applied sciences

MDPI

Article

BERT-Based Transfer-Learning Approach for Nested Named-Entity Recognition Using Joint Labeling

Ankit Agrawal [1], Sarsij Tripathi [2], Manu Vardhan [1], Vikas Sihag [3], Gaurav Choudhary [4] and Nicola Dragoni [4,*]

[1] Department of Computer Science & Engineering, National Institute of Technology Raipur, Raipur 492010, Chhattisgarh, India; aagrawal.phd2017.cse@nitrr.ac.in (A.A.); mvardhan.cs@nitrr.ac.in (M.V.)
[2] Department of Computer Science & Engineering, Motilal Nehru National Institute of Technology Allahabad, Prayagraj 211004, Uttar Pradesh, India; sarsij@mnnit.ac.in
[3] Department of Cyber Security, Sardar Patel University of Police, Security and Criminal Justice, Jodhpur 342037, Rajasthan, India; vikas.sihag@policeuniversity.ac.in
[4] DTU Compute, Department of Applied Mathematics and Computer Science, Technical University of Denmark (DTU), 2800 Kongens Lyngby, Denmark; gauravchoudhary7777@gmail.com
* Correspondence: ndra@dtu.dk

Abstract: Named-entity recognition (NER) is one of the primary components in various natural language processing tasks such as relation extraction, information retrieval, question answering, etc. The majority of the research work deals with flat entities. However, it was observed that the entities were often embedded within other entities. Most of the current state-of-the-art models deal with the problem of embedded/nested entity recognition with very complex neural network architectures. In this research work, we proposed to solve the problem of nested named-entity recognition using the transfer-learning approach. For this purpose, different variants of fine-tuned, pretrained, BERT-based language models were used for the problem using the joint-labeling modeling technique. Two nested named-entity-recognition datasets, i.e., GENIA and GermEval 2014, were used for the experiment, with four and two levels of annotation, respectively. Also, the experiments were performed on the JNLPBA dataset, which has flat annotation. The performance of the above models was measured using F1-score metrics, commonly used as the standard metrics to evaluate the performance of named-entity-recognition models. In addition, the performance of the proposed approach was compared with the conditional random field and the Bi-LSTM-CRF model. It was found that the fine-tuned, pretrained, BERT-based models outperformed the other models significantly without requiring any external resources or feature extraction. The results of the proposed models were compared with various other existing approaches. The best-performing BERT-based model achieved F1-scores of 74.38, 85.29, and 80.68 for the GENIA, GermEval 2014, and JNLPBA datasets, respectively. It was found that the transfer learning (i.e., pretrained BERT models after fine-tuning) based approach for the nested named-entity-recognition task could perform well and is a more generalized approach in comparison to many of the existing approaches.

Keywords: named-entity recognition; transfer learning; BERT model; conditional random field; pre-trained model; fine-tuning

check for updates

Citation: Agrawal, A.; Tripathi, S.; Vardhan, M.; Sihag, V.; Choudhary, G.; Dragoni, N. BERT-Based Transfer-Learning Approach for Nested Named-Entity Recognition Using Joint Labeling. *Appl. Sci.* **2022**, *12*, 976. https://doi.org/10.3390/app12030976

Academic Editors: Arturo Montejo-Ráez and Salud María Jiménez-Zafra

Received: 15 November 2021
Accepted: 14 January 2022
Published: 18 January 2022

1. Introduction

There is much focus on identifying and classifying important words present in text into their respective semantic classes, such as DNA, RNA, cell, or protein [1]. These important words are known as named entities (NEs), and the task is known as named-entity recognition (NER). The task of named-entity recognition is important because it further helps in different natural language processing (NLP) tasks such as question answering [2], machine translation [3], relation extraction [4], and many more [5,6]. It is often the case that one entity resides within or overlaps with another entity. The text data of different domains commonly contain overlapping entities. However, most of the research work focuses on flat entities only, i.e., they cannot identify the overlapping or nested entities present in the text [7]. In flat named-entity recognition (or named-entity recognition), each token within the text corpus can be determined as anyone entity type only. In the overlapping or nested-entity recognition problem, each token can be classified as more than one entity type. Due to this, there is a potential loss of information in the flat entity recognition task, which also negatively impacts the subsequent natural language processing tasks. The solution is to try and identify overlapping entities. An example of overlapping entities within a sentence is illustrated in Figure 1.

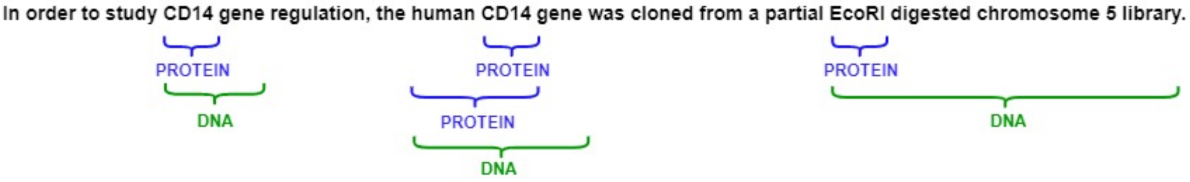
In order to study CD14 gene regulation, the human CD14 gene was cloned from a partial EcoRI digested chromosome 5 library.
PROTEIN
DNA
PROTEIN
PROTEIN
DNA
PROTEIN
DNA

Figure 1. Example of overlapping entities from GENIA dataset.

In the above figure, the word "CD14" is recognized as a PROTEIN type. However, the phrase "CD14 gene" is identified as a DNA type. Similarly, two other overlapping entities can be seen among the PROTEIN and DNA entities in the above example. The annotation was made at multiple levels for each word of the sentence to correctly capture the overlapping entities. For this research work, the experiments were performed using two different nested-entity datasets (GENIA and GermEval 2014) and a flat named-entity-recognition dataset (JNLPBA).

The Bidirectional Encoder Representation from Transformers (BERT) language model recently came into the picture and achieved state-of-the-art results over 11 different natural language processing problems, including named-entity recognition. In the past, many researchers have proposed to solve the problem of nested named-entity recognition using very complex neural network architectures. In recent times, transfer learning has achieved great success and is very widely used to solve different problems in the fields of computer vision and natural language processing. Transfer learning is a well-known approach in which a deep learning model is trained on a large unlabeled dataset (pretrained model) and is further trained on the downstream task dataset (labeled dataset) to fine-tune the pretrained parameters of the pretrained model. The main idea in this research work was to evaluate the effectiveness of the BERT-based transfer-learning approach for the problem of nested entity recognition. The proposed transfer-learning approach (fine-tuning a pretrained BERT-based model) is easy to implement and is simpler than the other models based on complex neural network architecture, which were mostly used earlier to solve the nested named-entity-recognition problem. There are several variants of the pretrained BERT models that are different based on their pretraining on domain-specific texts, or using different vocabulary (word case also matters), etc. Since the pretrained BERT model can be fine-tuned to solve the flat named-entity-recognition problem, we converted the problem of nested entity recognition to the flat named-entity-recognition problem by using the joint labeling technique. The conventional flat named-entity-recognition models could be used without any modification once the labels of different levels were joined together into a single level.

The contributions of this research work are as follows:

- We proposed to solve the nested named-entity-recognition problem using the transfer-learning approach, i.e., by fine-tuning pretrained, BERT-based language models.
- The proposed transfer-learning approach (fine-tuning a pretrained BERT language model) could outperform many of the existing heavily engineered, complex-architecture-based approaches for the nested named-entity-recognition problem.

- The nested datasets were jointly labeled so that conventional named-entity-recognition models could also be used, which treated the nested named entity problem as the flat named-entity-recognition task.
- The experiment was carried with two other well-known machine-learning models (conditional random field and the Bi-LSTM-CRF) for the performance comparison. In addition, the performance of the best-performing proposed model was compared with the existing research work.
- This research work compared the performance of different variants of the pretrained BERT models for the nested named-entity-recognition problem based on domain, size of models (base or large), and cased and uncased versions.
- The results were analyzed and discussed in detail while clarifying the factors that were important in the variants of the pretrained BERT models for different categories, which further led to providing good results for the nested named-entity-recognition problem.

The sections of this paper are arranged as follows: Section 2 covers related works in which similar existing research works have been discussed. Section 3 presents the proposed transfer learning based approach, followed by existing machine-learning models in Section 4, datasets in Section 5, and the evaluation tool used in this research work in Section 6. Section 7 discusses the experimental results and compares the performance by comparing the result of the other models and the existing approaches. Section 8 concludes the paper.

2. Related Works

As discussed above, the annotation was performed at multiple levels to capture the nested information in the named-entity-recognition dataset. Apart from the machine learning model, different modeling techniques must be used to solve the problem of nested named-entity recognition in most cases. This modeling technique includes three different approaches: layering (inside-out and outside-in), cascading, and joint labeling [8,9]. In most of the research works, the layering approach was used. In this research work, the joint labeling modeling technique was used, as it allowed the use of the conventional named-entity-recognition models for identifying the nested entities, and joined the different levels of the nested dataset such that they could be treated as flat entities. Examples of jointly labeled sentences from the nested datasets can be found in Section 5.

Recently, Plank et al., 2021 [10] experimented with a comparison between the cross-language (i.e., German) and in-language for Danish nested entity recognition with different variants of the BERT model. They also presented a new multidomain named entity dataset and experimented with the domain shift problem. They found that BERT-based language models could not perform well for the out-of-domain setup. In another work by Mulyar et al., 2021 [11], a new variant of the BERT model was presented that could perform eight different tasks of clinical information extraction at the same time. It was found that the BERT fine-tuning baseline model performed well in comparison to the proposed multitask model, as a single-task-specific model could better exploit the dataset and its properties. Similarly, Bang et al., 2021 [12] proposed an approach to detect “fake news” related to COVID-19 using different versions of fine-tuned, pretrained, BERT-based language models with the robust loss function.

In the past, the nested named-entity-recognition problem has been solved using one of the following: the neural-network-based approach, the non-neural-network-based approach, and the graph-based approach [13,14].

A new model based on a layered neural network model was presented by Wang et al., 2020 [15] in which pyramid-shaped layers were present, and each layer length was reduced by one when moving from bottom to top. The word embeddings were passed, and each layer l represented the l-gram of the input text. The above model produced good results for different nested named-entity-recognition datasets. Another outside-to-inside approach was proposed by Shibuya et al., 2020 [16]; it also was a neural-network-based approach in which a new objective function and a decoding method that worked iteratively was presented. The model performed similarly to the above neural-network-based model for the nested named-entity-recognition dataset. Another work by Wang et al., 2020 [17] proposed an approach based on a head-tail detector to detect the boundary tokens explicitly. In addition, they have proposed a token-interaction tagger to determine the internal connection among the tokens present within the boundary. There are a number of other neural-network-based approaches that have obtained good results using complex architecture to solve the problem of nested named-entity recognition, such as those presented in [18–20], and many more.

The approaches based on a non-neural network include the constituency-parser-based approach and the graph-based approach. Initially, the constituency-parser-based approach was used by Finkel et al., 2009 [21], in which they represented the sentences using a constituency tree and proposed a CRF-based constituency parser. Recently, a similar approach was proposed by Fu et al., 2020 [22] in which the nested named-entity-recognition problem was solved using a partially observed Tree-CRF model by proposing a new MASKED INSIDE algorithm for computation of probability of partial trees.

Different graph-based approaches have also been used widely for the problem of nested named-entity recognition. They began with the hypergraph-based representation proposed by Lu et al., 2015 [23] to detect correct head, type, and boundary information using a single framework. A similar approach was presented by Wang et al., 2019 [24] and Muis et al., 2018 [25], in which a new segmental hypergraph and mention separator and a multigraph were used for modeling and representation of nested entities, respectively. There are also hybrid models in which graph-based approaches were combined with neural networks to identify the overlapping entities, as in Luo et al., 2020 [26].

Overall, different types of approaches have been used in the past to solve the problem of classification of nested named entities that can provide a good result. However, all the above approaches are either complex in nature or have a complex architecture. In addition, there is a need to explore the transfer-learning approach (i.e., by using the fine-tuned, pretrained language model) for solving the nested named-entity-recognition problem using whichever one has more generalization capabilities compared to any of the existing approaches. Moreover, there are very few existing research works that used a joint labeling modeling technique for nested entity recognition. Hence, in this research work, we proposed to solve this problem using transfer learning (by fine-tuning different variants of pretrained BERT language models) using joint labeling of the nested tags for the nested entity recognition. We also implemented the conditional random field model and the Bi-LSTM-CRF model for comparison of the performance of both models using different NER datasets.

3. Transfer-Learning Approach

In this research work, the nested named-entity-recognition problem was solved using the transfer-learning approach. In this approach, a pretrained language model is used that is already trained on a large unlabeled text dataset. The pretrained model is further trained on a small task-specific text dataset to fine-tune the pretrained parameters. The main motive of using the above-mentioned transfer-learning approach is that it enhances the generalization capability of the model for the low-resource, task-specific text dataset while leveraging the high-resource dataset. The language model is pretrained on the high resource dataset, which is unlabeled (i.e., a plain-text dataset) and is available in abundance. The pretraining task requires a significant computational resource, as a large model is trained on a large plain-text dataset for a considerable amount of time (usually days). However, the fine-tuning of the downstream task is very easy and can be done quickly.

Moreover, prominent NLP researchers have released different pretrained BERT-based language models for public use. In this work, the experiments were performed with different variants of the pretrained BERT language models that fell broadly in the three different categories: Google AI, SciBERT, and the BioBERT pretrained BERT language models. The pretrained models belonging to these categories differed based on the domain of the datasets on which they were pretrained. In addition, there were multiple variants of the pretrained BERT language model in each category that differed based on the case, vocabulary size, language, etc. The experiments were also performed using the conditional random field (CRF) and Bi-LSTM-CRF models so that their results could also be compared with the results of the different variants of the pretrained BERT-based language models. The details of the models and the parameters used were as follows.

3.1. Pretrained BERT Models Used in the Transfer-Learning-Based Approach

Bidirectional Encoder Representations from Transformers (BERT) is a new unsupervised contextualized language representation model that is highly popular for natural language processing tasks. It has been shown that the requirement of heavily engineered task-specific architectures has been reduced significantly by using pretrained representations [27]. This was the first fine-tuning based model to achieve state-of-the-art results on 11 different natural language processing tasks. For natural language processing tasks, the pretraining of the language models has already proved to be effective [27,28]. The pretrained language representation can be applied to downstream natural language processing tasks in two ways: (a) using a fine-tuning-based approach; and (b) using a feature-based approach. The fine-tuning-based approach is minimally dependent on task-specific parameters, i.e., training is performed over downstream tasks while fine-tuning the pretrained parameters. In the feature-based approach, task-specific architectures, including pretrained parameters, are used as additional features. However, during pretraining, the same objective function is used by both approaches, in which language representations are learned using unidirectional models [27]. The BERT model uses the "masked language model" (MLM) and "next sentence prediction" (NSP) pretraining objectives, which mixes the left and right context, allowing the pretraining of a bidirectional deep transformer while removing unidirectional constraints.

In this paper, we used the fine-tuning-based approach, which used already-pretrained models. The pretrained models were trained for different pretraining tasks on unlabeled data from scratch. The details of the pretrained models used in this research work are discussed in further subsections. While performing fine-tuning for downstream tasks, the BERT model began with the parameters of the pretrained models. These parameters were fine-tuned as per the downstream task, which here was nested named-entity recognition. The pretraining and the fine-tuning scheme discussed above can be seen in Figure 2, which was inspired by [27,29].

For this research work, we used the scikit-learn wrapper provided by [30] for fine-tuning the BERT-based models belonging to different categories. For fine-tuning of each of the pretrained BERT models over the named-entity-recognition datasets, the number of epochs was set to 3 (as overfitting was observed for epochs more than 3), the maximum sequence length was set according to the max token length of the wordpiece tokenizer in the training set (plus two to the max token length for the '[CLS]' and '[SEP]' delimiter tokens that BERT uses, so that no data were truncated), the gradient accumulation step was set to 2, the batch size was 8, the validation fraction was set to 0.05, ignore_label was set to other tags according to the dataset, and num_mlp_layers was set to 0 so that linear classifier was used for classification along with the cross entropy loss function for single-label classification. Note that for most of the above and remaining other hyperparameters, the default values were used (the same as in the original BERT paper). For each model, the experiment was conducted three times, with learning rates of 3×10^{-5}, 4×10^{-5}, and 5×10^{-5}. The average result of each run and the standard deviation are reported in the Results section. The scikit-learn wrapper for fine-tuning BERT and the default settings for named-entity-recognition problems were used for the rest of the parameters [30]. The BERT base and large models have 12 and 24 layers (or transformer blocks), 768 and 1024 hidden sizes, and 12 and 16 self-attention heads, respectively. The Adam optimization algorithm and gelu activation function was used in the original BERT model [27]. No manual feature extraction is required in BERT-based models.

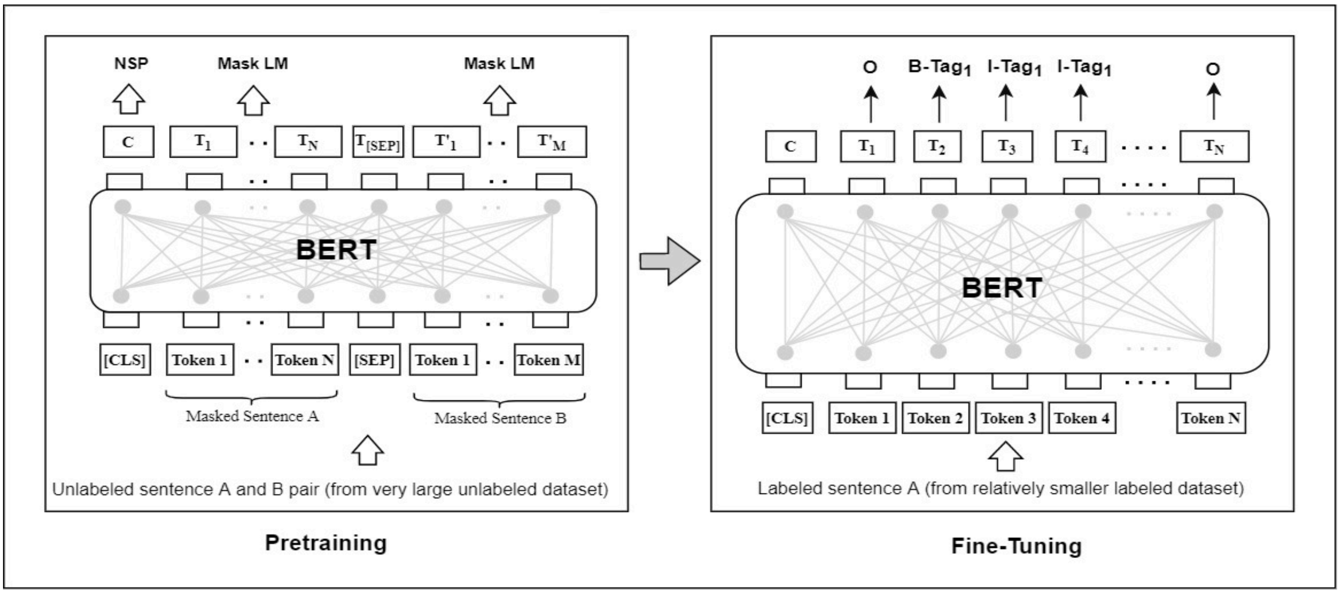

Figure 2. The pretraining and fine-tuning scheme for the BERT model. The same pretrained BERT model can be used for various natural language processing tasks.

The pretrained BERT-based models used for the experiment can be broadly classified into three categories, which are outlined below.

3.1.1. Google AI's Pretrained BERT Models

The basic details of the BERT models belonging to this category were already discussed. They have a multilayer, bidirectional, transformer encoder architecture. Here, the input is an arbitrary span of contiguous text passed as a sequence of tokens. WordPiece embedding, which has a vocabulary of 30,000 tokens, is used [27,31]. This model was trained over the corpus of a general domain, i.e., on BooksCorpus (800 million words) and English Wikipedia (2.5 billion words). The details of parameters used for pre-training the BERT$_{\text{BASE}}$ and BERT$_{\text{LARGE}}$ models are given in [27,30,32]. In addition, a multilingual BERT was used for named entity recognition, as one of the datasets was in the German language [30,32,33]. We experimented with both cased and uncased versions of the above models.

3.1.2. SciBERT Pretrained BERT Models

SciBERT follows the architecture of the BERT model, but was pretrained using scientific text. The designers used the vocabulary provided by BERT as BASEVOCAB. In addition, they constructed their own WordPiece vocabulary named SCIVOCAB using the scientific text corpus with the same vocabulary size. Similar to the above, they also produced both the cased and the uncased version of models. The SciBERT model was trained over scientific text corpus from the Semantic Scholar, which has 1.14 million full-text papers and a total of 3.17 billion tokens [34].

3.1.3. BioBERT Pretrained BERT Models

BioBERT is another pretrained, domain-specific language model, and was pretrained on large-scale biomedical text corpora for the purpose of biomedical text mining. It also has an architecture similar to that of the BERT model. It has been shown that BioBERT significantly outperformed in the three different biomedical text mining tasks, which included: biomedical named-entity recognition, biomedical question answering and biomedical relation extraction. In this paper, we experimented with five different versions of the pretrained BioBERT models. These models used the BERT$_{\text{BASE}}$ pretrained model and were further pretrained over combinations of PubMed and PMC corpora for the different numbers of steps. Further details on the models that were used in the experiment can be found in [30,35].

4. Existing Machine-Learning Models

4.1. Conditional Random Field Model

The conditional random field model is commonly used for sequence labeling, as it allows both the flow of probabilistic information across the sequence and discriminative training. Given some observation sequences, the conditional random field represents the probability of hidden state sequences. The non-independent and overlapping features in the observation sequence can be modeled using a conditional random field (CRF). Other theoretical details of the conditional random field model have been skipped, but can be found in [36,37]. Figure 3 shows the basic workflow diagram for the named-entity recognition using the conditional random field model used in this research work. For implementing the conditional random field model, python's sklearn-crfsuite library was used [38]. The training and testing dataset were initially available in the CoNLL 2002 format, which was further preprocessed and stored as a list of lists of tuples. The outermost list contained all the list of sentences; the individual sentences were also stored in the list data structure of python, and the words along with their respective features (if any) and correct labels were stored in the tuple data structure of python before passing the dataset for manual feature extraction. The feature extraction is discussed in detail in a further section. The complete extracted features were stored as a list of lists of dictionaries. Similar to above, the outermost list contained complete features of all the sentences; inner lists contained features of individual sentences, and the dictionaries inside the inner list contained the features of a particular word of a sentence in order. All the parameters used for the conditional random field (CRF) model for this research work were set according to [38].

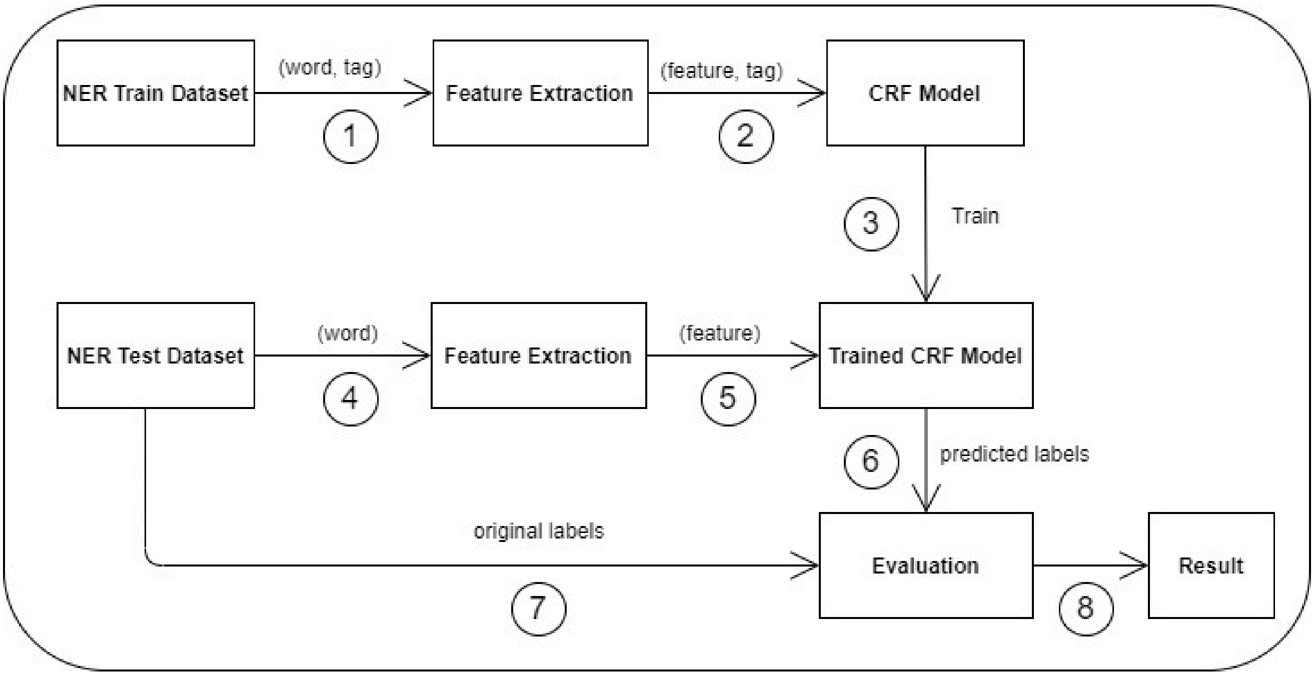

Figure 3. The workflow used for NER using conditional random field (CRF) model.

Table 1. Sample features extracted from a sentence of GermEval 2014 train dataset.

Sentence	Barauszahlungen	Sind	Grundsätzlich	Nicht	Möglich	.
Label	**O + O**	**O + O**	**O + O**	**O + O**	**O + O**	**O + O**
Features	{‘word.lower()’: ‘barauszahlungen’, ‘word.len()’: 15, ‘word.hasHyphen()’: False, ‘word[−4:]’: ‘ngen’, ‘word[−3:]’: ‘gen’, ‘word[−2:]’: ‘en’, ‘word[:2]’: ‘Ba’, ‘word[:3]’: ‘Bar’, ‘word[:4]’: ‘Bara’, ‘word.type’: ‘Alpha’, ‘word.case()’: ‘Title’, ‘word.pattern()’: ‘ULLLLLLLLLLLLLL’, ‘stem’: ‘barauszahl’, ‘BOS’: True, ‘+1:word.lower()’: ‘sind’, ‘+1:word.len()’: 4, ‘+1:word.hasHyphen()’: False, ‘+1:word.type’: ‘Alpha’, ‘+1:word.case()’: ‘Lower’, ‘+1:word.pattern()’: ‘LLLL’, ‘+2:word.lower()’: ‘grundsätzlich’, ‘+2:word.len()’: 13, ‘+2:word.hasHyphen()’: False, ‘+2:word.type’: ‘Alpha’, ‘+2:word.case()’: ‘Lower’, ‘+2:word.pattern()’: ‘LLLLLLLLLLLLL’}	{‘word.lower()’: ‘sind’, ‘word.len()’: 4, ‘word.hasHyphen()’: False, ‘word[−4:]’: ‘sind’, ‘word[−3:]’: ‘ind’, ‘word[−2:]’: ‘nd’, ‘word[:2]’: ‘si’, ‘word[:3]’: ‘sin’, ‘word[:4]’: ‘sind’, ‘word.type’: ‘Alpha’, ‘word.case()’: ‘Lower’, ‘word.pattern()’: ‘LLLL’, ‘stem’: ‘sind’, ‘−1:word.lower()’: ‘barauszahlungen’, ‘−1:word.len()’: 15, ‘−1:word.hasHyphen()’: False, ‘−1:word.type’: ‘Alpha’, ‘−1:word.case()’: ‘Title’, ‘−1:word.pattern()’: ‘ULLLLLLLLLLLLLL’, ‘+1:word.lower()’: ‘grundsätzlich’, ‘+1:word.len()’: 13, ‘+1:word.hasHyphen()’: False, ‘+1:word.type’: ‘Alpha’, ‘+1:word.case()’: ‘Lower’, ‘+1:word.pattern()’: ‘LLLLLLLLLLLLL’, ‘+2:word.lower()’: ‘nicht’, ‘+2:word.len()’: 5, ‘+2:word.hasHyphen()’: False, ‘+2:word.type’: ‘Alpha’, ‘+2:word.case()’: ‘Lower’, ‘+2:word.pattern()’: ‘LLLLL’}	{‘word.lower()’: ‘grundsätzlich’, ‘word.len()’: 13, ‘word.hasHyphen()’: False, ‘word[−4:]’: ‘lich’, ‘word[−3:]’: ‘ich’, ‘word[−2:]’: ‘ch’, ‘word[:2]’: ‘gr’, ‘word[:3]’: ‘gru’, ‘word[:4]’: ‘grun’, ‘word.type’: ‘Alpha’, ‘word.case()’: ‘Lower’, ‘word.pattern()’: ‘LLLLLLLLLLLLL’, ‘stem’: ‘grundsatz’, ‘−1:word.lower()’: ‘sind’, ‘−1:word.len()’: 4, ‘−1:word.hasHyphen()’: False, ‘−1:word.type’: ‘Alpha’, ‘−1:word.case()’: ‘Lower’, ‘−1:word.pattern()’: ‘LLLL’, ‘−2:word.lower()’: ‘barauszahlungen’, ‘−2:word.len()’: 15, ‘−2:word.hasHyphen()’: False, ‘−2:word.type’: ‘Alpha’, ‘−2:word.case()’: ‘Title’, ‘−2:word.pattern()’: ‘ULLLLLLLLLLLLLL’, ‘+1:word.lower()’: ‘nicht’, ‘+1:word.len()’: 5, ‘+1:word.hasHyphen()’: False, ‘+1:word.type’: ‘Alpha’, ‘+1:word.case()’: ‘Lower’, ‘+1:word.pattern()’: ‘LLLLL’, ‘+2:word.lower()’: ‘möglich’, ‘+2:word.len()’: 7, ‘+2:word.hasHyphen()’: False, ‘+2:word.type’: ‘Alpha’, ‘+2:word.case()’: ‘Lower’, ‘+2:word.pattern()’: ‘LLLLLLL’}	{‘word.lower()’: ‘nicht’, ‘word.len()’: 5, ‘word.hasHyphen()’: False, ‘word[−4:]’: ‘icht’, ‘word[−3:]’: ‘cht’, ‘word[−2:]’: ‘ht’, ‘word[:2]’: ‘ni’, ‘word[:3]’: ‘nic’, ‘word[:4]’: ‘nich’, ‘word.type’: ‘Alpha’, ‘word.case()’: ‘Lower’, ‘word.pattern()’: ‘LLLLL’, ‘stem’: ‘nicht’, ‘−1:word.lower()’: ‘grundsätzlich’, ‘−1:word.len()’: 13, ‘−1:word.hasHyphen()’: False, ‘−1:word.type’: ‘Alpha’, ‘−1:word.case()’: ‘Lower’, ‘−1:word.pattern()’: ‘LLLLLLLLLLLLL’, ‘−2:word.lower()’: ‘sind’, ‘−2:word.len()’: 4, ‘−2:word.hasHyphen()’: False, ‘−2:word.type’: ‘Alpha’, ‘−2:word.case()’: ‘Lower’, ‘−2:word.pattern()’: ‘LLLL’, ‘+1:word.lower()’: ‘möglich’, ‘+1:word.len()’: 7, ‘+1:word.hasHyphen()’: False, ‘+1:word.type’: ‘Alpha’, ‘+1:word.case()’: ‘Lower’, ‘+1:word.pattern()’: ‘LLLLLLL’, ‘+2:word.lower()’: ‘.’, ‘+2:word.len()’: 1, ‘+2:word.hasHyphen()’: False, ‘+2:word.type’: ‘None’, ‘+2:word.case()’: ‘None’, ‘+2:word.pattern()’: ‘.’}	{‘word.lower()’: ‘möglich’, ‘word.len()’: 7, ‘word.hasHyphen()’: False, ‘word[−4:]’: ‘lich’, ‘word[−3:]’: ‘ich’, ‘word[−2:]’: ‘ch’, ‘word[:2]’: ‘mö’, ‘word[:3]’: ‘mög’, ‘word[:4]’: ‘mögl’, ‘word.type’: ‘Alpha’, ‘word.case()’: ‘Lower’, ‘word.pattern()’: ‘LLLLLLL’, ‘stem’: ‘moglich’, ‘−1:word.lower()’: ‘nicht’, ‘−1:word.len()’: 5, ‘−1:word.hasHyphen()’: False, ‘−1:word.type’: ‘Alpha’, ‘−1:word.case()’: ‘Lower’, ‘−1:word.pattern()’: ‘LLLLL’, ‘−2:word.lower()’: ‘grundsätzlich’, ‘−2:word.len()’: 13, ‘−2:word.hasHyphen()’: False, ‘−2:word.type’: ‘Alpha’, ‘−2:word.case()’: ‘Lower’, ‘−2:word.pattern()’: ‘LLLLLLLLLLLLL’, ‘+1:word.lower()’: ‘.’, ‘+1:word.len()’: 1, ‘+1:word.hasHyphen()’: False, ‘+1:word.type’: ‘None’, ‘+1:word.case()’: ‘None’, ‘+1:word.pattern()’: ‘.’}	{‘word.lower()’: ‘.’, ‘word.len()’: 1, ‘word.hasHyphen()’: False, ‘word[−4:]’: ‘.’, ‘word[−3:]’: ‘.’, ‘word[−2:]’: ‘.’, ‘word[:2]’: ‘.’, ‘word[:3]’: ‘.’, ‘word[:4]’: ‘.’, ‘word.type’: ‘None’, ‘word.case()’: ‘None’, ‘word.pattern()’: ‘.’, ‘stem’: ‘.’, ‘EOS’: True, ‘−1:word.lower()’: ‘möglich’, ‘−1:word.len()’: 7, ‘−1:word.hasHyphen()’: False, ‘−1:word.type’: ‘Alpha’, ‘−1:word.case()’: ‘Lower’, ‘−1:word.pattern()’: ‘LLLLLLL’, ‘−2:word.lower()’: ‘nicht’, ‘−2:word.len()’: 5, ‘−2:word.hasHyphen()’: False, ‘−2:word.type’: ‘Alpha’, ‘−2:word.case()’: ‘Lower’, ‘−2:word.pattern()’: ‘LLLLL’}

Feature Extraction for the CRF Model

Feature extraction is an essential step for machine-learning models. The features describe the dataset effectively, and can be passed to the machine learning model for training and testing, respectively. The CRF model is trained over the extracted features from the training dataset, and later, extracted features from the test dataset are passed to the trained CRF model to predict the correct tags according to input test dataset features. In this subsection, the features used for the CRF model for different datasets are described in detail after their introduction. The base form (of any word) is the root of the verb without any suffixes (such as -ed, -s, and -ing). Similarly, stemming is the process of reducing any word to its stem but not necessarily to its dictionary root. Part-of-speech (POS) tagging is used to identify how the words are used in a sentence. There are different parts of speech tags, such as noun, verb, pronoun, adverb, etc. Chunking helps in the identification of phrases present in unstructured text. It has labels such as noun phrase (NP), verb phrase (VP), etc. For the GENIA and JNLPBA datasets, we used the GENIA tagger [39,40] to provide the base form, POS tagging, and chunking. Here, chunking also followed the Begin–Inside–Outside (BIO) format. The base form, POS tags, and chunking tags were appended to the original GENIA and JNLPBA datasets, to be used as a feature. Since the GENIA tagger would not have provided good results on the German dataset, we used nltk's snowball German stemmer and appended its outputs to the original GermEval 2014 dataset. The other features for all the three datasets included: begin of sentence (BOS) and end of sentence (EOS) markers for the beginning and end of sentences as Boolean type; word in lowercase; length of word; suffix and prefix of word; type of word (i.e., whether the word was a type of digit, alphanumeric, alphabetic, or none of above); whether the word has a hyphen (as Boolean type); pattern present in word (i.e., pattern obtained after replacing the following: uppercase characters present in the word with "U", lowercase characters with "L", full-stop and comma characters with a full-stop character, digits with "D", symbols ("_", "+", "*", "/", "=", "|") with "#", symbols (":", ";", "!", "?") with ";", and braces (">", ")", "}", "]") with ")" and braces ("<", "(", "{", "[") with "("; case information of the word (i.e., whether the word was in title case, uppercase, lowercase, or none of these); and finally, context information of the word having features mentioned as above, with a window size of 2. An example of a sample feature extracted from a sentence of the training set of the GermEval 2014 dataset is presented in Table 1. The English meaning (according to Google Translate) of the sample sentence presented in Table 1 is "Cash payments are generally not possible".

4.2. Bidirectional LSTM-CRF Model

The Bi-LSTM-CRF model is a combination of the bidirectional LSTM and CRF layer. Here, the model had access to the sentence-level label information, as well as the past and future input features. The GLOVE-based pretrained word vectors, which were trained over 840 billion tokens with 300 dimensions, have been used for word embedding [41]. The FastText German word embeddings [42] were only used for the GermEval 2014 dataset, as the results using above word embeddings were not good for the German dataset. Firstly, the word embedding for each word was obtained using the vocabulary and the pretrained word vectors. Secondly, the contextual word representation was obtained by passing the token representation to the Bi-LSTM layer. Finally, the decoding of the contextual word representation was done for the prediction. The existing code from [43,44] was used for implementation.

5. Datasets

Three different datasets were used in this research work for experiments; namely, the GENIA, GermEval 2014 (German dataset), and JNLPBA datasets. All the above three datasets were divided into training and testing sets only to keep uniformity. However, the last 10% of sentences were taken from the training set of each dataset to be used as the validation set for the Bi-LSTM-CRF model only. The first two datasets (i.e., GENIA and GermEval 2014) have nested entities, and the JNLPBA dataset has flat entities. The GENIA and the JNLPBA datasets are from the biomedical domain. All the datasets were having named entities labeled in Begin–Inside–Outside (BIO2) format in which the first word of any entity starts with 'B-' indicating the beginning of the label and other remaining words of that entity begins with 'I-' indicating inside of the label. Also, the word (or the token) that are labeled with 'O' are not named entities. For the nested dataset, different levels of annotations were jointly labeled. A sample sentence is shown as an example for each of the nested datasets. The only disadvantage of joint labeling was that there was a significant increase in the number of classes in which each word could be identified. However, the advantage was that all the conventional models used for the flat named-entity recognition could be used for nested named-entity recognition. Further details of all three datasets are discussed below.

5.1. GermEval 2014 Dataset

This dataset is a nested dataset for German named-entity recognition and was presented by [45] for the GermEval 2014 Named-Entity Recognition Shared Task [46]. This dataset consists of around 31,000 manually labeled sentences from German online news and German Wikipedia. There are 12 categories of labels in the dataset, out of which 4 main categories are: ORGanization, LOCation, PERson, and OTHer. They also used two fine-grained labels for each of the above four main categories, i.e., -part and -deriv for partial and derived named entities [45]. For example, "EU" belongs to an Organization category; but "EU-Verwaltung" (English meaning: EU administration) is identified as Organization_part. There are many other examples in which phrases partly contains names, such as "deutschlandweit" (English meaning: Germany-wide). Similarly, the derivations are separately identified. For example, "österreichischen" (English meaning: Austrian) is identified as Location_deriv in the dataset [45,46]. This dataset has two levels of nested labeling. A sample sentence is presented in Table 2.

Table 2. Sample sentence from GermEval 2014 dataset along with nested level annotation (L1 and L2) and joint labeling. The English translation of the sentence (according to Google Translate) was: "From 4 p.m., the pursuers Aston Villa and Tottenham Hotspur will be challenged".

Sentence	Label L1	Label L2	Joint Label
Ab	O	O	O + O
16	O	O	O + O
Uhr	O	O	O + O
sind	O	O	O + O
dann	O	O	O + O
die	O	O	O + O
Verfolger	O	O	O + O
Aston	B-ORG	B-LOC	B-ORG + B-LOC
Villa	I-ORG	O	I-ORG + O
und	O	O	O + O
Tottenham	B-ORG	B-LOC	B-ORG + B-LOC
Hotspur	I-ORG	O	I-ORG + O
gefordert	O	O	O + O
.	O	O	O + O

5.2. GENIA Dataset

The GENIA dataset is a semantically annotated dataset that contains 2000 abstracts from the MEDLINE database [47]. It has four levels of nesting and five types of entities after simplification (DNA, Protein, cell-line, RNA, and cell-type). We followed [21,23] and kept about 90% of data in the training set, and about 10% of the data were present in the testing set. A sample sentence showing labels with four nested levels is presented in Table 3.

Table 3. Sample sentence from GENIA dataset along with nested level annotation (L1, L2, L3, and L4) and joint labeling.

Sentence	Label L1	Label L2	Label L3	Label L4	Joint Label
In	O	O	O	O	O + O + O + O
order	O	O	O	O	O + O + O + O
to	O	O	O	O	O + O + O + O
study	O	O	O	O	O + O + O + O
CD14	B-protein	B-DNA	O	O	B-protein + B-DNA + O+O
gene	O	I-DNA	O	O	O + I-DNA + O+O
regulation	O	O	O	O	O + O + O + O
,	O	O	O	O	O + O + O + O
the	O	O	O	O	O + O + O + O
human	O	B-protein	B-DNA	O	O + B-protein + B-DNA + O
CD14	B-protein	I-protein	I-DNA	O	B-protein + I-protein + I-DNA + O
gene	O	O	I-DNA	O	O + O + I-DNA + O
was	O	O	O	O	O + O + O + O
cloned	O	O	O	O	O + O + O + O
from	O	O	O	O	O + O + O + O
a	O	O	O	O	O + O + O + O
partial	O	O	O	O	O + O + O + O
EcoRI	B-protein	B-DNA	O	O	B-protein + B-DNA + O+O
digested	O	I-DNA	O	O	O + I-DNA + O+O
chromosome	O	I-DNA	O	O	O + I-DNA + O+O
5	O	I-DNA	O	O	O + I-DNA + O+O
library	O	I-DNA	O	O	O + I-DNA + O+O
	O	O	O	O	O + O + O + O

5.3. *JNLPBA Dataset*

The GENIA project organized the BioNLP Shared Task 2004 [48], in which the JNLPBA dataset was introduced. Like the GENIA dataset, it also has five types of entities (DNA, Protein, cell-line, RNA, and cell-type). In the training set of the JNLPBA dataset, there are about 2000 MEDLINE abstracts, and in the testing dataset, there are 404 MEDLINE abstracts. Further details of all the above datasets are presented in Table 4.

Table 4. Named-entity recognition datasets used in this research work.

Dataset	Training Dataset			Testing Dataset			No. of Entity Types (Except Others)
	No. of Abstracts	No. Sent	No. of Tokens	No. of Abstracts	No. Sent	No. of Tokens	
GENIA	1800 (approx.)	16,692	503,857	200 (approx.)	1854	57,024	5
GermEval 2014	-	26,202	494,506	-	5100	96,499	12
JNLPBA	2000	20,546	494,551	404	4260	101,443	5

6. Evaluation Tool and Metrics

In this research work, the F1-score was reported, as it is a standard metric used to evaluate the performance in the problem of named-entity recognition. The F1-score is the harmonic mean of the two other metrics, precision and recall. The F1-score strikes a balance between the precision and the recall. For this research work, we used the third-party tool used in [49] and many others, provided during a CoNLL 2000 shared task for evaluating the F1-score [50].

7. Results and Discussion

This section discusses the performances of the CRF model, Bi-LSTM-CRF model, and different pre-trained BERT models belonging to different categories. The above performances were evaluated using the GENIA dataset (nested biomedical NER dataset), GermEval 2014 dataset (the German language nested NER dataset), and JNLPBA dataset (flat biomedical NER dataset). Since the labels of different levels in the nested datasets were joined together into a single label level, all three different datasets were treated as flat named entity datasets. In addition, a comparison of the best-performing pretrained BERT models for each dataset was made with the existing approaches.

7.1. Discussion of Results for BERT-Based Models

The results for the pretrained BERT models belonging to different categories are discussed in this subsection for the above three different named-entity-recognition datasets. The average F1-score and the standard deviation of three runs are presented for each of the models in Table 5 below.

For the GENIA dataset, the overall best F1-score of 74.38 was obtained by the biobert-base-cased pretrained BERT model (C.1). In category A of the pretrained BERT models (i.e., Google AI's pretrained BERT models), the large and cased version performed better in comparison to the base and the uncased versions of the pretrained models. The best-performing model was the large-cased model. The worst performance in this category was obtained by both the cased and uncased multilingual BERT models. In category B of the pretrained BERT models (i.e., the SciBERT pretrained BERT models), the uncased model with scivocab (B.1) performed best, with an F1-score of 74.07, followed by the remaining models in this category. It is important to note that here, the uncased model performed better than the cased model, and all the models in this category performed better than the models in category A. In category C of the pretrained BERT models (i.e., the BioBERT pretrained BERT models), the base cased model (C.1) performed the best, with an overall F1-score of 74.38 for the GENIA dataset; its performance was followed by models C.2, C.4, and C.5. Since the BioBERT model obtained the overall best results, it was clear that this result was obtained due to domain-based pretraining. In addition, most of the models in this category performed better than the models in other two categories.

For the GermEval 2014 dataset (German language NER dataset), the overall best F1-score of 85.29 was obtained by Google's multilingual base cased model (A.6). In category A of the pretrained BERT models (i.e., Google AI's pretrained BERT models), the performances of both the multilingual BERT models were far better than any of the other models. Their performance was followed by the large BERT models (cased and uncased), and then similarly by the base models. In addition, the results were dependent on both the domain of the pretraining dataset and the model size in this case, as the large model performed slightly better than the base models. In category B of the pre-trained BERT models (i.e., the SciBERT pretrained BERT models), the models with the BASEVOCAB vocabulary performed better than the SCIVOCAB-vocabulary-based models. In addition, the performances of the cased models were significantly better than that of the uncased models. The best results in the category were obtained by the basevocab cased model (B.4); i.e., an F1-score of 79.05. In category C of the pretrained BERT models (i.e., the BioBERT pretrained BERT models), the best-performing model in this category (C.5) had a F1-score of 77.02, which was far behind the overall best F1-score of the model (A.6). Here, it was observed that the results for the cased model were slightly better than the uncased for all the models. It is important to note that all the models in category A performed better than the models in the other two categories.

For the JNLPBA dataset, the overall best F1-score of 80.68 was obtained by the scivocab-uncased pretrained BERT model (B.1). In category A of the pre-trained BERT models (i.e., Google AI's pretrained BERT models), the base cased model (A.1) had the best F1-score. Its performance was followed by both the large cased and uncased versions of the pretrained model. The difference in results was not that significant among other models. In category B and category C of the pretrained BERT models (i.e., the SciBERT and BioBERT pretrained BERT models, respectively), almost all the models had an F1-score greater than 80. However, as discussed above, the overall best F1-score of 80.68 was obtained by the B.1 model, followed by the C.4 model, which attained an F1-score of 80.48 among the pre-trained BERT models in the C category. It was observed for the JNLPBA dataset, in most of the cases, the uncased version of the models performed slightly better than the cased version of the model. In category B, the SCIVOCAB-vocabulary-based models also performed better in comparison to the BASEVOCAB-based pretrained models. In addition, most of the time, the models in categories B and C performed better in comparison to the models in category A.

7.2. Discussion of Results for the CRF Model

The results obtained by the CRF model are also presented in Table 5. This model is still the most popular for the named-entity-recognition problem and can be used for both nested and non-nested datasets. The named-entity recognition for all three datasets could be treated as a flat named-entity-recognition problem. The CRF model used the feature described above obtained an F1-score of 65.15, 68.93, and 74.23 for the GENIA, GermEval 2014, and JNLPBA datasets, respectively. The results obtained for all three datasets had a very significant difference from any of the pretrained BERT models. This model obtained the worst results, even after the manual feature extraction for each of the datasets.

7.3. Discussion of Results for the Bi-LSTM-CRF Model

The results obtained by the Bi-LSTM-CRF model are recorded in Table 5. This model is also very widely used for sequence-tagging tasks such as named-entity recognition. As mentioned before, the GLOVE (and the FastText) word embeddings were used for obtaining the word embeddings. The Bi-LSTM-CRF model obtained an F1-score of 70.19, 76.14, and 77.56 for the GENIA, GermEval 2014 (using German FastText word vectors), and JNLPBA datasets, respectively. An F1-score of 70.21 was obtained using the GLOVE word vectors (for the English language) for the GermEval 2014 dataset, which was very poor, and hence was not included in the results table. The results obtained for all three datasets were much better than for the CRF model, but they were still significantly worse than for the fine-tuning-based pretrained BERT models for all the datasets.

Table 5. Result for the CRF, Bi-LSTM-CRF, and fine-tuned pretrained BERT models for different categories of three different NER datasets. For all the datasets, the best results of the pretrained BERT model in each category are shown in bold. The results of the overall best-performing model for each of the NER datasets are in bold and underlined.

S. No.		**Model Details**		**Dataset Details**		
				Nested Dataset		**Non-Nested Dataset**
		Model Category	**Model Name**	**F1 Score (GENIA Test Dataset)**	**F1 Score (GermEval 2014 Test Dataset)**	**F1 Score (JNLPBA Test Dataset)**
A.	1.	Google AI's pretrained BERT models	bert-base-uncased	72.91 ± 0.09	79.08 ± 0.24	**80.13 ± 0.04**
	2.		bert-large-uncased	73.11 ± 0.11	80.76 ± 0.24	79.98 ± 0.14
	3.		bert-base-cased	73.19 ± 0.08	79.76 ± 0.25	79.51 ± 0.24
	4.		bert-large-cased	**73.38 ± 0.09**	81.37 ± 0.24	80.11 ± 0.16
	5.		bert-base-multilingual-uncased	72.49 ± 0.24	84.72 ± 0.17	79.42 ± 0.11
	6.		bert-base-multilingual-cased	72.44 ± 0.24	**85.29 ± 0.23**	79.65 ± 0.12
B.	1.	SciBERT pretrained BERT models	scibert-scivocab-uncased	**74.07 ± 0.18**	76.35 ± 0.14	**80.68 ± 0.22**
	2.		scibert-scivocab-cased	73.56 ± 0.13	77.38 ± 0.13	80.60 ± 0.07
	3.		scibert-basevocab-uncased	73.34 ± 0.05	78.66 ± 0.11	80.33 ± 0.18
	4.		scibert-basevocab-cased	73.57 ± 0.21	**79.05 ± 0.23**	79.82 ± 0.18
C.	1.	BioBERT pretrained BERT models	biobert-base-cased	**74.38 ± 0.14**	75.67 ± 0.26	80.43 ± 0.14
	2.		biobert-v1.1-pubmed-base-cased	74.29 ± 0.07	75.76 ± 0.32	80.42 ± 0.14
	3.		biobert-v1.0-pubmed-base-cased	73.63 ± 0.07	76.32 ± 0.36	80.25 ± 0.15
	4.		biobert-v1.0-pubmed-pmc-base-cased	73.79 ± 0.21	76.62 ± 0.11	**80.48 ± 0.04**
	5.		biobert-v1.0-pmc-base-cased	73.84 ± 0.18	**77.02 ± 0.25**	80.44 ± 0.24
D.		CRF model		65.15 ± 0.21	68.93 ± 0.23	74.23 ± 0.38
E.		Bi-LSTM-CRF		70.19 ± 0.56	76.14 ± 0.31	77.56 ± 0.36

7.4. Comparison of the Results with Other Existing Approaches

The best results of the pretrained BERT models for each of the three datasets were compared with the results of existing approaches. The comparison results for the GENIA dataset with the existing approaches are presented in Table 6.

Similarly, a comparison was made in terms of performance for the GermEval 2014 dataset with the existing approaches. The comparison results for the GermEval 2014 dataset are presented in Table 7.

A similar comparison was made for the JNLPBA dataset with the existing approaches. The comparison results for this dataset are presented in Table 8.

A few important points observed from the above discussion of results are as follows:

- On comparing the CRF, Bi-LSTM-CRF, and BERT-based language models, it was found that almost all the BERT-based models performed better than both the other models. The performance of the Bi-LSTM-CRF models was better than that of the CRF model, but not the fine-tuning-based, pretrained BERT-based models.
- There was a huge impact of the language on the BERT-based model, which was clear from the results for the GermEval 2014 (German) nested NER dataset. Even the Bi-LSTM-CRF model performed poorly if the English GLOVE word vectors were used for the word embedding (due to which the German FastText word vectors were used only for the GermEval 2014 dataset).
- The transfer-learning-based approach without any modifications or any external resources performed well on the GENIA, GermEval 2014, and JNLPBA datasets compared to many of the existing approaches. In Tables 6–8, comparisons were made with existing research work. There were a number of other research works in this area that achieved better results than the presented transfer-learning approach. Note that we are still far from the state-of-the-art results for the above three datasets. Our approach did not possess any kind of complexity in architecture or implementation. The same was not true for the other existing research works. In this study, we wanted to compare the performances of the pretrained, BERT-based transfer-learning approach without using any external resources such as embeddings, unsupervised training on the new dataset, etc. The study was conducted for a performance comparison between the pretrained BERT models based on domain, model size (base or large), and cased and uncased versions.
- Domain-based pretrained models could perform significantly better than the other BERT models pretrained on different domains. For example, the BioBERT-based models performed better on the GENIA dataset, Google's multilingual BERT-based model performed better on the GermEval 2014 dataset, and the SciBERT-based model performed better on the JNLPBA dataset (followed by the BioBERT).
- The model size of the pretrained BERT model can also put some impact on the results (in most cases). However, the result difference may not be very significant between the base and large models in all the cases.
- In most cases (for the GENIA and GermEval 2014 datasets), the performance of the cased version of the model was better than that of the uncased version of the model. However, the uncased versions of the BERT language model performed better on the JNLPBA dataset.
- Some of the common postprocessing methods from the existing research works have been carried to improve the prediction of best-performing models. However, the performance declined, rather than improving. So, postprocessing is not recommended for the named-entity-recognition problem.

Table 6. Comparison of results with existing approaches for GENIA dataset.

Source	Used Approach	F1-Score
[21]	Parser-based	70.33
[23]	Mention-hypergraph-based	68.70
[25]	Multigraph-based	70.80
[51]	Neural-network-based (LSTM, hypergraph features)	73.80
[52]	Neural-network-based (LSTM-CRF, seq2seq, contextual embeddings)	73.90
[13]	Neural-network-based (boundary aware Bi-LSTM)	73.90
This Paper	Transfer-learning-based (best BERT model)	74.38
[16]	Neural-network-based (Bi-LSTM-CRF, contextual embeddings)	77.36
[14]	Neural-network-based (seq2seq, contextual embeddings)	78.31

Table 7. Comparison of results with existing approaches for GermEval 2014 dataset.

Source	Used Approach	F1-Score
[13]	Neural-network-based (boundary aware Bi-LSTM)	71.7
[53]	Neural-network-based (feed forward, Bi-LSTM, Win-bi-LSTM)	76.12
[54]	Neural-network-based (Bi-LSTM-CRF)	75.3
[17]	Neural-network-based (head–tail pair, token interaction tagger)	72.6
This Paper	Transfer-learning-based (best BERT model)	85.29
[55]	Neural-network-based (PolDeepNer2)	87.69
[56]	Transfer-learning-based (unsupervised pretraining, pretrained BERT)	88.6

Table 8. Comparison of results with existing approaches for JNLPBA dataset.

Source	Used Approach	F1-Score
[57]	Neural-network-based (Bi-LSTM, embeddings)	78.4
[17]	Neural-network-based (head–tail pair, token interaction tagger)	74.9
[58]	Neural-network-based (Bi-LSTM, embeddings)	75.87
This Paper	Transfer-learning-based (best BERT model)	80.48
[59]	Neural-network-based (BLSTM-CNN-Char and Spark NLP)	81.29
[60]	Transformer-based	82.0

It is important to note that the existing approaches for the nested named-entity-recognition problem are complex. At the same time, the presented transfer-learning-based approach is much simpler than any of the other existing approaches and can be easily used for similar problems. In addition, it is important to note that the presented transfer-learning-based approach had no requirement for manual feature extraction or the word vectors, while these were needed for the conditional random field and Bi-LSTM-CRF models.

8. Conclusions

In this research work, the transfer-learning approach was used to solve the nested named-entity-recognition problem. The presented transfer-learning approach fine-tuned the pretrained BERT language models for the NER task. The experiments were conducted with different variants of the pretrained BERT-based language models belonging to three popular categories based on the domain. The performance comparison has been done with the existing approaches for each of the datasets. In addition, the experiments were conducted using the conditional random field (CRF) and the Bi-LSTM-CRF models for performance comparison. Manual feature extraction and word embeddings were required for the CRF and Bi-LSTM-CRF models. However, there were no such requirements for the presented transfer-learning approach. The performance was evaluated using two biomedical datasets and a German language NER dataset, out of which one biomedical dataset (i.e., the GENIA dataset) and the German language dataset (i.e., GermEval 2014 dataset) contained nested annotations. The different levels of annotation were joined together for the nested datasets so that the nested named-entity-recognition problem could be treated as a flat named-entity-recognition problem. It was found that the performance of the presented transfer-learning approach was much better than that of the other two models and many of the existing approaches. The presented transfer-learning approach achieved better results than many of the existing research works for the nested and non-nested NER datasets. This research work presented a performance comparison between the pretrained BERT models based on domain, size of models, and cased and uncased versions. It was found that the performance of the presented BERT-based language model depended on the domain and the language of the downstream task. In addition, the presented transfer-learning-based approach had more generalization capability and was much simpler than any of the existing approaches. The presented transfer-learning approach can be used for any of the similar downstream natural language processing tasks. In the future, we will conduct a similar study of several different natural language processing tasks other than named-entity recognition to further test the performance and generalization capabilities of the presented transfer-learning approach.

Author Contributions: Conceptualization, A.A., S.T. and M.V.; methodology, A.A.; software, A.A.; validation, A.A.; investigation, A.A.; resources, A.A., S.T. and M.V.; data curation, A.A.; writing—original draft preparation, A.A.; writing—review and editing, S.T., M.V., V.S., G.C. and N.D.; visualization, A.A.; supervision, S.T. and M.V.; project administration, V.S., G.C. and N.D.; funding acquisition, G.C. and N.D. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the Danish Industry Foundation through “CIDI-Cybersecure IoT in Danish Industry” under Project 2018-0197.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that are used in this research work are openly available at the following links: 1. GermEval 2014 dataset: https://sites.google.com/site/germeval2014ner/data (accessed on: 4 March 2021); 2. GENIA dataset: http://www.geniaproject.org/genia-corpus (accessed on: 5 March 2021); 3. JNLPBA dataset: http://www.geniaproject.org/shared-tasks/bionlp-jnlpba-shared-task-2004 (accessed on: 5 March 2021). All the above datasets can also be found at the GitHub repositories of [13,57].

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

References

1. Li, C.; Wang, G.; Cao, J.; Cai, Y. A Multi-Agent Communication Based Model for Nested Named Entity Recognition. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2021**, *29*, 2123–2136. [CrossRef]
2. Alzubi, J.A.; Jain, R.; Singh, A.; Parwekar, P.; Gupta, M. COBERT: COVID-19 Question Answering System Using BERT. *Arab. J. Sci. Eng.* **2021**, 1–11. [CrossRef]

3. Chauhan, S.; Saxena, S.; Daniel, P. Fully unsupervised word translation from cross-lingual word embeddings especially for healthcare professionals. *Int. J. Syst. Assur. Eng. Manag.* **2021**, 1–10. [CrossRef]
4. Kumar, N.; Kumar, M.; Singh, M. Automated ontology generation from a plain text using statistical and NLP techniques. *Int. J. Syst. Assur. Eng. Manag.* **2016**, *7*, 282–293. [CrossRef]
5. Kumar, R.B.; Suresh, P.; Raja, P.; Sivaperumal, S. Artificial intelligence powered diagnosis model for anaesthesia drug injection. *Int. J. Syst. Assur. Eng. Manag.* **2021**, 1–9. [CrossRef]
6. Parthasarathy, J.; Kalivaradhan, R.B. An effective content boosted collaborative filtering for movie recommendation systems using density based clustering with artificial flora optimization algorithm. *Int. J. Syst. Assur. Eng. Manag.* **2021**, 1–10. [CrossRef]
7. Dai, X. Recognizing Complex Entity Mentions: A Review and Future Directions. In Proceedings of the ACL 2018, Student Research Workshop, Melbourne, Australia, 15–20 July 2018; pp. 37–44. Available online: https://aclanthology.org/P18-3006.pdf (accessed on 15 March 2021).
8. Alex, B.; Haddow, B.; Grover, C. Recognising Nested Named Entities in Biomedical Text. In Proceedings of the Biological, Translational, and Clinical Language Processing, Prague, Czech Republic, 29 June 2007; pp. 65–72. Available online: https://aclanthology.org/W07-1009.pdf (accessed on 15 March 2021).
9. Chen, Y.; Zheng, Q.; Chen, P. A Boundary Assembling Method for Chinese Entity-Mention Recognition. *IEEE Intell. Syst.* **2015**, *30*, 50–58. [CrossRef]
10. Plank, B.; Jensen, K.N.; Van Der Goot, R. DaN+: Danish Nested Named Entities and Lexical Normalization. In Proceedings of the 28th International Conference on Computational Linguistics, Bacelona, Spain, 8–13 December 2020; pp. 6649–6662.
11. Mulyar, A.; Uzuner, O.; McInnes, B. MT-clinical BERT: Scaling clinical information extraction with multitask learning. *J. Am. Med. Inform. Assoc.* **2021**, *28*, 2108–2115. [CrossRef] [PubMed]
12. Bang, Y.; Ishii, E.; Cahyawijaya, S.; Ji, Z.; Fung, P. Model Generalization on COVID-19 Fake News Detection. *arXiv* **2021**, arXiv:2101.03841.
13. Zheng, C.; Cai, Y.; Xu, J.; Leung, H.-F.; Xu, G. 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), Hong Kong, China, 3–7 November 2019; pp. 357–366. Available online: https://aclanthology.org/D19-1034.pdf (accessed on 11 March 2021).
14. Straková, J.; Straka, M.; Hajic, J. Neural Architectures for Nested NER through Linearization. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July 2019; pp. 5326–5331. Available online: http://aclanthology.lst.uni-saarland.de/P19-1527.pdf (accessed on 13 March 2021).
15. Wang, J.; Shou, L.; Chen, K.; Chen, G. Pyramid: A Layered Model for Nested Named Entity Recognition. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 5918–5928. Available online: https://aclanthology.org/2020.acl-main.525.pdf (accessed on 11 March 2021).
16. Shibuya, T.; Hovy, E. Nested Named Entity Recognition via Second-best Sequence Learning and Decoding. *Trans. Assoc. Comput. Linguistics* **2020**, *8*, 605–620. [CrossRef]
17. Wang, Y.; Li, Y.; Tong, H.; Zhu, Z. HIT: Nested Named Entity Recognition via Head-Tail Pair and Token Interaction. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 6027–6036. Available online: https://aclanthology.org/2020.emnlp-main.486.pdf (accessed on 12 March 2021).
18. Chen, Y.; Wu, L.; Deng, L.; Qing, Y.; Huang, R.; Zheng, Q.; Chen, P. A Boundary Regression Model for Nested Named Entity Recognition. *arXiv* **2020**, arXiv:2011.14330.
19. Dadas, S.; Protasiewicz, J. A Bidirectional Iterative Algorithm for Nested Named Entity Recognition. *IEEE Access* **2020**, *8*, 135091–135102. [CrossRef]
20. Tan, C.; Qiu, W.; Chen, M.; Wang, R.; Huang, F. Boundary Enhanced Neural Span Classification for Nested Named Entity Recognition. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 9016–9023. [CrossRef]
21. Finkel, J.R.; Manning, C.D. Nested Named Entity Recognition. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Singapore, 6–7 August 2009; pp. 141–150. Available online: https://aclanthology.org/D09-1015.pdf (accessed on 7 March 2021).
22. Fu, Y.; Tan, C.; Chen, M.; Huang, S.; Huang, F. Nested Named Entity Recognition with Partially-Observed TreeCRFs. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 2–9 February 2021.
23. Lu, W.; Roth, D. Joint Mention Extraction and Classification with Mention Hypergraphs. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 857–867. Available online: https://aclanthology.org/D15-1102.pdf (accessed on 14 March 2021).
24. Wang, B.; Lu, W. Neural Segmental Hypergraphs for Overlapping Mention Recognition. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018.
25. Muis, A.O.; Lu, W. Labeling Gaps Between Words: Recognizing Overlapping Mentions with Mention Separators. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 9–11 September 2018.
26. Luo, Y.; Zhao, H. Bipartite Flat-Graph Network for Nested Named Entity Recognition. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 6408–6418.
27. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. 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, Minneapolis, MN, USA; 2019; Volume 1, pp. 4171–4186. Available online: https://aclanthology.org/N19-1423.pdf (accessed on 19 March 2021).

28. Howard, J.; Ruder, S. Fine-tuned Language Models for Text Classification. *arXiv* **2018**, arXiv:1801.06146.
29. Kang, M.; Lee, K.; Lee, Y. Filtered BERT: Similarity Filter-Based Augmentation with Bidirectional Transfer Learning for Protected Health Information Prediction in Clinical Documents. *Appl. Sci.* **2021**, *11*, 3668. [CrossRef]
30. Nainan, C. Scikit-Learn Wrapper to Finetune BERT. Available online: https://github.com/charles9n/bert-sklearn (accessed on 5 January 2021).
31. Wu, Y.; Schuster, M.; Chen, Z.; Le, Q.V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; et al. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *arXiv* **2016**, arXiv:1609.08144.
32. Rush, A.; Wolf, T.; Debut, L.; Sanh, V.; Chaunmond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; et al. Transformers: State-of-the-Art Natural Language Processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Online, October 2020; pp. 38–45.
33. Team, T.H. Multi-Lingual Models. Available online: https://huggingface.co/transformers/multilingual.html (accessed on 25 August 2021).
34. Beltagy, I.; Cohan, A.; Lo, K. SciBERT: Pretrained Contextualized Embeddings for Scientific Text. 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), Hong Kong, China, 3–7 November 2019.
35. Lee, J.; Yoon, W.; Kim, S.; Kim, D.; Kim, S.; So, C.H.; Kang, J. BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* **2019**, *36*, 1234–1240. [CrossRef] [PubMed]
36. Wallach, H.M. Conditional Random Fields: An Introduction. 2004. Available online: http://www.inference.org.uk/hmw26/papers/crf_intro.pdf (accessed on 7 March 2021).
37. Zhu, X. CS838-1 Advanced NLP: Conditional Random Fields. 2007. Available online: http://pages.cs.wisc.edu/~{}jerryzhu/cs838/CRF.pdf (accessed on 7 March 2021).
38. Korobov, M. Sklearn-Crfsuite Docs. Available online: https://sklearn-crfsuite.readthedocs.io/en/latest/tutorial.html (accessed on 11 August 2021).
39. Tsuruoka, Y.; Tateishi, Y.; Kim, J.-D.; Ohta, T.; McNaught, J.; Ananiadou, S.; Tsujii, J. Developing a Robust Part-of-Speech Tagger for Biomedical Text. In Proceedings of the Advances in Informatics 10th Panhellenic Conference on Informatics, PCI 2005, Volos, Greece, 11–13 November 2005; Bozanis, P., Houstis, E.N., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; pp. 382–392.
40. Tsuruoka, Y.; Tsujii, J. Bidirectional Inference with the Easiest-First Strategy for Tagging Sequence Data. In Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Vancouver, BC, Canada, 6–8 October 2005; Association for Computational Linguistics: Vancouver, BC, Canada, 2005; pp. 467–474.
41. Pennington, J.; Socher, R.; Manning, C.D. GloVe: Global Vectors for Word Representation. Available online: https://nlp.stanford.edu/projects/glove/ (accessed on 5 January 2021).
42. Inc, F. Word Vectors for 157 Languages. Available online: https://fasttext.cc/docs/en/crawl-vectors.html (accessed on 7 August 2021).
43. Huang, Z.; Xu, W.; Yu, K. Bidirectional LSTM-CRF models for sequence tagging. *arXiv* **2015**, arXiv:1508.01991.
44. Genthial, G. Intro to Tf.Estimator and Tf.Data. Available online: https://guillaumegenthial.github.io/introduction-tensorflow-estimator.html (accessed on 6 August 2021).
45. Benikova, D.; Biemann, C.; Reznicek, M. NoSta-D Named Entity Annotation for German: Guidelines and Dataset. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, Iceland, 26–31 May 2014; pp. 2524–2531. Available online: http://www.lrec-conf.org/proceedings/lrec2014/pdf/276_Paper.pdf (accessed on 4 March 2021).
46. Benikova, D.; Biemann, C.; Kisselew, M.; Pado, S. GermEval 2014 Named Entity Recognition Shared Task: Companion Paper. 2014. Available online: https://www.inf.uni-hamburg.de/en/inst/ab/lt/publications/2014-benikovaetal-germeval2014.pdf (accessed on 10 March 2021).
47. Kim, J.-D.; Ohta, T.; Tateisi, Y.; Tsujii, J. GENIA corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics* **2003**, *19*, i180–i182. [CrossRef] [PubMed]
48. Project, G. BioNLP/JNLPBA Shared Task. 2004. Available online: http://www.geniaproject.org/shared-tasks/bionlp-jnlpba-shared-task-2004 (accessed on 11 March 2021).
49. Nguyen, T.-S.; Nguyen, L.-M. Nested Named Entity Recognition Using Multilayer Recurrent Neural Networks BT-Computational Linguistics. In Proceedings of the NAACL-HLT 2018, New Orleans, LA, USA, 1–6 June 2018; Hasida, K., Pa, W.P., Eds.; Springer: Singapore, 2018; pp. 233–246.
50. Tjong Kim Sang, E.F.; Buchholz, S. Introduction to the CoNLL-2000 Shared Task: Chunking. In Proceedings of the Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop, Lisbon, Portugal, 13–14 September 2000; pp. 127–132. Available online: https://www.clips.uantwerpen.be/conll2000/pdf/12732tjo.pdf (accessed on 8 March 2021).
51. Katiyar, A.; Cardie, C. 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, New Orleans, LA, USA; 2018; Volume 1 (Long Papers), pp. 861–871. Available online: https://aclanthology.org/N18-1079.pdf (accessed on 16 March 2021).
52. Wang, B.; Lu, W.; Wang, Y.; Jin, H. A Neural Transition-based Model for Nested Mention Recognition. *arXiv* **2018**, arXiv:1810.01808.

53. Shao, Y.; Hardmeier, C.; Nivre, J. Multilingual Named Entity Recognition using Hybrid Neural Networks. In Proceedings of the Sixth Swedish Language Technology Conference (SLTC); 2016. Available online: https://uu.diva-portal.org/smash/get/diva2:1055627/FULLTEXT01.pdf (accessed on 13 March 2021).
54. Pikuliak, M.; Simko, M.; Bielikova, M. Towards Combining Multitask and Multilingual Learning. In Proceedings of the SOFSEM 2019: Theory and Practice of Computer Science, Nový Smokovec, Slovakia, 27–30 January 2019; Catania, B., Královič, R., Nawrocki, J., Pighizzini, G., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 435–446.
55. Marcińczuk, M.; Radom, J. A Single-run Recognition of Nested Named Entities with Transformers. *Procedia Comput. Sci.* **2021**, *192*, 291–297. [CrossRef]
56. Labusch, K.; Neudecker, C.; Zellhöfer, D. BERT for Named Entity Recognition in Contemporary and Historic German. In Proceedings of the KONVENS, Erlangen, Germany, 9–11 October 2019; pp. 8–11.
57. Sohrab, M.G.; Miwa, M. Deep exhaustive model for nested named entity recognition. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 2843–2849.
58. Gridach, M. Character-level neural network for biomedical named entity recognition. *J. Biomed. Inform.* **2017**, *70*, 85–91. [CrossRef] [PubMed]
59. Kocaman, V.; Talby, D. Biomedical Named Entity Recognition at Scale. *Intell. Comput. Theor. Appl.* **2021**, 635–646. [CrossRef]
60. Yuan, Z.; Liu, Y.; Tan, C.; Huang, S.; Huang, F. Improving Biomedical Pretrained Language Models with Knowledge. In Proceedings of the 20th Workshop on Biomedical Language Processing, Online, 11 June 2021.