robin J Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wiegers, and Zhiyong Lu. 2016. BioCreative V CDR task corpus: a resource for chemical disease relation extraction. *Database (Oxford)*, 2016.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2022. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):50–70.

Ling Luo, Po-Ting Lai, Chih-Hsuan Wei, Cecilia N Arighi, and Zhiyong Lu. 2022. Biored: a rich biomedical relation extraction dataset. *Briefings in Bioinformatics*, 23(5):bbac282.

Ling Luo, Chih-Hsuan Wei, Po-Ting Lai, Robert Leaman, Qingyu Chen, and Zhiyong Lu. 2023. AIONER: all-in-one scheme-based biomedical named entity recognition using deep learning. *Bioinformatics*, 39(5):btad310.

Zita Marinho, Afonso Mendes, Sebastião Miranda, and David Nogueira. 2019. Hierarchical nested named entity recognition. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 28–34, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Tahir Mehmood, Alfonso Gerevini, Alberto Lavelli, and Ivan Serina. 2019. Leveraging multi-task learning for biomedical named entity recognition. In *AI*IA 2019 – Advances in Artificial Intelligence*, pages 431–444, Cham. Springer International Publishing.

Sunil Mohan and Donghui Li. 2019. Medmentions: A large biomedical corpus annotated with umls concepts. *Preprint*, arXiv:1902.09476.

Vincenzo Moscato, Marco Postiglione, Carlo Sansone, and Giancarlo Sperlí. 2023. Taughtnet: Learning multi-task biomedical named entity recognition from single-task teachers. *IEEE Journal of Biomedical and Health Informatics*, 27(5):2512–2523.

Evangelos Pafilis, Sune P Frankild, Lucia Fanini, Sarah Faulwetter, Christina Pavloudi, Aikaterini Vasileiadou, Christos Arvanitidis, and Lars Juhl Jensen. 2013. The SPECIES and ORGANISMS resources for fast and accurate identification of taxonomic names in text. *PLoS One*, 8(6):e65390.

Yesol Park, Gyujin Son, and Mina Rho. 2024. Biomedical flat and nested named entity recognition: Methods, challenges, and advances. *Applied Sciences*, 14(20).

Abhishek Sharma, Amrita, Sudeshna Chakraborty, and Shivam Kumar. 2022. Named entity recognition in natural language processing: A systematic review. In *Proceedings of Second Doctoral Symposium on Computational Intelligence*, pages 817–828, Singapore. Springer Singapore.

Yongliang Shen, Xinyin Ma, Zeqi Tan, Shuai Zhang, Wen Wang, and Weiming Lu. 2021. 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)*, pages 2782–2794, Online. Association for Computational Linguistics.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. DiffusionNER: Boundary diffusion for named entity recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3875–3890, Toronto, Canada. Association for Computational Linguistics.

Yongliang Shen, Xiaobin Wang, Zeqi Tan, Guangwei Xu, Pengjun Xie, Fei Huang, Weiming Lu, and Yueting Zhuang. 2022. 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)*, pages 947–961, Dublin, Ireland. Association for Computational Linguistics.

Larry Smith, Lorraine K. Tanabe, Rie Johnson nee Ando, Cheng-Ju Kuo, I-Fang Chung, Chun-Nan Hsu, Yu-Shi Lin, Roman Klinger, Christoph M. Friedrich, Kuzman Ganchev, Manabu Torii, Hongfang Liu, Barry Haddow, Craig A. Struble, Richard J. Povinelli, Andreas Vlachos, William A. Baumgartner, Lawrence Hunter, Bob Carpenter, Richard Tzong-Han Tsai, Hong-Jie Dai, Feng Liu, Yifei Chen, Chengjie Sun, Sophia Katrenko, Pieter Adriaans, Christian Blaschke, Rafael Torres, Mariana Neves, Preslav Nakov, Anna Divoli, Manuel Maña-López, Jacinto Mata, and W. John Wilbur. 2008. Overview of biocreative ii gene mention recognition. *Genome Biology*, 9(2):S2.

Cong Sun, Zhihao Yang, Lei Wang, Yin Zhang, Hongfei Lin, and Jian Wang. 2021. Biomedical named entity recognition using bert in the machine reading comprehension framework. *Journal of Biomedical Informatics*, 118:103799.

Mario Sänger, Samuele Garda, Xing David Wang, Leon Weber-Genzel, Pia Droop, Benedikt Fuchs, Alan Akbik, and Ulf Leser. 2024. Hunflair2 in a cross-corpus evaluation of biomedical named entity recognition and normalization tools. *Bioinformatics*, 40(10):btae564.

Zeqi Tan, Yongliang Shen, Shuai Zhang, Weiming Lu, and Yueting Zhuang. 2021. A sequence-to-set network for nested named entity recognition. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3936–3942. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. 2021. Automated concatenation of embeddings for structured prediction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2643–2660, Online. Association for Computational Linguistics.

Xuan Wang, Yu Zhang, Xiang Ren, Yuhao Zhang, Marinka Zitnik, Jingbo Shang, Curtis Langlotz, and Jiawei Han. 2018. Cross-type biomedical named entity recognition with deep multi-task learning. *Bioinformatics*, 35(10):1745–1752.

Chih-Hsuan Wei, Alexis Allot, Kevin Riehle, Aleksandar Milosavljevic, and Zhiyong Lu. 2022. tmvar 3.0: an improved variant concept recognition and normalization tool. *Bioinformatics*, 38(18):4449–4451.

Chih-Hsuan Wei, Hung-Yu Kao, and Zhiyong Lu. 2015. GNormPlus: An integrative approach for tagging genes, gene families, and protein domains. *Biomed Res Int*, 2015:918710.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Hang Yan, Yu Sun, Xiaonan Li, and Xipeng Qiu. 2023. An embarrassingly easy but strong baseline for nested named entity recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1442–1452, Toronto, Canada. Association for Computational Linguistics.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2022. LinkBERT: Pretraining language models with document links. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8003–8016, Dublin, Ireland. Association for Computational Linguistics.

Yu Yin, Hyunjae Kim, Xiao Xiao, Chih Hsuan Wei, Jaewoo Kang, Zhiyong Lu, Hua Xu, Meng Fang, and Qingyu Chen. 2024. Augmenting biomedical named entity recognition with general-domain resources. *Journal of Biomedical Informatics*, 159:104731.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. Named entity recognition as dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6470–6476, Online. Association for Computational Linguistics.

Sheng Zhang, Hao Cheng, Jianfeng Gao, and Hoifung Poon. 2023. Optimizing bi-encoder for named entity recognition via contrastive learning. In *The Eleventh International Conference on Learning Representations*.

A Datasets and Experimental Setup

For the English subset of CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003), the original dataset splits are used, which are provided in a pre-tokenized format. For the GENIA dataset, the splits from Yan et al., 2023 are adopted. The entity counts per split of these datasets can be found in Table 7.

Dataset	Entity Type	Train	Dev	Test
CONLL	LOC	7,140	1,837	1,668
	MISC	3,438	922	702
	ORG	6,321	1,341	1,661
	PER	6,600	1,842	1,617
GENIA	Cell Line	3,069	372	403
	Cell Type	5,854	576	578
	DNA	7,707	1,161	1,132
	Gene or protein	28,874	2,466	2,900
	RNA	699	139	106

Table 7: Statistics for the datasets used in the single-task experiments of section 5.1.

To train a multi-task model, six biomedical datasets are utilized: BC2GM (Smith et al., 2008), BC4CHEMD (Krallinger et al., 2015), BC5CDR (Li et al., 2016), JNLPBA (Collier et al., 2004), Linnaeus (Gerner et al., 2010), and NCBI Disease (Doğan et al., 2014). The dataset splits (Table 8) follow those established by Crichton et al., 2017, which have been extensively used in prior studies, including Wang et al., 2018; Huang et al., 2019; Khan et al., 2020; Moscato et al., 2023.

Dataset	Entity Type	Train	Dev	Test
BC2GM	Gene or protein	15,035	3,032	6,243
BC4CHEMD	Chemical	29,263	29,305	25,210
BC5CDR	Chemical	5,114	5,239	5,277
	Disease	4,169	4,224	4,394
JNLPBA	Cell Line	3,369	389	490
	Cell Type	6,162	522	1,906
	DNA	8,416	1,040	1,045
	Gene or protein	27,015	2,379	4,988
	RNA	844	106	118
Linnaeus	Species	2,079	700	1,412
NCBI Disease	Disease	5,111	779	952

Table 8: Statistics for the datasets used in the multi-task experiment of section 5.2.

In the aforementioned experiments, models are trained on the respective training splits, checkpoint selection is made on the development splits, and evaluation is conducted on the test splits.

For the cross-corpus evaluation, the experimental setup of Sänger et al., 2024 is replicated. A multi-task model is trained using an ensemble of nine datasets[9]: BioRED (Luo et al., 2022), GNormPlus (Wei et al., 2015), Linnaeus (Gerner et al., 2010), NCBI Disease (Doğan et al., 2014), NLM-Chem (Islamaj et al., 2021a), NLM-Gene (Islamaj et al., 2021b), S800 (Pafilis et al., 2013), SCAI Chemical (Kolarik et al., 2008), and SCAI Disease (Gurulingappa et al., 2010). The model is trained on the training sets, with checkpoint selection being performed on the development splits. The evaluation is conducted on an independent corpus consisting of the full annotated data of three datasets[10]: BioID (Arighi et al., 2017), MedMentions (Mohan and Li, 2019), and tmVar3 (Wei et al., 2022). Dataset statistics for the training corpora and the independent test corpora can be found in Table 9 and Table 10, respectively.

Dataset	Entity Type	Train	Dev	Test
BioRED	Cell Line	103	22	50
	Chemical	2,830	818	751
	Disease	3,643	982	917
	Gene	4,404	1,087	1,170
	Species	1,429	370	393
GNormPlus	Gene	4,964	504	4,468
Linneaus	Species	1,725	206	793
NCBI Disease	Disease	4,083	666	2,109
NLM-Chem	Chemical	21,102	5,223	11,571
NLM-Gene	Gene	11,209	1,314	2,687
S800	Species	2,236	410	1,079
SCAI Chemical	Chemical	852	83	375
SCAI Disease	Disease	1,281	250	710

Table 9: Statistics of the training corpora used in the cross-corpus evaluation scenario of section 5.3.

Dataset	Entity Type	Number of mentions
BioID	Species	7,939
tmVar3	Gene	4,059
MedMentions	Disease	19,298
	Chemical	19,198

Table 10: Statistics of the corpora used for the cross-corpus evaluation described in section 5.3.

Finally, in order to assess the model's global prediction ability, synthetic datasets were derived from the BC5CDR corpus, in line with (Huang et al.,

[9]The datasets were obtained in February 2025 from https://github.com/flairnlp/flair. Their splits and preprocessing choices were replicated.

[10]The preprocessed datasets were downloaded from https://github.com/hu-ner/hunflair2-experiments in February 2025.

2019) experimental setup. The original training set was randomly divided into two disjoint subsets: BC5-Disease (containing only Disease annotations) and BC5-Chemical (containing only Chemical annotations). The same procedure was followed for the development splits. The statistics of these synthetic datasets can be found in Table 11. By training models on the BC5-Disease and BC5-Chemical subsets and evaluating them on the full test split of the BC5CDR corpus, we can test the models global prediction abilities, as described in section 5.4.

Dataset	Entity Type	Train	Dev
BC5-Disease	Disease	2,172	2,279
BC5-Chemical	Chemical	2,459	2,665

Table 11: Statistics of the synthetic datasets created for assessing global prediction ability.

B Training Details

Hyperparameter	GENIA	Others
# epochs	100	100
Early stop	30	30
Batch size	16	16
Max. # tokens	405	405
Gradient norm clipping	1.0	1.0
Dropout on logits	0.1	0.1
SRU module		
# latent embeddings (multiplier)	10	2
Half-context for pos. embeddings	240	150
Dropout on pos. embeddings	0.2	0.2
Dropout on latent embeddings	0.2	0.2
Encoder optimizer		
LR	3e-5	2e-5
Weight decay	1e-3	1e-3
Warm up (in epochs)	1	1
Actions generation cycle optimizer		
LR	3e-4	3e-4
Weight decay	1e-3	1e-3
Warm up (in epochs)	0.5	0.5

Table 12: Hyperparameters used for the experiments. The column 'Others' refers to every experiment except the single-task on the GENIA dataset.

All models are developed using the PyTorch tensor library and trained on a single NVIDIA A100 80GB GPU. The encoder module and the action generation module are tuned using two separate AdamW optimizers with linear warm-up, set with different initial learning rates and weight decays. Both optimizers are set with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-6}$. Models are trained with early stopping based on performance on the development set.[11] The hyperparameters of all experiments can be found in Table 12. Additionally, while the token scaling parameter α in equation (5) of section 3.3 was trained for the single-task experiment on the GENIA dataset, it was frozen and set to 1 for all other experiments.

The encoder module was built on top of the HuggingFace *transformers* library (Wolf et al., 2020). Specifically, the `LinkBERT-large` encoder from Yasunaga et al., 2022 was used for all models trained with biomedical corpora, while the `xlm-roberta-large` encoder introduced by Conneau et al., 2020 was used for the single task model trained on the CoNLL-2003 dataset.

C Single-task performance on the datasets used for the cross-corpus experiment

Dataset	*Merged*	*Disjoint*
BioRED	90.73	90.90
GNormPlus	85.00	86.00
Linnaeus	78.16	92.23
NCBI Disease	85.69	85.70
NLM-Chem	84.42	85.65
NLM-Gene	88.35	88.13
S800	74.24	75.79
SCAI Chemical	85.21	85.64
SCAI Disease	80.78	82.14

Table 13: In-corpus micro-F1 scores for the model used in the cross-corpus evaluation experiment of section 5.3.

D Human evaluation of global predictions in a cross-domain setting

To assess the model's ability to generalize across domains, three models were trained:

- *SRU-NER-CoNLL*: a single-task model trained on the CoNLL corpus;
- *SRU-NER-BC5*: a single-task model trained on the BC5CDR corpus;
- *SRU-NER-MTL*: a multi-task model trained on both corpora.

All models were trained using the `LinkBERT-large` encoder from Yasunaga

[11] In the case of multi-task models where multiple datasets are tagged with the same entity type (the models of sections 5.2 and 5.3), despite the entity types being considered disjoint for training purposes, validation F1 scores on the development set for checkpoint selection are computed by merging the types, as described in the begining of section 5.2.

et al., 2022. To evaluate cross-domain generalization, the models capable of recognizing general-domain entity types (*SRU-NER-CoNLL* and *SRU-NER-MTL*) were used to annotate the test split of the biomedical corpus, while the models trained on biomedical entity types (*SRU-NER-BC5* and *SRU-NER-MTL*) were used to annotate the test split of the general-domain corpus. Since gold annotations for these out-of-domain predictions were not available, two linguists manually assessed their correctness. Inter-annotator agreement per entity type is reported in Table 14.

Entity	**Agreement** (%)
Chemical	92.98
Disease	91.09
LOC	100.00
ORG	87.76
PER	88.89

Table 14: Inter-annotator agreement for the evaluated entity types.

Based on the assessment of correct predicted spans by the two human annotators, a precision score was computed by taking the ratio of correctly identified spans to the total number of predicted spans, for each model, entity type and linguist. A simulated recall score per model, entity type and linguist was also computed by considering the total number of spans of each entity type that were considered correct by at least one of the annotators, across all the predictions made by the three models. Finally, precision and simulated recall scores per model and entity type were obtained by averaging across the two human annotators.

The results can be found in Table 6, in the main text. One can see that the precision scores of the multi-task model are higher than the single-task ones across all entity types, while the recall values of the multi-task model are worse for all entity types except ORG.

For reference, the in-corpus performance of the three models is present in Table 15.

Model	**CoNLL**	**BC5CDR**
SRU-NER-CoNLL	90.51	—
SRU-NER-BC5	—	90.61
SRU-NER-MT	91.01	90.51

Table 15: In-corpus performance of the three models used for evaluation of global predictions in a cross-domain setting. The single-task model SRU-NER-BC5 is the same as the one used for comparison in the multi-task experiment of section 5.2.