Simple yet Powerful: An Overlooked Architecture for Nested Named Entity Recognition

Anonymous ACL submission

Abstract

Named Entity Recognition (NER) is an important task in Natural Language Processing that aims to identify text spans belonging to predefined categories. Traditional NER research ignores nested entities, which are entities contained in other entity mentions. Although several methods have been proposed to address this case, most of them rely on complex task-specific structures and ignore potentially useful baselines for the task. We argue that this creates an overly optimistic impression of their performance. This paper revisits the Multiple LSTM-CRF (MLC) model, a simple, overlooked, yet powerful approach based on training independent sequence labeling models for each entity type. Extensive experiments with three nested NER corpora show that, regardless of the simplicity of this model, its performance is better or at least as well as more sophisticated methods. Furthermore, we show that the MLC architecture achieves state-of-the-art results in the Chilean Waiting List corpus by including pre-trained language models. In addition, we propose new task-specific metrics that adequately measure the ability of models to detect nestings. The results show that standard NER metrics do not measure well the ability of a model to detect nested entities, while our task-specific metrics provide new evidence on how existing approaches handle the task.

1 Introduction

Named Entity Recognition (NER) is a widely studied task in Natural Language Processing (NLP) that seeks to identify text spans expressing references to predefined categories such as person names, locations, and organizations (Chinchor and Robinson, 1997). NER, or in general the task of recognizing entity mentions[1], has drawn the attention of the community due to its relevance in several NLP applications. Nested Named Entity Recognition is a particular case of NER where entities are nested within each other (Finkel and Manning, 2009). Traditional NER models simplify the nested entities by keeping the outermost entity and removing the inner ones. This simplified problem is better known as flat NER and is commonly regarded as a sequence labeling problem since each token can be associated with at most one label. However, removing part of these entities could be a problem in model performance due to the loss of relevant information and inner dependencies.

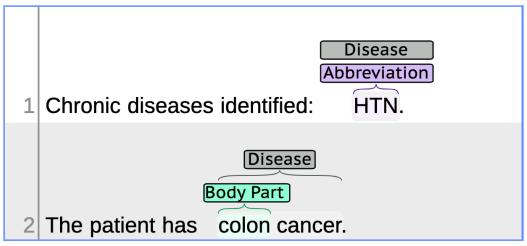

Figure 1: An example of a multi-label entity in the Chilean Waiting List corpus, followed by a nesting of different types. The annotation was translated from its original language.

Several methods have been proposed to address the nesting problem. Traditional approaches have focused on creating representations of nested entities through structures such as hypergraphs (Lu and Roth, 2015; Muis and Lu, 2017; Katiyar and Cardie, 2018; Wang and Lu, 2018). However, they usually suffer from heavy feature engineering, structural ambiguity, or complex models. Another category is region-based, which divides the problem into two sequential stages. First, the detection of entity boundaries, and then the assignment of entity types to these regions (Sohrab and Miwa, 2018; Zheng et al., 2019; Yu et al., 2020). One of the main drawbacks of this method is its high time complexity. There are also approaches that attempt to transform the nested NER task into a sequence labeling prob-

[1]Mentions are defined as references to entities that could be named, nominal or pronominal (Florian et al., 2004).

lem (Alex et al., 2007; Ju et al., 2018; Shibuya and Hovy, 2020). Although these studies have shown competitive performance, we realized that most of them have three critical issues, discussed below.

First, with the incorporation of large pre-trained language models, the standard LSTM-CRF (Lample et al., 2016) sequence labeling architecture received substantial improvements for flat NER tasks (Liu et al., 2017). However, little research has been conducted on adapting this architecture to nested NER using a single entity approach, i.e., training independent flat NER models for each entity type. In this paper, we revisit this architecture, naming it Multiple LSTM-CRF (MLC). Despite the apparent simplicity, we show that this model yields very positive results, outperforming several recent approaches explicitly designed for nested entities.

Second, we note that most of the literature ignores the case in which the same text span is tagged with more than one entity type, as shown in Figure 1. This case is very common in the Chilean Waiting List corpus (Báez et al., 2020), and it was first noticed by Alex et al. (2007), but was not analyzed further. One of the main advantages of our architecture is that it addresses this problem.

Third, we argue that the way the community is evaluating this task does not adequately measure the effectiveness of a model at identifying nested entities. Specifically, the current metric calculates the micro F1-score over all entities, which is the same metric used in flat NER. Consequently, a model that performs well over flat entities, but not nested ones, may also obtain good results. To alleviate this problem, we first identify the different types of nesting by formalizing the task and then proposing new task-specific metrics for these cases.

In summary, the main contributions of our work are the following:

- Due to the lack of a consensual definition of nested NER, we introduce a formalization of the task by identifying the different types of nesting, and we also propose new task-specific evaluation metrics to measure performance on nesting.

- We conduct an empirical study comparing several nested NER architectures in three datasets from different languages, with particular attention to the impact of using pre-trained language models and nesting metrics. Experimental results confirm the effectiveness of the MLC model, achieving state-of-the-art in the Chilean Waiting List corpus and competitive performance in the rest of the corpora.

2 Related Work

In recent years there has been a growing interest from the research community in nested NER. Several studies have been conducted to address nested entities, which can be mainly divided into three categories:

Region-based: These approaches divide the problem into two stages: identifying entity boundaries and then categorizing these regions. Sohrab and Miwa (2018) designed a model that enumerates all possible spans within a limited length and then used boundary and average internal token representation to predict entity types. Another region-based model was proposed by Zheng et al. (2019), which uses a sequence labeling layer to detect entity boundaries, and then classified selected regions into their categorical types. Yu et al. (2020) used ideas from a biaffine model, scoring all possible start-end tokens in a sentence to predict nested entities. Although these methods have proven to be effective, they often suffer from high time complexity and fail to identify entities tagged with more than one type.

Structure-based: There have also been attempts to capture the structure of nested entities. Finkel and Manning (2009) represented each input sentence as a constituency tree of nested entities and used a CRF-based approach to predict entity types. Lu and Roth (2015) proposed a mention hypergraph representation to extract entity mentions. Next, Muis and Lu (2017) improved on previous work by modeling nested NER with mention separators and handcrafted features. Similarly, Katiyar and Cardie (2018) designed a directed hypergraph using LSTM features to learn the nesting structure. Wang et al. (2020) recursively introduce the embedding of tokens and regions into flat NER layers simulating the shape of a pyramid. However, these approaches usually suffer from spurious structures and structural ambiguities, as explained in Wang and Lu (2018).

Sequence labeling-based: Some studies report that sequence labeling methods can also perform well on this task. Early work mainly exploited the potential of conditional random fields (CRF). Alex et al. (2007) proposed three CRF-based methods to reduce the nested NER as several BIO tagging problems. Their best approach, called cascaded CRF, uses one model per entity type using the output of

the previous flat NER model as a feature for the current one. Ju et al. (2018) took advantage of inner entity information to encourage outer entity recognition. They dynamically stacked LSTM-CRF layers predicting entities in an inside-to-outside way until no entities were extracted. Straková et al. (2019) formulated the nested NER task as a sequence-to-sequence problem using an LSTM to decode entity types. Finally, Shibuya and Hovy (2020) recognized entities iteratively from outermost ones to inner ones using a recursive CRF-based method. The MLC approach falls into this category by using a sequence labeling approach capable of handling both nested entities and entities tagged with more than one label.

3 Methods

3.1 Problem Definition

One of the main issues in our knowledge of nested NER is that the task definition has not been addressed in-depth, and clarification of the different nesting cases is needed. By analyzing several corpora with nested entities, we have identified the following nesting cases:

Multi-label entities (ME): This case has been little explored in the literature. As explained in Alex et al. (2007), it consists of entities tagged with more than one entity type. With the release of the Chilean Waiting List corpus, it is interesting to study this case since 10.75% of the entities are involved in this type of nesting. For example, the entity “HTN", which stands for hypertension, is tagged as a disease and an abbreviation.

Nested entities of different types (NDT): This is the most frequent type of nesting in nested NER datasets. It consists of an entity containing a shorter entity tagged with a different type. An example is “colon cancer", where a body part (colon) is contained in a disease.

Nested entities of the same type (NST): This case usually occurs when entities are originally represented by a hierarchy, which is later pruned to reduce the entity space, resulting in the merging of entities of different levels of granularity. Although it appears in most corpora, it is much more frequent in GENIA (Kim et al., 2003). For example, the DNA “Drosophila homeodomain" contains another DNA, “homeodomain".

To better understand these cases, we formally define what we mean by nested entities and the nested NER task.

Definition 1 (Nested entities) *Given an input sequence $X = \{x_1, x_2, ..., x_n\}$ of words, an entity Q is defined by a tuple (S_q, E_q, T_q), where S_q and E_q $\in [1, n]$ represents entity boundaries in X, and T_q in $\mathcal{E}$ (the entity space) corresponds to entity type. Given two entities Q and R, we say that Q is nested in R if $S_r \leq S_q$ and $E_q \leq E_r$. The particular case of $S_q = S_r$ and $E_q = E_r$ corresponds to an entity with multiple labels. Note that under this definition we consider the three types of nesting.*

Definition 2 (Nested NER) *Given an input sequence $X = \{x_1, x_2, ..., x_n\}$, nested NER aims to correctly identify the boundaries for every entity Q in X and assign it the correct entity type from a predefined list of categories. This identification must be made for cases where nested entities are involved and when not.*

3.2 Model

In recent years, with advances in deep learning, sequence labeling architectures have received substantial improvements in the NER task. Therefore, we decided to revisit a method that belongs to this category but, despite its effectiveness, has been little studied.

Multiple LSTM-CRF (MLC): This approach consists of training multiple flat NER models, one for each entity type. The predicted labels of the input sentences correspond to the union of the outputs of each of these models, thus retrieving both nested entities and entities tagged with multiple labels. The main advantage of this approach is that it can easily incorporate all the progress made for the flat NER task into the nested NER task.

The apparent simplicity of MLC would lead us to believe that it should be considered as a natural baseline for any proposed architecture in nested NER. However, we realized that the few papers that have incorporated this model had used it as a baseline (Muis and Lu, 2017; Lin et al., 2019; Fei et al., 2020), but their reported results are not competitive. We believe that the problem lies in the fact that they do not use the potential of recent advances in flat NER architectures, such as the addition of domain-specific embeddings or pre-trained language models. These are the elements that we will incorporate in our work to show the effectiveness of the model.

Figure 2 shows an overview of the MLC model. Specifically, to create each flat NER module, we follow the LSTM-CRF approach proposed by Lam-

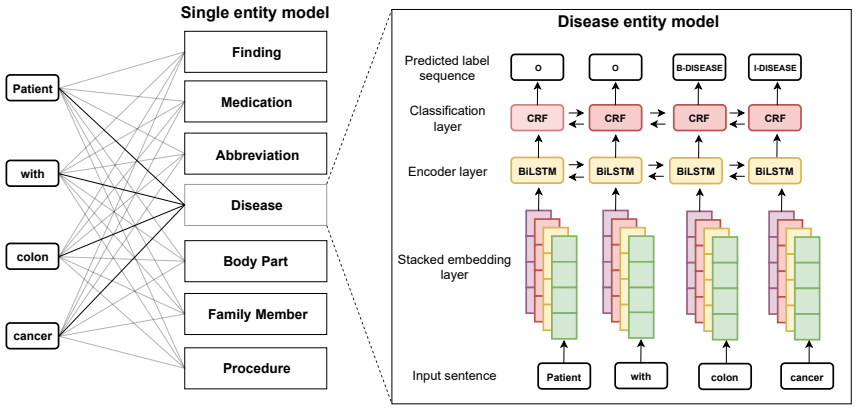

Figure 2: Overview of the MLC architecture, where each entity type has an associated flat NER model. The right side of the figure shows, as an example, the flat NER module for the Disease tag in the Chilean Waiting List dataset.

ple et al. (2016), one of the most widely used architectures for sequence labeling. To encode sentences, we use different combinations of embeddings in the stacked embedding layer. First, we concatenate domain-specific word embeddings with character embeddings retrieved from a bidirectional character-level LSTM. Next, we enrich word representations by adding contextualized embeddings from Flair (Akbik et al., 2018) and BERT (Devlin et al., 2019a), which have proven to be particularly effective on NER. The output is fed into a BiLSTM encoding layer to obtain long-contextual information. Finally, we use a CRF-loss and the Viterbi algorithm to decode the most likely tag sequence using the IOB2 tagging format.

4 Experiments

In this section, we present the datasets, baselines, and settings used in our experiments.

4.1 Datasets

Since most previous work on nested NER has been done in English datasets, we conducted our experiments with three corpora containing nested entities for three different languages and domains. The statistics for each corpus are shown in Table 1.

GENIA V3.02[2] **(Kim et al., 2003)** English biomedical corpus created from 2,000 MEDLINE abstracts. It is composed of 36 fine-grained entity types and 55,740 entity mentions, of which 17.3% are involved in nesting. We followed the same setup as the previous work (Finkel and Manning, 2009; Lu and Roth, 2015; Zheng et al., 2019), collapsing sub-types into their five super-types, using the first 90% of the sentences for the training set and the remaining 10% in the test set.

GermEval 2014[3] **(Benikova et al., 2014)** German dataset sampled from German Wikipedia and German online news. It consists of a total of 41,124 entity mentions, where 14.9% of them are involved in nesting. The corpus contains two levels of nesting and 12 entity types.

Chilean Waiting List[4] **(Báez et al., 2020)** Spanish clinical corpus created from real diagnoses of the Chilean healthcare system. It is composed of 43,730 entity mentions and seven entity types. From a nested NER point of view, it is a good resource since 46.7% of the entities are involved in nesting. In addition, to date, there are no reported results on nested NER in this dataset.

Studying previous work, we have noticed that comparisons between models are not entirely fair since the data partitions used vary between different papers. Therefore, for a fair comparison, in both the GENIA and GermEval datasets, we trained the models using the preprocessed version released in Zheng et al. (2019). In the case of the Chilean corpus, we used the public files released by the authors, which are already tokenized.

4.2 Baselines

We compare our results with several state-of-the-art models in GENIA and GermEval. Table 2 shows the different types of nesting that each of these baselines is capable of addressing. Based on the released source code, we have reproduced the following models to use as a reference for analyzing both traditional and task-specific metrics:

[2] http://www.geniaproject.org/genia-corpus/pos-annotation

[3] https://sites.google.com/site/germeval2014ner/data

[4] https://zenodo.org/record/5591011

	GENIA			GermEval			Chilean Waiting List		
	Train	Test	Dev	Train	Test	Dev	Train	Test	Dev
tokens	454,882	57,021	48,932	452,853	96,499	41,653	149,574	18,436	16,754
sentences	15,023	1,854	1,669	24,000	5,100	2,200	8,014	990	890
avg sent len	30.3	30.8	29.3	18.9	18.9	18.9	18.7	18.6	18.8
entities	45,929	5,474	4,337	31,545	6,693	2,886	35,480	4,289	3,971
avg entity len	2.9	2.9	3.1	1.4	1.4	1.5	2.6	2.7	2.6
nested entities (%)	17.0	20.6	16.8	15.0	14.7	14.1	46.4	45.9	46.7
nested entities	7,795	1,130	727	4,721	986	407	16,456	1,969	1,856
- different type	3,712	589	369	4,230	892	366	12,635	1,555	1,398
- same type	4,132	547	358	536	93	44	0	0	0
- multi-label entities	0	0	0	2	2	0	4,241	470	502

Table 1: Statistics of the datasets.

Pyramid is a structure-based architecture that recognizes entities in a bottom-up manner, from the shortest to the longest, assimilating the shape of a pyramid. It is currently the state-of-the-art method without using external supervision (Wang et al., 2020).

Recursive-CRF is a sequence labeling-based approach that extracts nested entities iteratively in an outside-to-inside way using a recursive CRF-based algorithm (Shibuya and Hovy, 2020).

Layered is a sequence labeling-based model designed to identify nested entities by dynamically stacking LSTM-CRF layers. It predicts entities in an inside-to-outside way until no more entities are extracted. (Ju et al., 2018).

Exhaustive is a region-based model that enumerates all possible regions as potential entity mentions, and then classifies them into their entity types (Sohrab and Miwa, 2018).

Boundary is a region-based method that combines ideas from the Layered and Exhaustive models. It uses a BiLSTM layer to detect boundary-relevant regions and then uses these representations to predict categorical entity labels (Zheng et al., 2019).

Biaffine is a region-based architecture that leverages contextualized paragraph-level embeddings to create a Biaffine model. This approach scores candidate pairs of start and end tokens in a sequence and then classifies them into predefined categories using nested entities constraints (Yu et al., 2020).

4.3 Implementation Details

Pre-trained Word Embeddings. To encode sentences, we selected pre-trained word embeddings in the same domain of each corpus. For the experiments with GENIA, we used biomedical embeddings trained on MEDLINE abstracts (Chiu et al., 2016). In GermEval, we incorporated German FastText embeddings (Grave et al., 2018), and for the Chilean dataset, we used pre-trained embeddings from a large clinical corpus, which can be downloaded from here[5]. During the training process, the embeddings were not left static.

Model	ME	NDT	NST
Layered	✓	✓	✓
Exhaustive	✗	✓	✓
Boundary	✗	✓	✓
Biaffine	✗	✓	✓
Pyramid	✓	✓	✓
Recursive-CRF	✓	✓	✓
MLC	✓	✓	✗

Table 2: Nesting types identified by the architectures used in our experiments. Multi-label entities (ME), nesting of different types (NDT), and nesting of the same type (NST).

Contextual Word Embeddings. To study the impact of adding pre-trained language models, we used BERT (Devlin et al., 2019b), and Flair (Akbik et al., 2018), which is a character-level language model. In the case of BERT, since it uses WordPiece tokenization, we computed word embeddings using the average of subtoken embeddings. A version of these models was available for all the languages and domains involved in our study, except for Spanish. Therefore, we added new language models in the Spanish clinical domain to the Flair framework. We trained these models on the same corpus as the word embeddings used for the Chilean dataset, following the same settings and assumptions reported in the Flair paper.

Regarding the Biaffine model, the BERT embeddings were created using the paragraph-level context. However, Fu et al. (2020) explains that this method provides better performance in resolving correlations, so it is not an entirely fair comparison with models that use sentence-level context. For

[5]http://doi.org/10.5281/zenodo.3924799

Parameter	Range	MLC
max epochs	[20, 100]	100
optimizer	[SGD, Adam, AdamW]	SGD
batch size	[8, 32]	16
learning rate	[0.0001, 0.1]	0.1
char emb dim	[20, 50]	25
dropout	[0.2, 0.8]	0.3
BiLSTM depth	[1, 3]	3
BiLSTM hidden size	[128, 512]	128

Table 3: Hyperparameter search space and the best values found for the MLC model.

this reason, we do not make a comprehensive comparison with this model in terms of contextualized embeddings.

Parameters. We used a unified setting for all the experiments with MLC. The best hyperparameters were chosen by performing a random search over the range of values shown in Table 3, selecting the best configuration based on performance on the development set. To perform a fair comparison with baselines, we used the best hyperparameters reported in their papers.

We trained the MLC architecture using the SGD optimizer to a maximum of 100 epochs, with mini-batches of size 16 and a learning rate of 0.1. To control the overfitting problem, we employed a learning rate scheduler and an early stopping strategy. We also applied dropout regularization (Srivastava et al., 2014) after the embedding layer and BiLSTM. The MLC model was implemented using the Flair framework (Akbik et al., 2019), and the rest of the baselines were executed with the official code provided by the authors. All the experiments were performed using a Tesla V100 GPU.

4.4 Evaluation Metrics

Overall Performance. Performance was evaluated using precision, recall, and micro F1-score, which is the standard metric used in nested NER. An entity is considered correct when both entity types and boundaries are predicted correctly.

Nested Performance. Since flat entities are much more common than nested entities, the above metric ends up confusing flat and nested results and, consequently, is not able to reflect well the ability of a model to detect nesting. To alleviate this issue, we analyze task-specific metrics proposed in previous work that adequately measure the model's ability to detect nested and non-nested entities. Precisely, we compute scores for the following cases: non-nested entities (m_{flat}), nested entities (m_{nested}), inner entities (m_{inner}) and outer entities (m_{outer}). We consider an entity to be nested if it contains any entity or is contained within another entity. Thus, the m_{nested} metric considers both m_{inner} and m_{outer} scores.

However, none of these existing metrics capture the ability of the models to recognize both inner and outer entities simultaneously. For this reason, and to demonstrate whether the choice of a model in a dataset depends on the types of nesting present, we compute a score for nesting ($m_{nesting}$) and on the different types of nesting described in the task formalization (m_{ME}, m_{NDT}, m_{NST}). A nesting is considered correct if both inner and outer entities are recognized correctly.

The above metrics are calculated using precision, recall, and micro F1-score, but we only report the last one for brevity. We emphasize that most of these metrics have not been used before in nested NER research. Therefore, we believe it is crucial to incorporate them in future work as it allows us to measure and differentiate the performance of models on nested and non-nested entities.

4.5 Main Results

Table 4 shows the overall performance of the proposed model against baselines on three different datasets. Despite its simplicity, we observe that the MLC architecture outperforms existing state-of-the-art models on the Chilean Waiting List by +1.6% in terms of the F1 measure. By contrast, although state-of-the-art is not obtained in GENIA and GermEval, we can see that MLC outperforms many specialized nested NER architectures, thus being a competitive approach. One possible reason for the excellent performance is that we use one model per entity type, which means that the number of possible labels is only one per model, avoiding the problem of nested entities and making the classification step more straightforward compared to other architectures. Compared with the statistics in Table 1, we can conclude that it is more challenging to obtain good results when the corpora have entities of a more considerable length. This can be explained by the strict metric we are using, where the boundaries and the entity types are requested to match.

We further analyze the effect of adding pre-trained language models in our experiments. As we believed, all models benefit from incorporating contextual word embeddings, improving their performance considerably compared to their base

	GENIA			GermEval			Chilean Waiting List		
Model	P	R	F1	P	R	F1	P	R	F1
Layered	73.9	68.7	71.2	71.8	64.1	67.7	75.0	72.8	73.9
Exhaustive	74.1	69.7	71.8	78.6	64.6	70.9	76.3	71.7	68.2
Boundary	76.7	71.8	74.2	74.4	65.5	69.7	74.0	67.6	70.7
Pyramid	78.1	72.8	75.3	77.8	66.9	71.9	79.6	75.4	77.5
Biaffine	79.1	73.7	76.3	89.0	77.4	82.8	81.5	67.1	73.6
Recursive-CRF	75.8	75.2	75.5	85.1	78.2	81.5	75.1	77.2	76.1
MLC	77.6	74.2	75.8	86.8	77.2	81.7	77.7	78.3	78.0
LM-based									
Biaffine [BERT]	79.9	76.5	**78.1**	88.3	85.0	86.6	78.7	70.8	74.5
Recursive-CRF									
- Flair	77.1	78.0	77.6	83.4	82.9	83.2	78.0	79.9	78.9
- BERT	76.4	77.4	76.9	84.3	83.0	83.6	76.6	77.8	77.2
- Flair + BERT	77.4	76.8	77.1	84.8	82.1	83.4	77.1	77.9	77.5
Pyramid									
- Flair	77.8	75.6	76.7	83.4	80.0	81.7	80.1	77.2	78.6
- BERT	79.1	76.9	78.0	87.7	85.8	**86.7**	78.0	73.6	75.7
- Flair + BERT	80.4	75.0	77.6	87.7	84.4	86.0	78.5	77.2	77.9
MLC									
- Flair	80.1	75.2	77.6	85.3	82.4	83.8	80.6	80.5	**80.5**
- BERT	79.4	74.3	76.8	85.1	80.3	82.6	79.7	78.8	79.3
- Flair + BERT	78.8	75.2	75.5	84.7	80.1	82.3	79.9	78.1	79.0

Table 4: Overall results on three nested NER datasets.

version. In GermEval, a general-purpose corpus, the language model that best improves the model's performance is BERT, while in the other corpora, it is Flair. Also, we can see that stacking Flair and BERT embeddings does not produce better results. We attribute this to the high dimensionality of these representations and to the fact that the two language models were trained on different corpora.

Regarding the Chilean corpus, which contains the highest percentage of nested entities, we observe that the MLC model with Flair embeddings improves by +2.5% compared to its base version without pre-trained language models. This demonstrates the effectiveness of using Flair over BERT in this corpus. We suspect that it is due to the large number of misspelled and out-of-vocabulary words found in the unstructured clinical text. As pointed out in Akbik et al. (2018), handling these types of words is one of the main advantages when using its character-level language model.

Despite the promising results, we hypothesize that benchmarking against the standard nested NER metric may not be a good indicator of model performance on nesting since most of the entities are not nested. Therefore, we analyze the results using nested metrics.

4.6 Nested Results

In most cases, the revisited nested metrics presented in Table 5 are relatively consistent with results in Table 4. This means that models which obtain state-of-the-art using the standard metrics also perform well according to these metrics. For example, in the Chilean Waiting List, the best model (MLC) achieves the best results according to the m_{flat}, m_{inner}, m_{outer}, m_{nested} metrics, which is a remarkable result considering the large number of nestings present in this corpus. Another observation is that, unlike the other datasets, in GENIA is more complex to recognize inner entities over the outermost ones. This finding could be helpful when designing future architectures for this corpus.

As expected, the models with better performance according to the standard metric are also associated with good results using the m_{flat} metric. This may not be a good indicator in the nested NER task since most of the entities in these corpora are not nested, and the proper performance on nestings is not reflected. This issue becomes much more evident when analyzing our proposed nesting metrics, presented in Table 6. We observe that the results are significantly lower than those for the previous metrics of Tables 4 and 5. This reveals the difficulty of correctly recognizing the nesting cases. One possible reason for this low performance is that these metrics are strict, as inner and outer entities must be correctly predicted.

Although the selected baselines are designed to deal with nestings of the same type, their m_{NST} results in GENIA and GermEval are poor, while the results using the m_{NDT} metric are much higher. This suggests that NST is the most difficult case to

GENIA				
Model	m_{flat}	m_{nested}	m_{inner}	m_{outer}
Layered	73.2	62.3	42.9	79.8
Exhaustive	76.6	55.0	42.6	67.9
Boundary	77.4	59.5	42.0	75.6
Biaffine [BERT]	81.2	**65.8**	**49.3**	80.5
Pyramid [BERT]	81.1	65.2	46.1	**82.4**
Recursive-CRF [Flair]	**81.5**	62.3	46.9	77.4
MLC [Flair]	80.7	63.8	41.7	82.2

GermEval				
Model	m_{flat}	m_{nested}	m_{inner}	m_{outer}
Layered	68.8	60.9	62.0	59.7
Exhaustive	73.4	56.1	65.7	45.7
Boundary	70.9	54.5	54.1	55.0
Biaffine [BERT]	88.4	76.6	**78.1**	75.0
Pyramid [BERT]	**88.5**	**76.7**	77.3	**76.1**
Recursive-CRF [BERT]	85.5	73.0	74.9	71.0
MLC [Flair]	86.0	71.6	74.5	68.4

Chilean Waiting List				
Model	m_{flat}	m_{nested}	m_{inner}	m_{outer}
Layered	73.4	74.5	82.4	64.5
Exhaustive	71.7	63.8	71.5	53.4
Boundary	73.4	61.1	65.5	55.4
Biaffine [BERT]	76.2	72.5	75.2	69.2
Pyramid [Flair]	79.0	78.1	84.7	69.3
Recursive-CRF [Flair]	80.3	77.4	82.8	70.4
MLC [Flair]	**80.9**	**80.1**	**86.2**	**72.5**

Table 5: Results on nested and non-nested entities.

GENIA				
Model	$m_{nesting}$	m_{ME}	m_{NDT}	m_{NST}
Layered	26.2	-	41.7	9.7
Exhaustive	25.8	-	41.2	17.7
Boundary	26.6	-	40.5	17.8
Biaffine [BERT]	**34.5**	-	**51.9**	**22.9**
Pyramid [BERT]	33.4	-	49.5	20.9
Recursive-CRF [Flair]	31.5	-	49.1	19.4
MLC [Flair]	27.9	-	47.8	0

GermEval				
Model	$m_{nesting}$	m_{ME}	m_{NDT}	m_{NST}
Layered	37.3	-	40.4	16.2
Exhaustive	27.8	-	38.2	9.7
Boundary	21.2	-	25.5	7.8
Biaffine [BERT]	55.7	-	**64.3**	20.8
Pyramid [BERT]	**56.5**	-	63.8	21.4
Recursive-CRF [BERT]	51.1	-	58.9	**23.9**
MLC [Flair]	49.1	-	59.3	0

Chilean Waiting List				
Model	$m_{nesting}$	m_{ME}	m_{NDT}	m_{NST}
Layered	51.6	71.1	49.5	-
Exhaustive	28.4	0	41.7	-
Boundary	28.2	0	35.4	-
Biaffine [BERT]	41.8	0	55.1	-
Pyramid [Flair]	54.9	**73.7**	57.9	-
Recursive-CRF [Flair]	56.0	71.7	58.8	-
MLC [Flair]	**60.6**	72.5	**60.0**	-

Table 6: Our task-specific metrics. If columns have no results, it means that there was not a significant number of examples.

identify for all models. Therefore, we believe that a model should not be prematurely discarded based on its limitation to handle a particular type of nesting. For example, although the MLC architecture cannot strictly identify the NST case in GENIA and GermEval, it obtains excellent results on the NDT case and the outermost entities involved in the NST. In contrast, concerning the m_{ME} metric, we note that the performance of the four models addressing this case is quite good, suggesting that it is not a complex case to recognize but still not taken into account when building nested NER models.

Finally, we highlight that in the Chilean corpus where the state-of-the-art is reached, almost half of the complete nestings ($m_{nesting}$) are correctly recognized, which is a reliable indicator of our model performance on the nested NER task. These results suggest that the MLC architecture should be considered in future state-of-the-art comparisons due to its effectiveness. Besides, we argue that there is still much work to be done in nested NER, as most models fail to simultaneously recognize the inner and outer entities of nestings, which is one of the main objectives of the task.

5 Conclusions and Future Work

This paper presented an effective but overlooked neural model for nested NER based on sequence labeling architectures. Specifically, we revisited the Multiple LSTM-CRF (MLC) approach, which uses a single flat NER model per entity type. We argue that this approach has not been analyzed in-depth since large pre-trained language models have not been incorporated. Our experimental results show that by adding a character-level language model to the MLC architecture, it achieves state-of-the-art in the Chilean Waiting List corpus. One of the main advantages of using this approach is that it can handle entities tagged with more than one entity type, barely addressed in previous works.

In addition, to alleviate some gaps found in current evaluation metrics, we proposed new task-specific metrics that adequately measure the performance of models on nested entities. The results according to these metrics are low, especially when it comes to recognizing complete nestings, i.e., inner and outer entities simultaneously. This finding shows that most models are better at identifying flat entities or part of nested entities, which is not the primary goal of the task. We hope that our study will help raise awareness in the research community that overlooking intuitive models and using only standard metrics when evaluating a new complex solution can be misleading and create an overly optimistic impression of the new solution's performance.

References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. Flair: An easy-to-use framework for state-of-the-art nlp. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Beatrice Alex, Barry Haddow, and Claire Grover. 2007. Recognising nested named entities in biomedical text. In *Biological, translational, and clinical language processing*, pages 65–72, Prague, Czech Republic. Association for Computational Linguistics.

Pablo Báez, Fabián Villena, Matías Rojas, Manuel Durán, and Jocelyn Dunstan. 2020. The Chilean waiting list corpus: a new resource for clinical named entity recognition in Spanish. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pages 291–300, Online. Association for Computational Linguistics.

Darina Benikova, Chris Biemann, and Marc Reznicek. 2014. NoSta-D named entity annotation for German: Guidelines and dataset. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2524–2531, Reykjavik, Iceland. European Language Resources Association (ELRA).

Nancy Chinchor and Patricia Robinson. 1997. Muc-7 named entity task definition. In *Proceedings of the 7th Conference on Message Understanding*, volume 29, pages 1–21.

Billy Chiu, Gamal Crichton, Anna Korhonen, and Sampo Pyysalo. 2016. How to train good word embeddings for biomedical NLP. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, pages 166–174, Berlin, Germany. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. Bert: Pre-training of deep bidirectional transformers for language understanding.

Hao Fei, Yafeng Ren, and Donghong Ji. 2020. Dispatched attention with multi-task learning for nested mention recognition. *Inf. Sci.*, 513:241–251.

Jenny Rose Finkel and Christopher D. Manning. 2009. Nested named entity recognition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 141–150, Singapore. Association for Computational Linguistics.

R. Florian, H. Hassan, A. Ittycheriah, H. Jing, N. Kambhatla, X. Luo, N. Nicolov, and S. Roukos. 2004. A statistical model for multilingual entity detection and tracking. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 1–8, Boston, Massachusetts, USA. Association for Computational Linguistics.

Yao Fu, Chuanqi Tan, Mosha Chen, Songfang Huang, and Fei Huang. 2020. Nested named entity recognition with partially-observed treecrfs.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Meizhi Ju, Makoto Miwa, and Sophia Ananiadou. 2018. A neural layered model for nested named entity recognition. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1446–1459, New Orleans, Louisiana. Association for Computational Linguistics.

Arzoo Katiyar and Claire Cardie. 2018. Nested named entity recognition revisited. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 861–871, New Orleans, Louisiana. Association for Computational Linguistics.

Jin-Dong Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003. Genia corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics (Oxford, England)*, 19 Suppl 1:i180–2.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. 2019. Sequence-to-nuggets: Nested entity mention

detection via anchor-region networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5182–5192, Florence, Italy. Association for Computational Linguistics.

Liyuan Liu, Jingbo Shang, Frank F. Xu, Xiang Ren, Huan Gui, Jian Peng, and Jiawei Han. 2017. Empower sequence labeling with task-aware neural language model.

Wei Lu and Dan Roth. 2015. Joint mention extraction and classification with mention hypergraphs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 857–867, Lisbon, Portugal. Association for Computational Linguistics.

Aldrian Obaja Muis and Wei Lu. 2017. Labeling gaps between words: Recognizing overlapping mentions with mention separators. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2608–2618, Copenhagen, Denmark. Association for Computational Linguistics.

Takashi Shibuya and Eduard Hovy. 2020. Nested named entity recognition via second-best sequence learning and decoding.

Mohammad Golam Sohrab and Makoto Miwa. 2018. Deep exhaustive model for nested named entity recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2843–2849, Brussels, Belgium. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Jana Straková, Milan Straka, and Jan Hajic. 2019. Neural architectures for nested NER through linearization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5326–5331, Florence, Italy. Association for Computational Linguistics.

Bailin Wang and Wei Lu. 2018. Neural segmental hypergraphs for overlapping mention recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 204–214, Brussels, Belgium. Association for Computational Linguistics.

Jue Wang, Lidan Shou, Ke Chen, and Gang Chen. 2020. Pyramid: A layered model for nested named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5918–5928, Online. Association for Computational Linguistics.

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.

Changmeng Zheng, Yi Cai, Jingyun Xu, Ho-fung Leung, and Guandong Xu. 2019. A boundary-aware neural model for nested named entity recognition. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 357–366, Hong Kong, China. Association for Computational Linguistics.