future internet

MDPI

Article

On the Use of Knowledge Transfer Techniques for Biomedical Named Entity Recognition †

Tahir Mehmood [1,2,*], Ivan Serina [1], Alberto Lavelli [2], Luca Putelli [1] and Alfonso Gerevini [1]

[1] Department of Information Engineering, University of Brescia, Via Branze 38, 25121 Brescia, Italy
[2] NLP Research Group, Fondazione Bruno Kessler, Via Sommarive 18, 38123 Trento, Italy
* Correspondence: t.mehmood@unibs.it or t.mehmood@fbk.eu

† This paper is an extended version of our papers published in (1) "Combining multitask learning with transfer learning for biomedical named entity recognition", Published in Published in the 24th International Conference on Knowledge-Based and Intelligent Information & Engineering, Virtual Event, 16–18 September (2020); (2) "Leveraging multi-task learning for biomedical named entity recognition", Published in XVIIIth International Conference of the Italian Association for Artificial Intelligence, Rende, Italy, 19–22 November (2019); (3) "Multi-task learning applied to biomedical named entity recognition task", Published in 6th Italian Conference on Computational Linguistics, Bari, Italy, 13–15 November (2019).

Abstract: Biomedical named entity recognition (BioNER) is a preliminary task for many other tasks, e.g., relation extraction and semantic search. Extracting the text of interest from biomedical documents becomes more demanding as the availability of online data is increasing. Deep learning models have been adopted for biomedical named entity recognition (BioNER) as deep learning has been found very successful in many other tasks. Nevertheless, the complex structure of biomedical text data is still a challenging aspect for deep learning models. Limited annotated biomedical text data make it more difficult to train deep learning models with millions of trainable parameters. The single-task model, which focuses on learning a specific task, has issues in learning complex feature representations from a limited quantity of annotated data. Moreover, manually constructing annotated data is a time-consuming job. It is, therefore, vital to exploit other efficient ways to train deep learning models on the available annotated data. This work enhances the performance of the BioNER task by taking advantage of various knowledge transfer techniques: multitask learning and transfer learning. This work presents two multitask models (MTMs), which learn shared features and task-specific features by implementing the shared and task-specific layers. In addition, the presented trained MTM is also fine-tuned for each specific dataset to tailor it from a general features representation to a specialized features representation. The presented empirical results and statistical analysis from this work illustrate that the proposed techniques enhance significantly the performance of the corresponding single-task model (STM).

Keywords: biomedical named entity recognition; deep learning; single-task model; ELMo; transfer learning; multitask learning

check for updates

Citation: Mehmood, T.; Serina, I.; Lavelli, A.; Putelli, L.; Gerevini, A. On the Use of Knowledge Transfer Techniques for Biomedical Named Entity Recognition. *Future Internet* **2023**, *15*, 79. https://doi.org/10.3390/fi15020079

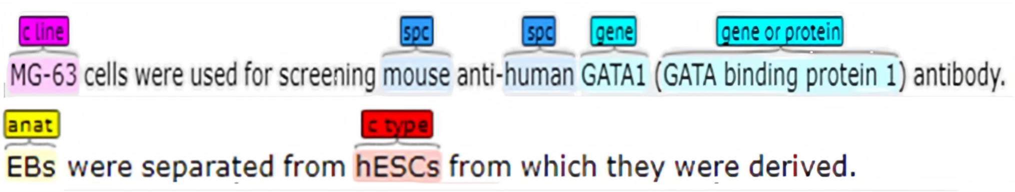
Academic Editors: Choong Seon Hong and Latif U. Khan

Received: 31 December 2022
Revised: 7 February 2023
Accepted: 11 February 2023
Published: 17 February 2023

1. Introduction

In today's era, text data are publishing at a rapid rate and these online text data carry valuable information. Nevertheless, a major share of these data corresponds to unstructured forms, and manually dealing with such a large amount of free text is challenging and problematic. Processing such a quantity of text data requires intelligent techniques based on the problem domain. Natural language processing (NLP), a subfield of artificial intelligence, is used to process unstructured text data, fulfilling the users' needs. NLP enables computers to comprehend, interpret, and manipulate human languages and has been applied to various tasks, including topic discovery and modeling, sentiment analysis, and information extraction, among others. Information extraction (IE) refers to extracting relevant data from unstructured text. IE extends to numerous subtasks, one of which is known as named entity recognition (NER). Named entities refer to the proper nouns presented in the sentences. NER recognizes text of interest and labels them into predefined categories such as person, geographical location, organization, etc. NER is considered a sequence-labeling problem that determines the output tag of the input words presented in the sentence [1–4].

IE has also become a critical activity in the biomedical domain, as biomedical text is also publishing at an increasing rate. Biomedical named entity recognition (BioNER) identifies the biomedical concepts and assigns them to predefined categories such as genes, chemicals, diseases, etc., as shown in Figure 1. In practice, performing the BioNER task is more challenging than a standard NER task as biomedical documents are different from standard text data (e.g., newspaper articles) in several ways. Although there are certain practices followed by researchers for writing biomedical concepts, still, no strict rules exist for the biomedical domain. With open and growing biomedical literature, it becomes more challenging to follow the same naming convention. Another issue concerns the classification of the entities. Different human annotators, even with the same background, can sometimes associate the same word with diverse medical concepts, e.g., "*p53*" corresponds to a protein in the GENIA corpus. In contrast, the HUGA nomenclature annotates it as a gene "*TP53*" [5].


MG-63 cells were used for screening mouse anti-human GATA1 (GATA binding protein 1) antibody.

EBs were separated from hESCs from which they were derived.

Figure 1. An illustration of the biomedical named entity recognition task.

It is also very common in biomedical texts to use different spelling variations for the same entity. For example, "IL12", "IL 12", or "IL-12" refer to the same entity but adopt different writing conventions [6]. Another challenging aspect for BioNER is learning synonyms present in the text, e.g., *PTEN* and *MMAC1* both represent the same gene entity but with different synonyms.

Furthermore, long compound-word entities make the learning process for the BioNER model more complicated, as these entities are expressed using different types of characters. For instance, *"10-Ethyl-5-methyl-5,10-dideazaaminopterin"* and *"12-o-tetradecanoylphorbol 13-acetate"* contain alphanumeric and special characters. Different tokenizers handle these special characters differently. Therefore, applying different tokenizers may produce diverse outputs for the same entity. Descriptive entities, e.g., *"Pigment epithelium-derived factor"*, *"Medullary thymic epithelial cells"*, etc. make it challenging for entity boundary identification. Biomedical entities may also comprise nested entities, e.g., *"CIITA mRNA"* symbolizes an RNA mention; however, *"CIIT"* refers to DNA [7].

Additionally, a practice common in biomedical text writing is the use of acronyms for entities, where an acronym may refer to different entities. For instance, *"TCF"* can refer to *"Tissue Culture Fluid"* or the same acronym may apply to *"T cell factor"* [8]. Similarly, *"EGFR"* can stand for *"estimated glomerular filtration rate"* or *"epidermal growth factor receptor"*. Identifying an acronym for the specific entity depends on the context of the sentence; therefore, the BioNER system must learn how to distinguish them from each other. Additionally, the capitalization feature of the entities in biomedical literature does not provide valuable information about the entity.

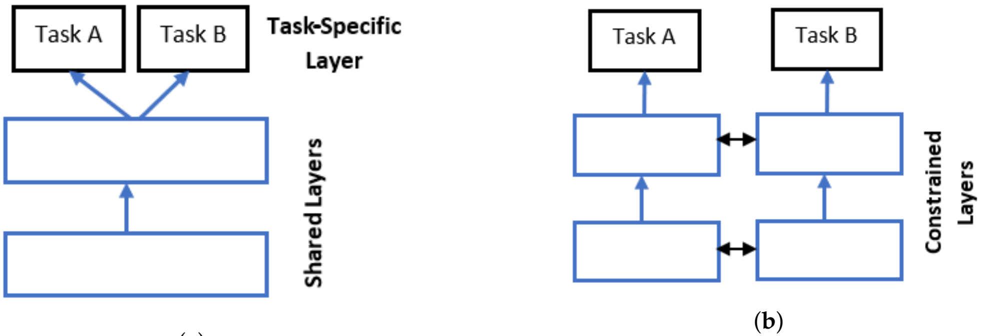
In view of the limitations above, as mentioned earlier, the BioNER task is more challenging compared to the standard NER task. The early BioNER methods (e.g., dictionary-based and rule-based approaches) are effective, but their performance is still limited against open and growing biomedical literature. As compared to the dictionary-based and rule-based methods, the classical machine learning algorithms have shown improved results. The machine learning algorithms require an extensive handcrafted feature engineering phase that has a direct impact on the performance of the models. The performance enhances with more discriminating features, while redundant and irrelevant ones may degrade the performance.

The state-of-the-art techniques are based on deep learning methods, which somehow eliminate the need for handcrafted features, while still producing the desired results. Deep learning (DL) architectures consist of many layers, through which these systems learn the features and complex structure of the data layer by layer. The implicit feature learning ability of the DL models has been successful in different fields, e.g., computer vision [9], speech recognition [10], and drug discovery [11], among others.

1.1. Multitask Learning

It has been observed that the performance of a deep learning model highly correlates with the quantity of annotated data, i.e., the model performance improves with the quantity of data available. Unfortunately, various biomedical tasks lack enough annotated text data, and for this reason, in many cases, deep learning models cannot generalize well. Producing manually annotated data is an expensive and time-consuming job. One solution to such a barrier is to get the benefit of other associated methods that share common features. In a single-task learning approach, various tasks cannot get any benefits from each other as their features cannot be shared among them.

The multitask learning (MTL) approach allows different tasks to share their knowledge among themselves using shared layers in their architecture, thus helping to improve the performance of another task [12]. MTL is an inductive learning process that learns to generalize by utilizing the knowledge of different tasks [13]. When the tasks are sufficiently related, they can provide an inductive bias that forces models to learn generally useful representations [14]. Two different methods are used in the MTL approach, i.e., hard parameter sharing and soft parameter sharing, which are shown in Figure 2. Hard parameter sharing is the most common method used in MTL, where a complete sharing (i.e., parameters) of hidden layers among different tasks is done. This article also focuses on the hard-parameter-sharing approach. In soft parameter sharing, separate models are created for different tasks. These models are then somehow enforced to loosely match the parameters of the shared layers, most commonly done by regularizing the parameters of the shared layers.



(**a**)

(**b**)

Figure 2. Hard-parameter- vs. soft-parameter-shared MTMs: (**a**) hard-parameter-shared MTM; (**b**) soft-parameter-shared MTM.

The MTM can be seen as an implicit data augmentation technique as well. Jointly training various models helps them to transfer their knowledge implicitly using a shared layer(s) [15]. The MTL strategy, therefore, increases the size of the data available to the MTM. The MTL approach helps the MTM to learn those features that cannot be learned in the single-task learning approach for any specific task. In other words, it is difficult for an STM to learn complex features of task *B*, but it is easier to learn when it is trained along with task *A*. This is due to the fact that task *A* provides supervision to the MTM when it comes to learning complex features of task *B*.

MTL optimizes the model during its training to produce a generalized version of the MTM. In single-task learning, a model is more prone to overfitting for a specific task, whereas MTL decreases the possibility of trapping into overfitting as the model has to learn the common representation for all tasks. Training more tasks brings more generalization for MTMs. In MTL, a model focuses on more relevant features, as some tasks give information about irrelevant and relevant features in high-dimensional and noisy data. Similarly, the noise presented in the dataset has less impact in an MTL approach, since the noise is averaged during the training. By using the MTL approach, we keep certain layers of the model shared among various tasks while retaining some layers task-specific, this helps a model to learn useful features for its current task. Training associated tasks simultaneously allow the model to optimize the values of its parameters.

MTL has been widely applied in many domains, e.g., computer vision [16], speech recognition [10], and drug discovery [11]. Collobert and Weston [17] used a CNN-based MTM and trained multiple NLP tasks jointly such as POS tagging, NER, chunking, etc. [18] showed that using the MTL approach, their model increased the performance for historical spelling normalization. Peng and Dredze [19] used the MTL approach for different domains, i.e., Chinese word segmentation and named entity recognition. Plank et al. [20] used an auxiliary loss function for rare words and the primary loss function for the POS tagging task, targeting 22 languages including Finnish, French, and English. Yang et al. [21] used the MTL approach to perform different tasks simultaneously, including POS tagging, chunking, and NER in English, Dutch, and Spanish. Zhang and Weiss [22] used POS tagging as a regularizer of input representation for dependency parsing. Johansson [23] performed the parsing of multiple treebanks in a shared-features representation approach and used one treebank as input to another treebank. Søgaard and Goldberg [24] demonstrated that auxiliary tasks should be used at the innermost layers so that the main task can effectively learn from a shared representation. Hashimoto et al. [25] used a hierarchical model to learn different NLP tasks at successively deeper layers jointly.

1.2. Transfer Learning

Transferring the learned information from one domain to another domain is referred to as transfer learning [26]. Usually, a model is trained on a task in one domain, which is then reused on another related domain or related task [27]. The MTL can also be seen as transfer learning, but in MTL the tasks are learned simultaneously. In contrast, in transfer learning, the tasks are learned sequentially. The transfer learning is done in two stages: pretraining and domain adaptation. The pretraining stage involves the training of the base model, which is then reused on the target task in the adaptation phase. The pretraining phase is expensive, but it is usually required to be performed once. Therefore, it is best practice to choose the source task that can exhibit general representations for many target tasks.

In transfer learning, the model is trained on an auxiliary task, which is then reused on the main task. Similarly, the model can be trained on a source domain which can then be reused on the target domain. For instance, the model can be trained on book reviews and then reused on hotel reviews; in this case, the source and target domains are different, but the source and target tasks are the same. Similarly, the source and domains can be the same while the source and target tasks are different, e.g., the object detection model can be used for image classification. In a third case, both the domains and the tasks are different, e.g., spam classifier is used for radiology text report classification.

In transfer learning, a pretrained model can also be used as a feature extractor or model weight's initialization. Feature extraction is a feature engineering process that is performed using deep learning models instead of performing manually. When using a pretrained model as a feature representation, some of the layers (usually the early/shallow ones) are kept frozen. In this way, the base model works as an input feature for the target model. Moreover, feature extraction is found to be effective for similar tasks in the transfer learning approach. In this sense, the transfer learning technique can be considered a one-time feature engineering method that extracts input features (mostly low-level features, e.g., dots or lines in an image) to other similar tasks.

Transfer learning also involves a fine-tuning method [28], where the weights of the pretrained layers are used in the main task model, and then the whole model is fine-tuned. In this case, the weights of the layers are not kept frozen. The idea is to relearn new features rather than learn them from scratch. The naive example could be to learn how to count numbers coming after five (5), so instead of relearning numbers again from one (1), we start learning from the number six (6). The fine-tuning method actually helps a target model to adopt a task-specific representation from the general-purpose representation of a base model. This approach is useful when the objective is to implement a pretrained model for various tasks.

Radford et al. [29] fine-tuned a pretrained transformer-based language model for task-specific input transformations during an MTL approach. Oquab et al. [30] trained a model on a huge dataset to extract the features for a dataset with few training instances. Al-Stouhi and Reddy [31] empirically showed that the performance of a model could be improved using transfer learning for an imbalanced labels dataset. Yang et al. [32] used a pretrained POS tagging model for word segmentation. Zoph et al. [33] used a high-resource language pair to pretrain a machine translation model, which was then applied to a low-resource language pair. Yosinski et al. [27] performed experiments to compare the feature extractor and the fine-tuning techniques. They found that, for the feature extractor, the performance of the main task model depended on where the layers were cut. Researchers concluded that keeping the top layers' weight frozen could be helpful for similar tasks while keeping the weights of the middle layers frozen lead to performance degradation because of the complex coadaptations they learned. At last, keeping the weights of the lower layers frozen did not show much performance degradation as these layers were more general. In contrast to the feature extraction method, the authors found the fine-tuning method more effective, and it did not require substantial changes in the base model to produce better results.

2. Proposed Methods

2.1. MTM-CNN Model

In this section, a multitask model (MTM-CNN) is proposed that consists of a convolutional neural network (CNN) layer and BiLSTM layers as shown in Figure 3. The proposed MTM-CNN model varies from the model introduced by [34,35] in different ways.

Crichton et al. [34] proposed a multitask model (MTM) based on a CNN to perform BioNER. However, they only focused on the word-level features ignoring the character-level ones. Although the word-level features give much information about the entities, the character-level features help to extract common subword structures among the same entities. Moreover, using only the word-level features can lead to out-of-vocabulary problems when a specific word is not found in the pretrained word embedding. In addition, Wang et al. [35] also performed BioNER using MTM with a single shared BiLSTM layer and found that the word-level and character-level features enhanced the performance of the MTM.

Our proposed approach uses an orthographic-level representation of words, while models presented by Crichton et al. [34] and Wang et al. [35] do not explicitly consider this feature. Various studies have utilized words' orthographic-level information for their models [36–38]. The words' orthographic-level information provides some explicit information to the model, which can help deep learning models to learn orthographic-level features implicitly. This can also help the conditional random field (CRF) whose outputs highly depend on handcrafted features [39]. In an MTM-CNN, the orthographic-level information is used, speculating that the MTM-CNN can extract additional hidden features about the current entity. In this section, the orthographic-level feature is referred to as *case-level features* and both terms can be used interchangeably. The orthographic-level (case-level) representation in the MTM-CNN considers the structure-level information of a word. The case-level features considered in the experiments include the capitalization features of a word, e.g., whether all letters in the specific word are capitalized or lower case, or if the specific word starts with a capital or lower-case letter, whether the word contains digits or all alphabetic characters, etc.

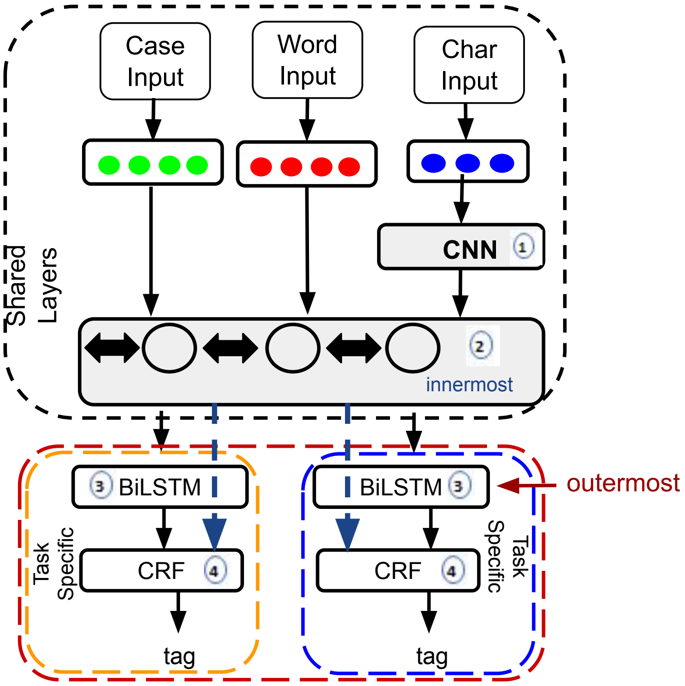


Figure 3. The proposed MTM-CNN (circles represent embeddings). Innermost indicates the task is trained without task-specific BiLSTM. Outermost indicates the task is trained with task-specific BiLSTM.

Differently from Wang et al. [35], the proposed MTM-CNN utilizes a CNN (represented by the circled 1 in Figure 3) instead of BiLSTM, to extract character-level features. Note that the model proposed by Crichton et al. [34] does not take into account the character-level features. Character-level information encourages the model to pull out generic subword structures among the same entities. Additionally, only considering the word-level features can be prone to the out-of-vocabulary problem. Many of the state-of-the-art approaches use a CNN at the character level [40,41] due to its unique feature extraction ability. A CNN perceives global-level features from local-level ones and therefore allows the CNN to pull out additional veiled features [34].

Third, the MTM-CNN implements stacked layers of BiLSTM units in contrast to [35]. The stacked BiLSTM units encourage each of them to perceive the hidden pattern of the data exhibited at various time stamps. This helps the BiLSTM network to gain knowledge of the features at a more abstract level. The first innermost BiLSTM layer (marked by the circled 2 in Figure 3) is shared among all the tasks while the second layer of BiLSTM (shown by the circled 3 in Figure 3) is task-specific. The MTM-CNN exploits CRF (represented by the circled 4 in the figure) at the output layer for final sequence labeling. CRF performs a tagging of the current token by considering neighboring tags at a sentence level [38]. Yang et al. [42] performed experiments with both CRF and Softmax and concluded that CRF produced better results compared to Softmax.

Another of our contributions is the use of different auxiliary tasks for the BioNER task, where two dissimilar auxiliary tasks—GENIA-POS tagging and CoNLL chunking (other than BioNER)—are exploited in the experiments to investigate their impact on the MTM-CNN. The auxiliary tasks are trained in the same way as the other BioNER tasks, i.e., with a task-specific BiLSTM layer (outermost).

Another important aspect of our work concerns analyzing the impact of auxiliary tasks at different levels of MTMs. In the MTL approach, different tasks provide a supervision signal to other tasks. It is important to inspect the proper supervision that can be at any level.

Following this hypothesis, the auxiliary tasks are trained at the innermost (shared) BiLSTM layer without any task-specific BiLSTM. Assuming that this approach makes the innermost shared BiLSTM layer a complete feature representation of that task and propagates more useful signals to the subsequent task-specific BiLSTM layer. The same hypothesis is applied to the auxiliary BioNER tasks where they are trained without task-specific BiLSTM. During the MTM-CNN training, each task is defined with its optimizer, and therefore the loss function related to the specific task is optimized.

2.2. MTM-CW Model

The MTM-CNN model presented in Section 2.1 comprised stacked layers of BiLSTM units. However, moving towards a deep LSTM network can cause the vanishing gradient problem as well [43]. Furthermore, using a very deep architecture for some tasks could also lead to catastrophic interference. In catastrophic interference, the neural network starts forgetting what it has learned previously [44]. In other words, the performance of the neural network drops notably for the previous instances given that it performs well only on the current new instances. The catastrophic forgetting usually occurs at the upper layers [45]. Additionally, the generalized features (learned in an MTL approach) allow the MTMs to perform better on different tasks, but at the same time, they also cause the catastrophic forgetting problem [46] . To address these issues, a new model called MTM-CW is proposed in this section. The proposed multitask model with character and word input representations (MTM-CW) propagates the input embedding information along with the outputs of different shared layers to the subsequent layers as shown in Figure 4. This also encourages a model for continual learning. This also allows successive layers to understand the additional abstract structure from input embeddings and the encoded representation of the previous layers to overcome the vanishing gradient problem and the catastrophic interference in stacked LSTM networks. The skip/residual connections (circled 5 and 6) are represented with dashed arrows in Figure 4 and these skip connections make this model different from our previous proposed model.

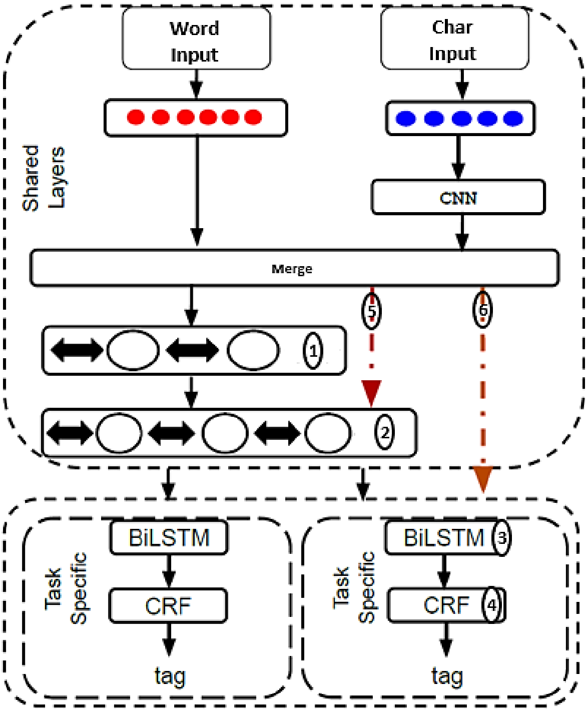


Figure 4. Proposed MTM-CW Model where dashed arrows show skip connections. Circles represent embedding.

2.3. Multi-Task Model with Transfer Learning

The proposed approach uses multitask learning with transfer learning. The MTM model is similar to the one proposed in [35,47], but it is extended with task-specific BiLSTM as shown in Figure 5. The MTM uses word and character representations of the sentence and is trained on different datasets. It is used as a base model and is reused as the starting point of an STM ($MTM \to STM$). The auxiliary task involves the training of an MTM on various datasets, whereas the main task is the training of an STM for each specific dataset that is initialized by the pretrained MTM. For transfer learning, we keep both models, the auxiliary task and the main task same. More specifically, the base MTM is fine-tuned for a specific dataset ($MTM \to STM$), i.e., neither new layer(s) is introduced nor cut off during fine-tuning of the base MTM. Introducing new task-specific layer(s) with randomly initialized weights could decrease the model's performance during the fine-tuning due to the lack of guidance for the new task [48].

Our approach is based on the idea that during the training phase, the MTM learns general features at common layers of the model. The MTL approach allows models to learn those task-specific features, which can be more challenging when learning them independently. In other words, the approach of learning rigorous features by model A during training can be more complex, and so model A discovers these complex features from another task during the MTL approach. When fine-tuning (transfer learning) is performed, the main model uses task-specific features along with generalized features to learn the main task. These generalized features at different levels further help the main model to learn task-specific features. In other words, these generalized features are fine-tuned into task-specific features. The purpose is to move from a generalized model (MTM) to a specialized model (STM). Yosinski et al. [27] performed experiments to compare the feature extractor and the fine-tuning techniques. They found that, for the feature extractor, the performance of the main task model highly depended on the number of layers that were eliminated. On the other hand, for the fine-tuning method, the authors did not find such constraints and produced better results with the fine-tuning approach.

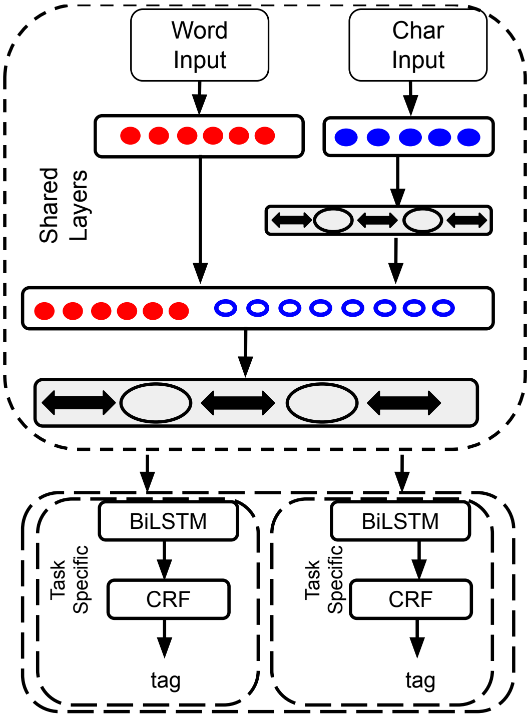


Figure 5. Our proposed model used for fine-tuning ($MTM \to STM$).

Our experiments were conducted for three variants of the proposed method, where the MTM was trained for a particular number of epochs and then the transfer learning was applied, fine-tuning the MTM for a specific dataset. In the first experiment ($MTM \xrightarrow{10} STM$), the MTM^{10} was trained for ten epochs to learn the standard features representation of various tasks and then fine-tuned on a specific dataset. In the second experiment ($MTM \xrightarrow{20} STM$), the MTM^{20} was trained for twenty epochs followed by fine-tuning it for a specific dataset. In the last experiment, the MTM^{cmp} was trained for complete epochs or till an early stop occurred (the early stop was also used for the two previous experiments as well) after which it was fine-tuned for a specific dataset ($MTM \xrightarrow{cmp} STM$).

2.4. Embeddings from Language Models

This section introduces the experiments regarding ELMo (Embeddings from Language Models). ELMo produces contextual-based embeddings, in contrast to word2vec [49] or Glove [50], which generate static single vector embeddings. ELMo word representations work on characters, which allows the network to use lexical knowledge to form reliable representations for out-of-vocabulary tokens. Contrary to other static word embeddings, ELMo produces word vectors at run time. ELMo uses a character-based CNN for input words. The output is the raw word vector and is fed to the first layer of BiLSTM. The output of the first layer forms the intermediate word vector which is also fed to the second layer of BiLSTM, which outputs another intermediate word vector. The final ELMo representation consists of the weighted sum of these two intermediate word vectors and the first raw vectors from the CNN [51]. The character-based CNN produces the noncontextualized word embedding based on the word's characters.

The purpose of this experiment was to explore the performance of the BioNER using pretrained static word embeddings vs. contextualized word embeddings. The contextualized embedding representation was integrated into the model shown in Figure 6, where the model used ELMo embedding with other input representations of the word.

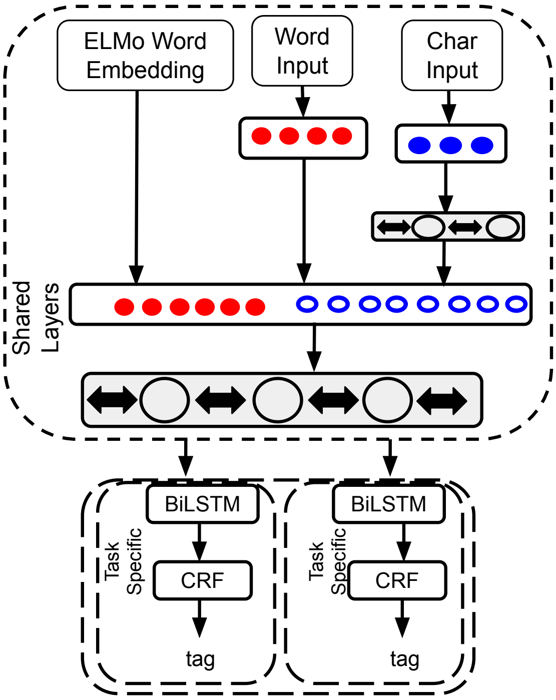


Figure 6. Integration of ELMo in the model.

3. Experiments

Our experiments were performed on 15 datasets which were also used by Wang et al. [35] and Crichton et al. [34]. The bioentities in these datasets were disease, species, cell component, cell, gene/protein, and chemical. Every dataset included training, validation, and test sets. Detained descriptions of the datasets used in the experiments can be found in Appendix A . We followed the experimental setup analogous to Wang et al. (https://github.com/yuzhimanhua/Multi-BioNER, accessed on 1 February 2023), where both training and validation datasets were used for training the model.

We used the IOBES tagging format [52], where I stands for inside-entity when the token occurs inside the entity span, O (outside-entity) represents tokens that do not belong to the entity class, B represents the beginning of an entity, E (end-entity) indicates the last token of the entity span, and S (single-entity) is used when the entity consists of a single token. For performance evaluation of the model, we used the (macroaveraged) F1-score metric (i.e., each class was considered equally important), as it is the most widely adopted for named entity recognition tasks [34,35,53]. As it is usual for named entity recognition tasks, the evaluation basically did not take into account the recognition of single tokens but rather that of the whole entity. More specifically, we evaluated the model performance using the macroaveraged F1-score since it allowed us to avoid biases towards the majority class. The macroaveraged F1-scores reported in the following sections represent the average scores of 10 runs. Henceforth, when mentioning F1-score, we mean the macroaveraged F1-score. For the Tables presented in the later sections, we show the best results with bold font.

Moreover, we utilized a domain-specific pretrained word embedding to avoid a high rate of out-of-vocabulary words problem. The WikiPubMed-PMC word embedding was used, which was trained on a huge quantity of data from PubMedCentral(PMC) articles and PubMed abstracts as well as on English Wikipedia articles [54], whereas the character-level embedding was initialized randomly, and the orthographic-level (case) embedding was described by the identity matrix in which the existence of a word's orthographic feature was defined with one in the diagonal of a matrix.

We also investigated different hypotheses extending the experiments on our proposed models. The first hypothesis concerned the effect of auxiliary tasks on the main task. The MTM-CNN was trained with similar BioNER auxiliary tasks while, in another experiment, the proposed model was additionally trained along with the dissimilar auxiliary tasks which were GENIA-POS tagging and CoNLL chunking.

We also investigated the appropriate layers where effective supervision and complete feature representation from the auxiliary task could be accomplished (see Table 4). For this reason, the auxiliary tasks were trained at the earlier layers (innermost layer in Figure 3) as well as at the later layers (outermost layer in Figure 3). Training at the earlier layers of the model did not include a task-specific BiLSTM layer for the auxiliary tasks while the main task was extended with a task-specific BiLSTM. The empirical results and statistical analysis showed that auxiliary tasks were more beneficial when used at the inner layer. This indicated that there was hierarchical learning between different tasks in the MTL approach.

The experiments for the Embeddings from Language Models (ELMo) implementation was based on AllenNLP (https://github.com/allenai/allennlp, accessed on 1 February 2023) and used a pretrained ELMo model trained on PubMed data. The LSTM unit size was 4096 and the output dimension was 512. In the experiments, the weighted average weights (the pretrained model weight can be found here https://allennlp.org/elmo, accessed on 1 February 2023) of all three layers (CNN, 2*biLMs) were used for the contextualized representation of the word, whereas for static word embedding, wiki-PubMed-PMC was utilized.

We also performed and discuss our statistical analysis in detail to find out the statistical significance among different results with graphical representations for a better and clear understanding. The statistical analysis was performed using Friedman's test [55] to determine the statistical significance of the difference among different models' results.

4. Results and Discussion of MTM-CNN

As a first step, a single-task model (STM-CNN) was implemented for all 15 datasets mentioned earlier. Afterward, the MTM-CNN was trained with all 15 datasets in an MTL approach. Table 1 depicts the comparison between the results of the MTM-CNN and its counterpart the STM-CNN. Each experiment was run 10 times, and the average F1-score of those 10 runs is reported in this section and the best results are shown in bold font.

Table 1. STM vs. MTM-CNN.

Datasets	STM-CNN	MTM-CNN
AnatEM	85.8	**86.9**
BC2GM	**80.9**	80.8
BC4CHEMD	**88.6**	87.3
BC5CDR	85.6	**87.8**
BioNLP09	87.0	**88.7**
BioNLP11EPI	81.4	**84.7**
BioNLP11ID	83.2	**87.6**
BioNLP13CG	81.2	**84.2**
BioNLP13GE	73.3	**79.8**
BioNLP13PC	86.3	**88.8**
CRAFT	**83.8**	83.1
Ex-PTM	72.7	**80.9**
JNLPBA	**74.4**	74.0
LINNAEUS	87.3	**87.7**
NCBI-disease	84.1	**85.6**
Average	82.4	**84.5**

We see that, for most of the datasets, the results improved markedly by using the MTM-CNN with an absolute average difference of F1-score up to 2.1%, showing the importance of the MTL approach in BioNER. More interestingly, the F1-score for BioNLP13GE increased from 73.3 to 79.8. The BC2GM, BC4CHEMD, CRAFT, and JNLPBA showed a performance degradation with the MTL approach. One possible reason could be the size of these datasets. The size of these datasets was bigger than the rest of the other datasets. For this reason, a performance increase was noticed for those datasets that had a small number of entity annotations. This can be seen for Ex-PTM which had a small number of entities and showed a noticeable improvement, increasing to 80.9 from 72.7 in F1-score, with the MTL approach. These results suggested that the MTM-CNN improved the performance of those datasets which did not have many examples. The results illustrated that the MTM-CNN could learn complex features that were difficult to learn in an STM-CNN.

Table 2 shows the comparison of various STM-CNNs with respect to state-of-the-art STMs. Conclusively, our STM-CNN showed an average F1-score gain of up to 4% compared to that in [34], while against Wang et al. [35], our STM-CNN model depicted an average F1-score increase pf up to 2%. For most of the datasets, the STM-CNN showed a performance gain compared to other approaches while the model proposed by Wang et al. [35] illustrated a performance increase on four of these datasets. A prominent increase can be noticed for BioNLP09, where the STM-CNN raised the F1-score from 84.2 to 87.0 against that of the model proposed by Wang et al. [35] The model proposed by Crichton et al. [34] did not exhibit any improvement on any dataset; their model [34] is CNN-based and it also does not consider the character-level information which may lead to the out-of-vocabulary problem. This might be the reason that this model failed to show a performance gain compared to the results of Wang et al. and the MTM-CNN.

The comparison of various MTMs is illustrated in Table 3. We observe that the MTM-CNN model outperformed the model proposed by Crichton et al. [34] with an absolute increase of the average F1-score of up to 4%. It is worth noticing that for BC2GM, the F1-score rose to 80.8 from 73.2 and for BioNLP11ID, the F1-score improved from 81.7 to 87.7. Compared to the multitask model presented by Wang et al. [35], the MTM-CNN showed an average increase of F1-score of 0.8%. The most prominent increase in F1-scores was noted for BioNLP11EPI (+1.6%), BioNLP11ID (+4.4%), BioNLP13CG (+1.8%), and JNLPBA (+1.9%) against Wang et al. [35]

Table 2. Single-task model results comparison.

Datasets	**Wang et al. [35]**	**Crichton et al. [34]**	**STM-CNN**
AnatEM	85.3	81.5	**85.8**
BC2GM	80.0	72.6	**80.9**
BC4CHEMD	**88.7**	82.9	88.6
BC5CDR	**86.9**	83.6	85.6
BioNLP09	84.2	83.9	**87.0**
BioNLP11EPI	77.6	77.7	**81.4**
BioNLP11ID	74.6	81.5	**83.2**
BioNLP13CG	**81.8**	76.7	81.2
BioNLP13GE	69.3	73.2	**73.3**
BioNLP13PC	85.4	80.6	**86.3**
CRAFT	81.2	79.5	**83.8**
Ex-PTM	67.6	68.5	**72.7**
JNLPBA	72.1	69.6	**74.4**
LINNAEUS	86.9	83.9	**87.3**
NCBI-disease	83.9	80.2	**84.1**
Average	80.4	78.4	**82.4**

Table 3. Results comparison for different multitask models.

Datasets	**Wang et al. [35]**	**Crichton et al. [34]**	**MTM-CNN**
AnatEM	86.0	82.2	**87.0**
BC2GM	78.9	73.2	**80.8**
BC4CHEMD	**88.8**	83.0	87.4
BC5CDR	**88.1**	83.9	87.9
BioNLP09	88.1	84.2	**88.7**
BioNLP11EPI	83.2	78.9	**84.8**
BioNLP11ID	83.3	81.7	**87.7**
BioNLP13CG	82.5	78.9	**84.3**
BioNLP13GE	**79.9**	78.6	79.8
BioNLP13PC	88.5	81.9	**88.8**
CRAFT	82.9	79.6	**83.2**
Ex-PTM	80.2	74.9	**81.0**
JNLPBA	72.2	70.0	**74.1**
LINNAEUS	**88.9**	84.0	87.8
NCBI-disease	85.5	80.4	**85.7**
Average	83.8	79.7	**84.6**

4.1. Effects of Different Auxiliary Tasks

In the previous experiments, all the auxiliary tasks were the same, i.e., both auxiliary and main tasks were the same. In order to see the effect of different tasks in the MTL approach, the experiments were extended with various tasks, also considering the same BioNER task but at a different level of layers in the MTM-CNN.

In this regard, three different approaches were adopted. In the first approach (MTM-CNN$^{\bigstar}$), during the training of the MTM-CNN, two additional but different auxiliary tasks were introduced: these were GENIA-POS tagging and CoNLL chunking. In the second approach, MTM-CNN$^{\bigstar^{in}}$, the auxiliary tasks (GENIA-POS tagging and CoNLL chunking) were trained at the innermost layer. This eliminated the task-specific BiLSTM (layer denoted by a circled 3 in Figure 3) for these two auxiliary tasks. Using auxiliary tasks at the innermost layer helped the outermost layer (circled 3) to learn from a complete representation of the auxiliary tasks. The presented approach illustrated the performance improvement compared to the first approach, which therefore motivated the third approach of auxiliary tasks of this section.

In the third approach (MTM-CNN^{in}), the auxiliary tasks were the same BioNER tasks used in the MTM-CNN but this time, the auxiliary tasks were trained at the innermost layer (without task-specific BiLSTM) and the main task was used at the outermost layer (having a shared BiLSTM layer, a connection from the circled 2 to the circled 4 in Figure 3)). For instance, the MTM-CNN^{in} was trained for AnatEM, then the rest of the datasets (BC2GM, BC4CHEMD, etc.) were treated as auxiliary tasks for AnatEM and did not have any task-specific BiLSTM. More specifically, in both the second and third approaches, the auxiliary tasks did not have the task-specific BiLSTM layer, while the main tasks had the task-specific BiLSTM layer.

Table 4 shows the results of approaches using different auxiliary tasks. It can be seen that by introducing the GENIA-POS and CoNLL chunking auxiliary tasks, the results (MTM-CNN$^{\bigstar}$) improved for ten datasets against the MTM-CNN with an F1-score gain of up to 2.5%. We see that the MTM-CNN$^{\bigstar}$ improved the F1-score for CRAFT from 80.9 to 83.6. When the auxiliary tasks were used at the innermost layer (they did not have task-specific BiLSTM), it was noticed that the results of the MTM-CNN$^{\bigstar^{in}}$ improved for twelve datasets (with an increase in F1-score of up to 3%) compared to those of the proposed MTM-CNN. For CRAFT, we noticed that the F1-score increased from 80.9 to 84.1 against the MTM-CNN. Similarly, when the same BioNER auxiliary tasks were used at the innermost layer (last column) as in the proposed MTM-CNN^{in}, the results improved for nine datasets. Conclusively, the MTM-CNN was found more effective when trained with auxiliary tasks other than the same BioNER tasks.

Table 4. Results comparison of proposed multitask learning approach with different auxiliary tasks and at different levels of the main model.

Datasets	MTM-CNN	MTM-CNN$^{\bigstar}$	MTM-CNN$^{\bigstar^{in}}$	MTM-CNN^{in}
AnatEM	86.9	87.1	**87.3**	86.6
BC2GM	80.8	**81.4**	81.3	81.2
BC4CHEMD	87.3	**88.6**	88.4	87.9
BC5CDR	87.8	88.1	**88.3**	88.0
BioNLP09	88.7	88.7	88.7	**88.9**
BioNLP11EPI	84.7	84.6	**84.9**	84.5
BioNLP11ID	87.6	**88.0**	87.6	87.5
BioNLP13CG	84.2	84.4	84.5	**84.6**
BioNLP13GE	79.8	79.4	**80.0**	80.0
BioNLP13PC	88.8	88.7	**89.0**	88.7
Ex-PTM	**83.1**	81.4	81.5	81.1
CRAFT	80.9	83.6	**84.1**	83.5
JNLPBA	**74.0**	72.4	72.6	72.4
LINNAEUS	87.7	**88.9**	88.3	88.4
NCBI-disease	85.6	85.7	**86.0**	85.7
Average	84.5	84.7	**84.8**	84.6

4.2. Statistical Analysis of MTM-CNN

The performance of the models was analyzed statistically, and the Friedman test was applied to the models' outputs. The Friedman test is suitable when three or more comparisons are drawn [55,56]. Figure 7 shows that the difference between the results produced by the proposed models and their variants was statistically significant. All MTMs results were significantly better than the STM-CNN results. The result of the MTM-CNN$^{\bigstar^{in}}$ (GENIA-POS and CoNLL used at the innermost layer) was found significantly better with respect to all approaches except for the $MTM\text{-}CNN^{\bigstar}$ (GENIA-POS and CoNLL used at the outermost layer).

The output ranks of the Friedman test were considered to analyze which models were statistically superior to other model(s). Figure 8 shows the models according to their statistical ranks where the leftmost one represents the best model which decreases from left to right. It can be seen that the proposed MTM-CNN$\bigstar^{in}$ was significantly better than the rest of the approaches. Using the auxiliary tasks at the innermost layer was found the most effective, producing significantly better results with respect to most of the other approaches.

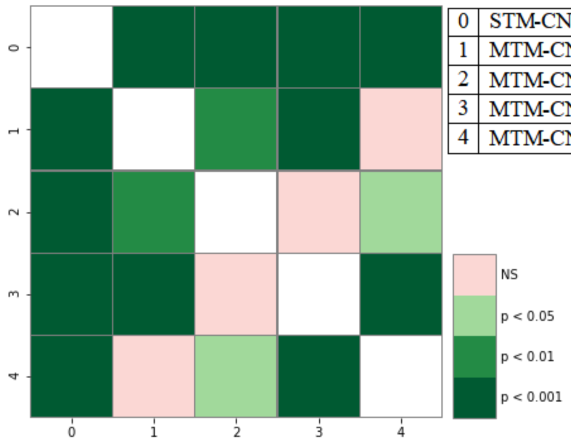


Figure 7. Post hoc pairwise analysis of Friedman's test for the MTM-CNN.

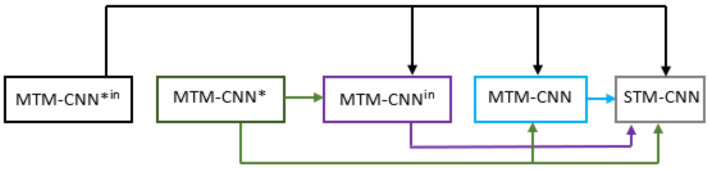


Figure 8. Graphical representation of the Friedman test for the MTM-CNN. Models were drawn according to their ranks starting with the best model from left to right.

5. Results and Discussion of MTM-CW

Table 5 shows the comparison of the MTM-CW with previous approaches [35]. A substantial improvement can be observed in the F1-score for the MTM-CW compared to these models. The MTM-CW elevated the F1-score for twelve and eleven datasets compared with Wang et al. [35] and the MTM-CNN, respectively. We found that the MTM-CW increased the F1-score to 87.1 from 83.2 for BioNLP11ID, compared with the model presented by Wang et al. [35]. To observe whether skip connections (connections from previous layers) truly contributed to the performance of the model, the skip connections (numbered 5 and 6) were dropped (Figure 4). However, we see that the MTM-CW$^{w/o}$ elevated the F1-score to 88.1 from 87.1 for BioNLP11ID when compared against the MTM-CW. The MTM-CW without skip connections (MTM-CW$^{w/o}$) made it similar to the MTM-CNN (Section 2.1) but with two shared BiLSTM layers. It can be observed that few datasets showed moderate performance increases, while for most datasets the performance dropped. This supports the intuition of proposing the MTM-CW, where propagating the information to the lower layers using skip connections positively impacted the model. Additionally, it is worth noticing that even after removing those skip connections, the MTM-CW$^{w/o}$ still yielded a better F1-score compared to state-of-the-art models. This shows that with the increasing size of the training examples, more layers of LSTM should be practiced [43]. For this reason, the proposed model showed a performance gain compared to Wang et al. [35]. Conclusively, the proposed MTM-CW and MTM-CW$^{w/o}$ showed an average performance gain of 1.3% and 1.2%, respectively, against the model proposed by Wang et al. [35].

Table 5. Multitask models comparison where CW represents character and word, respectively.

Datasets	Wang et al. [35]	MTM-CNN	MTM-CW	MTM-CW$^{w/o}$
AnatEM	86.0	86.9	**87.5**	86.9
BC2GM	78.8	80.8	**81.5**	81.2
BC4CHEMD	88.8	87.3	**89.2**	87.4
BC5CDR	88.1	87.8	**88.5**	88.1
BioNLP09	88.0	88.7	88.5	**89.3**
BioNLP11EPI	83.1	84.7	**85.3**	85.0
BioNLP11ID	83.2	87.6	87.1	**88.1**
BioNLP13CG	82.4	84.2	**84.9**	84.6
BioNLP13GE	79.8	79.8	80.9	**82.2**
BioNLP13PC	88.4	88.8	**89.1**	89.0
CRAFT	82.8	83.1	**85.2**	83.4
Ex-PTM	80.1	80.9	81.7	**82.4**
JNLPBA	72.2	**74.0**	72.1	72.0
LINNAEUS	**88.8**	87.7	88.1	88.6
NCBI-disease	85.5	**85.6**	85.0	85.1
Average	83.7	84.5	**85.0**	**84.9**

Statistical Analysis of MTM-CW

To statistically evaluate the results obtained by the proposed MTM-CW, the Friedman test was performed [56]. Figure 9 illustrates the post hoc Friedman test where p values show the significance level. We found that the difference between results produced by all the models was statistically significant. The results of the proposed model (MTM-CW) were found statistically significant. The MTM-CW was also statistically significant against the MTM-CNN and its variants except for the MTM-CNN$^{\star in}$ (GENIA-POS and CoNLL used at the innermost layer).

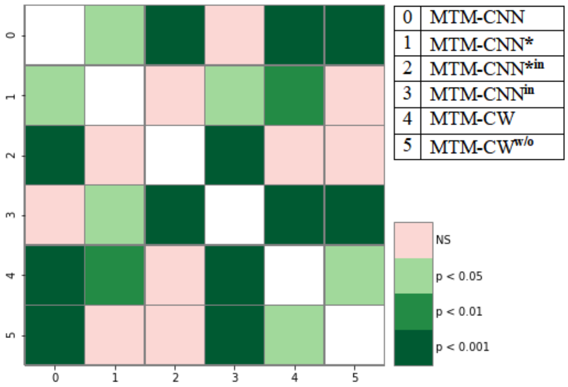


Figure 9. Post hoc pairwise analysis of Friedman's test for MTM-CNN vs. MTM-CW.

The statistical analysis was also extended with the pairwise comparison of different models to see which model was statistically better than the others. The graphical representation of the pairwise comparison is shown in Figure 10. The MTM-CW was found statistically better than the rest of the approaches on its right side. It can be seen that the MTM-CNN was statistically worse compared to the other models.

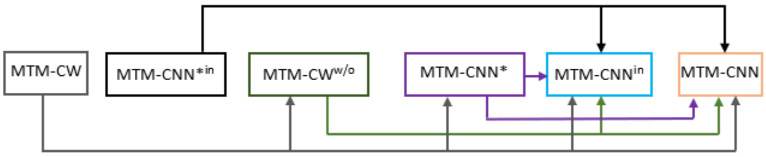


Figure 10. Graphical representation of the Friedman test for MTM-CW vs. MTM-CNN. The arrows show models that are statistically significant with each other.

6. Results and Discussion for Fine-Tuned MTM ($MTM \rightarrow STM$)

Table 6 displays the experimental results where the bold text shows the best F1-score. Note that in Table 6, the STM and MTM we indicate are the models presented in Figure 5 (Section 2.3). In this table, a performance increment for several datasets can be observed for the MTM compared to the STM, whereas a drop of F1-score is noticed for BC2GM, BC4CHEMD, BC5CDR, and CRAFT. This suggests that the MTM did not always manage to enhance the performance of the STM.

From Table 6, it can be seen that the fine-tuning ($MTM \xrightarrow{10} STM$) showed a performance increase for all datasets compared to the STM, except BC4CHEMD. The reason could be related to the lesser sparsity presented in the BC4CHEMD dataset as it is a single-entity dataset with a large number of chemical entities. The ($MTM \xrightarrow{10} STM$) model learned more prominent features in contrast to the MTL approach; when comparing it with the MTM, we saw improvements in the F1-score for all datasets except for BioNLP09, BioNLP13GE, and Ex-PTM. The results of the STM and MTM for these three datasets showed a substantial performance gain with the MTM approach. This indicated that it was difficult to learn complex features from these datasets independently in the STM and they could be learned in the MTM. The training of MTM^{10} extracted useful features for these datasets but it might have started to forget such features, or the noise presented in these datasets might have caused a performance drop.

Table 6. Results comparison of our different fine-tuned approaches of the MTM ($MTM \rightarrow STM$).

Datasets	**STM**	**MTM**	**MTM$\xrightarrow{10}$ STM**	**MTM$\xrightarrow{20}$ STM**	**MTM$\xrightarrow{cmp}$ STM**
AnatEM	86.7	87.5	87.9	**88.0**	**88.0**
BC2GM	81.7	81.6	82.1	**82.2**	82.0
BC4CHEMD	**90.4**	89.0	89.9	**90.4**	**90.4**
BC5CDR	88.5	88.4	88.8	89.0	**89.1**
BioNLP09	87.8	**89.0**	88.5	88.7	88.5
BioNLP11EPI	83.1	85.2	85.3	**85.5**	85.4
BioNLP11ID	86.3	87.5	87.6	87.8	**87.9**
BioNLP13CG	83.1	84.9	84.9	**85.2**	85.1
BioNLP13GE	76.4	**80.3**	80.1	80.1	80.2
BioNLP13PC	87.7	89.2	89.3	89.2	**89.3**
CRAFT	84.7	84.2	84.9	**85.3**	85.0
Ex-PTM	74.0	**82.1**	81.7	82.0	81.8
JNLPBA	72.2	72.8	**73.0**	72.1	71.9
LINNAEUS	87.6	88.4	88.8	88.2	**88.8**
NCBI-disease	84.9	86.2	86.2	85.9	**86.2**
Average	83.7	85.1	**85.3**	**85.3**	**85.3**

Continuing the experiments, the MTM model was trained for 20 epochs, (MTM^{20}), and fine-tuned ($MTM \xrightarrow{20} STM$) for each specific dataset. Comprehensively, we saw that among 15 datasets, JNLPBA was the only dataset for which ($MTM \xrightarrow{20} STM$) did not show an increase in F1-score compared to the STM; however, the degradation was comparable. An overall performance increment could be observed for nine datasets compared with the MTM, while for six datasets, the performance was degraded. Compared with the ($MTM \xrightarrow{10} STM$) approach, a performance improvement could be seen for ten datasets, while the performance degraded for five datasets. Comparing the F1-score with that of the MTM, the ($MTM \xrightarrow{20} STM$) was unable to show any increase in the F1-score for the protein datasets (BioNLP09, BioNLP13GE, Ex-PTM) while leveraging the performance for BioNLP11PEI. The same performance decrease behavior was also noticed for JNLPBA which comprises huge protein examples as well (see Table A1 in Appendix A). The performance degradation for the LINNAEUS dataset might be due to the insufficient number of examples for the entity class. This suggests that these datasets were more feasible with the MTL approach.

Note also that ($MTM\xrightarrow{20}STM$) had improved results for BC4CHEMD compared to those of the STM and MTM.

In a third experiment, a fully trained MTM (MTM^{cmp}) was fine-tuned for each dataset. It was observed that ($MTM\xrightarrow{cmp}STM$) showed a performance gain for 11 datasets compared to the MTM. When comparing the results with the STM, we found that JNLPBA was the only dataset among others, which was unable to improve the F1-score. Furthermore, comparing it with ($MTM\xrightarrow{10}STM$), the method illustrated a performance improvement for 13 datasets whereas comparing it with ($MTM\xrightarrow{20}STM$), a performance gain was noticed for 7 datasets. We further noticed that similarly to ($MTM\xrightarrow{20}STM$) and ($MTM\xrightarrow{10}STM$), the ($MTM\xrightarrow{cmp}STM$) also showed a performance drop for the protein datasets (BioNLP09, BioNLP13GE, Ex-PTM, and JNLPBA) when evaluated against the MTM. However, unlike in ($MTM\xrightarrow{20}STM$), a performance improvement could be seen for the LINNAEUS and NCBI datasets compared to that of the MTM. The ($MTM\xrightarrow{cmp}STM$) achieved the best F1-score for BC4CHEMD, for which the MTM performed worse with respect to the STM. Comparing it with the ($MTM\xrightarrow{10}STM$) method, JNLPBA was the only dataset for which ($MTM\xrightarrow{cmp}STM$) did not show any performance gain. Additionally, evaluating it with the ($MTM\xrightarrow{20}STM$) model, a performance drop was observed for BC2GM, BioNLP09, BioNLP11EPI, BioNLP13CG, CRAFT, Ex-PTM, and JNLPBA whereas for AnatEM and BC4CHEMD, the difference was negligible. We speculate that the reason for the drop in F1-score was related to the shared layer in ($MTM\xrightarrow{cmp}STM$) which had learned features that became more task-specific and therefore favoring only specific datasets.

The results of the proposed method were also compared with the other MTMs presented in this work and are shown in Table 7, where it can be observed that the proposed fine-tuned models ($MTM\rightarrow STM$) had increased in performance compared to the other methods. The proposed MTM-CW (see Section 2.2) had an increase in performance for a single dataset while the proposed MTM-CNN (see Section 2.1) showed a performance gain for only two datasets compared to all fine-tuned models.

Table 7. Results comparison of $MTM\rightarrow STM$ with state-of-the-art MTMs.

Datasets	**MTM-CNN**	**MTM-CW**	**MTM$\xrightarrow{10}$ STM**	**MTM$\xrightarrow{20}$ STM**	**MTM$\xrightarrow{cmp}$ STM**
AnatEM	86.9	87.5	87.9	**88.0**	**88.0**
BC2GM	80.8	81.5	82.1	**82.2**	82.0
BC4CHEMD	87.3	89.2	89.9	**90.4**	90.4
BC5CDR	87.8	88.5	88.8	89.0	**89.1**
BioNLP09	**88.7**	88.5	88.5	88.7	88.5
BioNLP11EPI	84.7	85.3	85.3	**85.5**	85.4
BioNLP11ID	87.6	87.1	87.6	87.8	**87.9**
BioNLP13CG	84.2	84.9	84.9	**85.2**	85.1
BioNLP13GE	79.8	**80.9**	80.1	80.1	80.2
BioNLP13PC	88.8	89.1	89.3	89.2	**89.3**
CRAFT	83.1	85.2	84.9	**85.3**	85.0
ExPTM	80.9	81.7	81.7	**82.0**	81.8
JNLPBA	**74.0**	72.1	73.0	72.1	71.9
LINNAEUS	87.7	88.1	88.8	88.2	**88.8**
NCBI	85.6	85.0	86.2	85.9	**86.2**
Average	84.5	85.0	**85.3**	**85.3**	**85.3**

Statistical Analysis of $MTM\rightarrow STM$

The results were evaluated using Friedman's statistical test as presented in Figure 11. The figure shows that all the variants of $MTM\rightarrow STM$ produced statistically significant results against their STM and MTM counterparts. The results were statistically significant with the previously mentioned approaches, the MTM-CNN and MTM-CW. Nevertheless, the $MTM\rightarrow STM$ did not generate significant results with each other.

The models are also shown according to their Friedman statistical ranks and are given in Figure 12. The figure indicates that all the fine-tuned models produced higher Friedman statistical scores. The $MTM\xrightarrow{cmp}STM$ generated the highest statistical rank indicating that this model covered a wider range of features, benefiting datasets suitable for both the MTL and STL approaches. The $MTM\xrightarrow{10}STM$ produced the lowest ranks among the fine-tuned models, which showed that the base model (MTM^{10}) for that experiment did not learn distinct features and therefore, using that model as a starting point for the STM did not benefit the model.

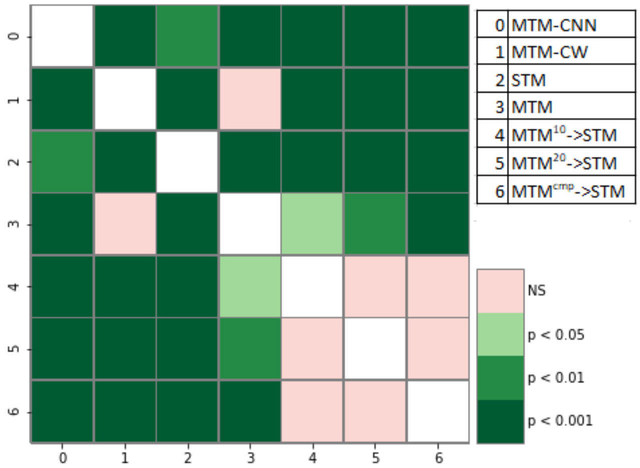


Figure 11. Post hoc Friedman test output for $MTM{\rightarrow}STM$.

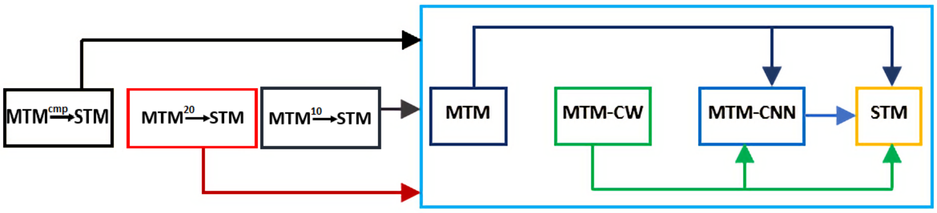


Figure 12. Graphical representation of Friedman test ranks produced by $MTM{\rightarrow}STM$. Models are shown according to their ranks starting with the best model from left to right.

7. Results and Discussion for ELMo

The comparison of different models trained with various approaches of ELMo contextual embeddings is shown in Table 8. It can be seen that the STM$_{EL}$ (the STM trained with ELMo embeddings) performed better compared with the MTM$_{EL}$ (the MTM trained with ELMo embeddings) with average F1-score gain of 0.2%. It is observed that the STM$_{EL}$ generated an F1-score of 91.3 for BC4CHEMD which dropped to 88.6 when the MTM$_{EL}$ was used. It can be hypothesized that the performance drop from the STM to the MTM can be due to the entity ambiguity and prevalence of unseen entity (the probability of confronting a word never seen during training). We speculate that the performance of the MTM$_{EL}$ dropped as it could not overcome the ambiguity problem that occurred during the MTL approach. Additionally, the MTL approach also performed implicit data augmentation which decreased the prevalence of the words (ELMo saw huge numbers of words in the MTL approach). More specifically, the performance of the contextual embeddings dropped for BioNER using MTL as it increased entity ambiguity and decreased prevalence. However, when performing the transfer learning approach (fine-tuning the MTM$_{EL}$ for each specific dataset), the performance of the STM ($MTM_{EL}^{cmp}{\rightarrow}STM_{EL}$) increased with an average F1-score gain of 0.6%, as again, we can assume that the entity ambiguity decreased and the prevalence increased for contextual embeddings during the STL approach. It is worth noticing that the $MTM_{EL}^{cmp}{\rightarrow}STM_{EL}$ increased the F1-score for BC4CHEMD to 91.1 from 88.6 obtained by the MTM$_{EL}$.

We emphasize that the use of contextual embedding (ELMo) still enhanced the performance of the model in the MTL approach which we can observe in Table 9. The results of different earlier proposed models are compared with the ELMo-based MTM (MTM$_{EL}$). The fourth column (MTM) shows the same as the MTM$_{EL}$ column but without the integration of the ELMo embeddings. We can instantly discover that by integrating the contextual embedding, the performance of the MTM increased substantially. Furthermore, the results of the MTM$_{EL}$ showed a noticeable increase compared to the other MTMs (the MTM-CNN and MTM-CW proposed earlier).

Table 8. Results comparison of different models trained on ELMo embeddings.

Datasets	**STM$_{EL}$**	**MTM$_{EL}$**	**MTM$_{EL}^{cmp}\rightarrow$ STM$_{EL}$**
AnatEM	**89.5**	88.9	**89.5**
BC2GM	**83.3**	82.3	83.1
BC4CHEMD	**91.3**	88.6	91.1
BC5CDR	**90.1**	89.3	90.0
BioNLP09	89.2	**90.1**	89.9
BioNLP11EPI	87.5	86.9	**87.7**
BioNLP11ID	87.7	87.8	**88.0**
BioNLP13CG	86.1	86.4	**87.1**
BioNLP13GE	80.8	81.7	**82.1**
BioNLP13PC	89.9	90.1	**90.5**
CRAFT	86.6	84.7	**86.9**
ExPTM	81.0	83.2	**83.8**
JNLPBA	72.9	**73.3**	72.8
LINNAEUS	88.4	**88.5**	88.2
NCBI	86.6	86.6	**86.7**
Average	86.1	85.9	**86.5**

Table 9. Results comparison of ELMo with SOTA MTMs.

Datasets	**MTM-CNN**	**MTM-CW**	**MTM**	**MTM$_{EL}$**
AnatEM	86.9	87.5	87.5	**88.9**
BC2GM	80.8	81.5	81.6	**82.3**
BC4CHEMD	87.3	**89.2**	89.0	88.6
BC5CDR	87.8	88.5	88.4	**89.3**
BioNLP09	88.7	88.5	89.0	**90.1**
BioNLP11EPI	84.7	85.3	85.2	**86.9**
BioNLP11ID	87.6	87.1	87.5	**87.8**
BioNLP13CG	84.2	84.9	84.9	**86.4**
BioNLP13GE	79.8	80.9	80.3	**81.7**
BioNLP13PC	88.8	89.1	89.2	**90.1**
CRAFT	83.1	**85.2**	84.2	84.7
ExPTM	80.9	81.7	82.1	**83.2**
JNLPBA	**74.0**	72.1	72.8	73.3
LINNAEUS	87.7	88.1	88.4	88.5
NCBI	85.6	85.0	86.2	**86.6**
Average	84.5	85.0	85.1	**85.9**

In Table 10, we also compare the fine-tuning approach for ELMo with our earlier fine-tuning approach (Section 6). We see that $MTM_{EL}^{cmp}\rightarrow STM_{EL}$ achieved distinguishable performance compared with our previous proposed approach. The $MTM_{EL}^{cmp}\rightarrow STM_{EL}$ increased the F1-score for 13 out of 15 datasets.

Table 10. Results comparison of transfer Learning for different models with and without ELMo.

Datasets	**MTM$^{10}\rightarrow$ STM**	**MTM$^{20}\rightarrow$ STM**	**MTM$^{cmp}\rightarrow$ STM**	**MTM$_{EL}^{cmp}\rightarrow$ STM$_{EL}$**
AnatEM	87.9	88.0	88.0	**89.5**
BC2GM	82.1	82.2	82.0	**83.1**
BC4CHEMD	89.9	90.4	90.4	**91.1**
BC5CDR	88.8	89.0	89.1	**90.0**
BioNLP09	88.5	88.7	88.5	**89.9**
BioNLP11EPI	85.3	85.5	85.4	**87.7**
BioNLP11ID	87.6	87.8	87.9	**88.0**
BioNLP13CG	84.9	85.2	85.1	**87.1**
BioNLP13GE	80.1	80.1	80.2	**82.1**
BioNLP13PC	89.3	89.2	89.3	**90.5**
CRAFT	84.9	85.3	85.0	**86.9**
ExPTM	81.7	82.0	81.8	**83.8**
JNLPBA	**73.0**	72.1	71.9	72.8
LINNAEUS	**88.8**	88.2	**88.8**	88.2
NCBI	86.2	85.9	86.2	**86.7**
Average	85.3	85.3	85.3	**86.5**

Statistical Analysis of ELMo

The statistical analysis of the results produced by ELMo and other models was performed using the Friedman test. The post hoc pairwise comparison is shown in Figure 13. The difference between the results produced by the STM$_{EL}$ and the MTM$_{EL}$ were statistically not significant but their results were statistically significant with respect to the results produced by the rest of the models. In Table 8, it can be observed that the MTM$_{EL}$ did not gain much improvement compared to the STM$_{EL}$, and for that reason, the difference of their results were statistically not significant. It is still worth noting that the results of the STM$_{EL}$ and the MTM$_{EL}$ were statistically significant compared to those of $MTM_{EL}^{cmp}\rightarrow STM_{EL}$. The models with ELMo embeddings also produced statistically significant results compared to our previously proposed approaches.

Further analyzing the ranks generated using the Friedman test, we observed that $MTM_{EL}^{cmp}\rightarrow STM_{EL}$ produced a higher statistical rank compared to the rest of the models as illustrated in Figure 14. It also showed that the STM$_{EL}$ attained the second highest statistical value, based on the Friedman test, followed by the MTM$_{EL}$.

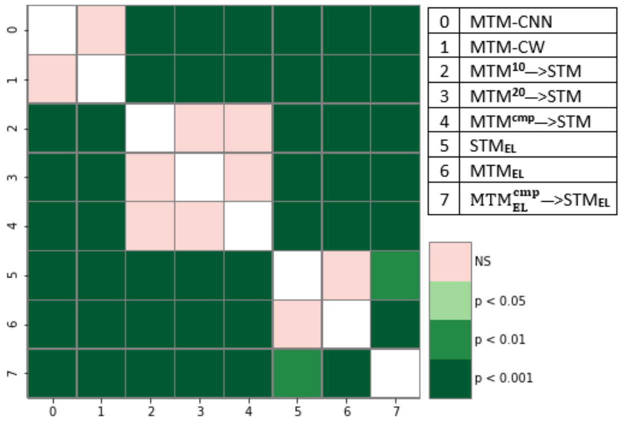


Figure 13. Post hoc Friedman test comparison of ELMo.

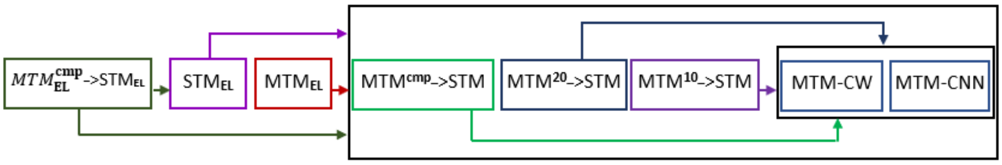


Figure 14. Graphical representation of the Friedman test ranks of ELMo. Models are shown according to their ranks starting with the best model from left to right.

8. Conclusions

This work presented several knowledge transfer techniques to enhance the performance of the BioNER task. The first technique presented followed the multitask learning (MTL) approach. Training different models simultaneously encouraged multiple tasks to transfer their knowledge implicitly using a shared layer(s). Thus, the size of the available data accumulated into the multitask model (MTM). Following the results given in this paper, it was found that the MTM indeed improved the performance of the BioNER task over that of a single-task model (STM). Our proposed MTM-CNN depicted an absolute increase in average F1-score of up to 0.8% and 4% compared to the model proposed by Wang et al. [35] and that proposed by Crichton et al. [34], respectively. Similarly, the proposed MTM-CW showed an average performance gain of 1.3% and 5.3%, respectively, against the model proposed in [34,35].

The second proposed technique was based on transfer learning. Transfer learning was utilized by applying pretrained MTMs trained for different epochs. These MTMs were used as the base model, which was then further fine-tuned for a specific task. The purpose was to get a generalized base model which was then specialized for a specific dataset. The presented results illustrated a performance gain compared to the MTL approach and other state-of-the-art approaches including the STM. Comprehensively, the proposed methodology showed an average F1-score gain of 1.6%. This work also used ELMo, which provides contextual word embedding, to see its impact on the results. The results showed that ELMo supported the learning ability of the model.

As a future work, the MTL approach will be extended to train it by combining different auxiliary tasks along with BioNER. This can include biomedical relation extraction, biomedical question-answering tasks, etc. The rationale is to explore and understand the underlying relationship between auxiliary tasks and main tasks. This can also help to remove those auxiliary tasks that compromise the performance of the MTM. In this case, the auxiliary task may distort the main task during MTM training. We also plan to use other available word embeddings for our work to find out the best embeddings for the STM and for the MTM. Another future research direction is related to the use of Transformer-based models pretrained on biomedical documents such as BlueBERT [57] and BioBERT [58]. We will explore and analyze the effectiveness of available transformers considering all 15 datasets for the STM and MTM.

Author Contributions: Conceptualization and supervision, T.M., I.S., A.G., and A.L.; methodology, T.M. and I.S.; software and investigation, T.M.; validation, T.M., L.P.; formal analysis, L.P. and A.L.; writing—original draft preparation, T.M., I.S., and A.L.; writing—review and editing, T.M., I.S., L.P., A.L., and A.G.; funding acquisition, I.S. and A.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by MIUR “Fondo Dipartimenti di Eccellenza 2018–2022” of the DII Department at the University of Brescia, Italy. The IBM Power Systems Academic Initiative substantially contributed to the experimental analysis.

Data Availability Statement: The datasets can be found at the following link https://github.com/cambridgeltl/MTL-Bioinformatics-2016, accessed on 1 February 2023.

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

Appendix A. Datasets

There were 15 datasets used in the different experiments of this work. The bioentities in these datasets were chemical, species, cell, gene/protein, cell component, and disease. A brief description of these datasets is given in Table A1. It is important to note that performing human annotation on bioentities is more difficult than normal text data. The biomedical concepts can be annotated differently depending on the background of the annotators. Considering the annotation of biomedical entities, i.e., genes, proteins, and RNA, the human interannotator agreement was 70% for these biomedical concepts [59]. This work used the preprocessed form of these datasets where the sentence was represented in the CoNLL (https://www.clips.uantwerpen.be/conll2003/ner/, accessed on 1 February 2023) column-based format [60]. Each word of the sentence was separated by a newline and the first column represented the word token of the sentence, and the second column was the label for such token. The sentences were separated by an empty line. The datasets used in this work contained separate training, development, and test sets. The name of the entities and their distribution in the dataset (percentagewise) are reported in Table A2. The values in the table represent the percentage of an individual entity (the O-outside tag is not included) contributing to the train/dev/test file.

Appendix A.1. AnatEM

The Anatomical Entity Mention (AnatEM) corpus [61] is an extended version of the original Anatomical Entity Mention (AnEM) corpus that also combines the Multi-level Event Extraction (MLEE) corpus. The AnEM comprises 500 randomly chosen PubMed abstracts and full text that are annotated for anatomical entity mentions. On the other hand, MLEE comprises 262 PubMed abstracts on cancer's molecular mechanisms linked to angiogenesis. The AnatEM comprises these two corpora and is further extended by 100 other documents following the AnEM document selection procedures. Similarly, 350 additional documents were added related to the cancer topics. The selection of these additional documents followed the same process implemented in the MLEE. The final version of AnatEM, therefore, consists of a total of 1212 documents.

Appendix A.2. BC2GM

The BC2GM (BioCreative||Gene Mention) corpus contains a total of 20,000 sentences coming from abstracts of biomedical publications [62]. The BC2GM covers genes, proteins, and other similar entities. However, they are all combined into a single entity class, i.e., gene.

Appendix A.3. BC4CHEMD

The BC4CHEMD, BioCreative IV Chemical and Drug corpus consists of 10,000 abstracts annotated for a single chemical entity containing chemical and drug names [63].

Appendix A.4. BC5CDR

The BioCreative V Chemical Disease Relation (BC5CDR) [64] comprises 1500 PubMed articles, of which 1400 articles were selected from 150,000 chemical–disease interactions that were annotated during the Comparative Toxicogenomics Database-Pfizer (CTD-Pfizer) process. The rest of the 100 articles were newly curated and were included in the test set.

Table A1. Datasets description [34].

Dataset	**Contents**	**Entity Counts**
AnatEM	Anatomy NE	13,701
BC2GM	Gene/protein NE	24,583
BC4CHEMD	Chemical NE	84,310
BC5CDR	Chemical, disease NEs	Chemical: 15,935; disease: 12,852
BioNLP09	Gene/protein NE	14,963
BioNLP11EPI	Gene/protein NE	15,811
BioNLP11ID	4 NEs	Gene/protein: 6551; organism: 3471 chemical: 973; regulon-operon: 87

BioNLP13CG	16 NEs	Gene/protein: 7908; cell: 3492;
		chemical: 2270; organism: 1715; tissue: 587;
		multitissue structure: 857;
		amino acid: 135; cellular component: 569;
		organism substance: 283; organ: 421;
		pathological formation: 228;
		immaterial anatomical entity: 102;
		organism subdivision: 98;
		anatomical system: 41; cancer: 2582;
		developing anatomical structure: 35

BioNLP13GE	Gene/protein NE	12,057
BioNLP13PC	4 NEs	Gene/protein: 10,891; chemical: 2487; complex: 1502; cellular component: 1013

CRAFT	6 NEs	SO: 18,974; gene/protein: 16,064;
		cl: 5495; taxonomy: 6868; chemical: 6053;
		GO-CC: 4180

Ex-PTM	Gene/protein NE	4698
JNLPBA	5 NEs	Gene/protein: 35,336; DNA: 10,589; cell type: 8639l; cell line: 4330; RNA: 1069
LINNAEUS	Species NE	4263
NCBI-Disease	Disease NE	6881

Table A2. Entities percentage distribution in training+development and test dataset.

Dataset	Entities Name	Train+Dev Set	Test Set
AnatEM	Anatomy	7.241	7.865
BC2GM	Gene	10.505	10.526
BC4CHEMD	Chemical	7.284	7.162
BC5CDR	Chemical	6.061	5.622
	Disease	5.971	5.740
BioNLP09	Protein	9.573	10.274
BioNLP11EPI	Protein	7.662	7.840
BioNLP11ID	Regulon-operon	0.047	0.131
	Chemical	7.036	0.700
	Organism	4.421	3.801
	Protein	4.575	4.134
BioNLP13CG	Gene_or_gene_product	9.975	9.236
	Cancer	2.423	2.896
	Amino_acid	0.088	0.123
	Simple_Chemical	2.631	2.550
	Organism	1.462	1.209
	Cell	4.464	3.987
	Tissue	0.579	0.559
	Organ	0.262	0.328
	Multi_tissue_structure	0.818	0.881
	Cellular_component	0.479	0.472
	Pathological_formation	0.191	0.241
	Immaterial_anatomical	0.075	0.078
	Organism_subdivision	0.060	0.091
	Anatomical_system	0.036	0.049
	Developing_anatomical_structure	0.018	0.040
	Organism_substance	0.197	0.238
BioNLP13GE	Protein	8.100	7.781
BioNLP13PC	Gene_or_gene_product	13.447	13.268
	Simple_chemical	3.272	3.571
	Complex	3.190	3.232
	Cellular_component	0.889	0.879
CRAFT	SO	4.330	3.860
	GGP	4.240	4.320
	Taxon	1.280	1.160
	CHEBI	1.210	1.250
	CL	1.330	1.190
	GO	0.960	0.990
Ex-PTM	Protein	7.967	7.616
JNLPBA	Protein	11.190	9.740
	DNA	5.130	2.810
	Cell_type	3.140	4.860
	Cell_line	2.780	1.470
	RNA	0.504	0.300
LINNAEUS	Species	1.153	1.350
NCBI-Disease	Disease	8.220	8.356

Appendix A.5. BioNLP09

The BioNLP09 is a 2009 shared-event task to extract different events among different classes [65]. The named entity was performed via the GENIA event corpus to facilitate the event extraction task. The 10,000 sentences in the corpus were annotated for protein and related entities into a single entity class, protein.

Appendix A.6. BioNLP 2011 Shared Task

The BioNLP 2009 shared task was extended and presented again in 2011. The BioNLP 2011 shared task covered various tasks including infection diseases (ID), epigenetics and post-translational modifications (EPI), and exhaustive post-translational modifications (Ex-PTM). The BioNLP11EPI events targeted the statements covering modifications in protein and DNA, and their reverse reactions as well, covering 1200 abstracts [66]. Ex-PTM covered more post-translational modifications in the protein-related literature, databases, and ontologies for a total of 360 PubMed abstracts [67]. The BioNLP11ID task enclosed the biomolecular mechanisms of infections that comprised 30 full articles [68].

Appendix A.7. BioNLP 2013 Shared Task

The BioNLP 2013 shared task datasets, Cancer Genetics (BioNLP13CG), GENIA Event Extraction (BioNLP13GE), and Pathway Curation (BioNLP13PC) were three tasks out of six tasks in total [69]. The BioNLP13CG task aimed to extract the information associated with cancer, e.g., cellular, tissue, etc. BioNLP13CG contains 600 abstracts from PubMed and is annotated for more than 17,000 events [70], while the BioNLP13GE dataset consists of 34 full articles gathered from PubMed Central [71,72]. The BioNLP13PC dataset was annotated for about 16,000 events and contains 525 PubMed abstracts [73,74] that were collected, covering specific pathway reactions based on the pathway models from BioModels and Pathway DB [75].

Appendix A.8. CRAFT

The Colorado Richly Annotated Full Text (CRAFT) corpus contains 67 full-text articles from PubMed Central Open Access Subset, which are manually annotated [76]. These articles accumulate over 21,000 sentences, over 560,000 tokens, and approximately 100,000 concept annotations that contain different biomedical ontologies.

Appendix A.9. JNLPBA

The JNLPBA corpus was developed for a joint workshop on NLP in Biomedicine and its Applications, which comprised 2000 abstracts in the training set, while 404 abstracts in the test set made approximately 22,400 sentences. JNLPBA was developed from the GENIA corpus; however, unlike the GENIA corpus that consists of 36 classes, the JNLPBA only includes 5 classes [77].

Appendix A.10. LINNAEUS

The LINNAEUS corpus contains 100 full-text papers, selected randomly from the PMC open access set [78]. The entity mentions presented in the corpus were annotated manually and normalized according to the NCBI taxonomy. The corpus contains species mentions; however, 72% of these mentions do not contain direct species information, e.g., patients, child, etc.

Appendix A.11. NCBI-Disease

The NCBI-disease corpus has annotated disease mentions from 793 PubMed abstracts [79]. The corpus consists of 790 unique disease mentions; 698 from MeSH (698) and 92 from OMIM. Furthermore, 91% of the unique concepts are single disease concepts, while the rest contain a combination of concepts.

References

1. Mehmood, T.; Gerevini, A.E.; Lavelli, A.; Serina, I. Combining Multi-task Learning with Transfer Learning for Biomedical Named Entity Recognition. In Proceedings of the Knowledge-Based and Intelligent Information & Engineering Systems: 24th International Conference KES-2020, Virtual Event, 16–18 September 2020; Volume 176, pp. 848–857. [CrossRef]
2. Mehmood, T.; Gerevini, A.; Lavelli, A.; Serina, I. Leveraging Multi-task Learning for Biomedical Named Entity Recognition. In Proceedings of the AI*IA 2019—Advances in Artificial Intelligence—XVIIIth International Conference of the Italian Association for Artificial Intelligence, Rende, Italy, 19–22 November 2019; Volume 11946, pp. 431–444. [CrossRef]
3. Mehmood, T.; Gerevini, A.; Lavelli, A.; Serina, I. Multi-task Learning Applied to Biomedical Named Entity Recognition Task. In Proceedings of the Sixth Italian Conference on Computational Linguistics, Bari, Italy, 13–15 November 2019; Volume 2481.
4. Xu, M.; Jiang, H.; Watcharawittayakul, S. A Local Detection Approach for Named Entity Recognition and Mention Detection. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, BC, Canada, 30 July–4 August 2017; Volume 1, pp. 1237–1247. [CrossRef]

5. Lin, Y.F.; Tsai, T.H.; Chou, W.C.; Wu, K.P.; Sung, T.Y.; Hsu, W.L. A maximum entropy approach to biomedical named entity recognition. In Proceedings of the 4th International Conference on Data Mining in Bioinformatics, Seattle, WA, USA, 22 August 2004; pp. 56–61.
6. Settles, B. Biomedical named entity recognition using conditional random fields and rich feature sets. In Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP), Geneva, Switzerland, 28–29 August 2004; pp. 107–110.
7. 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.
8. Song, H.J.; Jo, B.C.; Park, C.Y.; Kim, J.D.; Kim, Y.S. Comparison of named entity recognition methodologies in biomedical documents. *Biomed. Eng. Online* **2018**, *17*, 158. [CrossRef]
9. Ciresan, D.C.; Meier, U.; Gambardella, L.M.; Schmidhuber, J. Convolutional Neural Network Committees for Handwritten Character Classification. In Proceedings of the 2011 International Conference on Document Analysis and Recognition, ICDAR 2011, Beijing, China, 18–21 September 2011; pp. 1135–1139. [CrossRef]
10. Deng, L.; Hinton, G.E.; Kingsbury, B. New types of deep neural network learning for speech recognition and related applications: an overview. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, 26–31 May 2013; pp. 8599–8603. [CrossRef]
11. Ramsundar, B.; Kearnes, S.M.; Riley, P.; Webster, D.; Konerding, D.E.; Pande, V.S. Massively Multitask Networks for Drug Discovery. *arXiv* **2015**, arXiv:1502.02072.
12. Mehmood, T.; Serina, I.; Lavelli, A.; Gerevini, A. Knowledge Distillation Techniques for Biomedical Named Entity Recognition. In Proceedings of the 4th Workshop on Natural Language for Artificial Intelligence (NL4AI 2020) Co-Located with the 19th International Conference of the Italian Association for Artificial Intelligence (AI*IA 2020), Anywhere, 25–27 November 2020; Volume 2735, pp. 141–156.
13. Caruana, R. Multitask learning. *Mach. Learn.* **1997**, *28*, 41–75. [CrossRef]
14. Mitchell, T.M. *The Need for Biases in Learning Generalizations*; (Rutgers Computer Science Tech. Rept. CBM-TR-117); Rutgers University: New Brunswick, NJ, USA, 1980.
15. Mehmood, T.; Lavelli, A.; Serina, I.; Gerevini, A. Knowledge Distillation with Teacher Multi-task Model for Biomedical Named Entity Recognition. In Proceedings of the Innovation in Medicine and Healthcare: Proceedings of 9th KES-InMed, Virtual Event, 14–16 June 2021; pp. 29–40.
16. Girshick, R.B. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, 7–13 December 2015; pp. 1440–1448. [CrossRef]
17. Collobert, R.; Weston, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the Twenty-Fifth International Conference on Machine Learning (ICML 2008), Helsinki, Finland, 5–9 June 2008; Volume 307, pp. 160–167. [CrossRef]
18. Bollmann, M.; Søgaard, A. Improving historical spelling normalization with bi-directional LSTMs and multi-task learning. *arXiv* **2016**, arXiv:1610.07844.
19. Peng, N.; Dredze, M. Multi-task multi-domain representation learning for sequence tagging. *arXiv* **2016**, arXiv:1608.02689.
20. Plank, B.; Søgaard, A.; Goldberg, Y. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. *arXiv* **2016**, arXiv:1604.05529.
21. Yang, Z.; Salakhutdinov, R.; Cohen, W. Multi-task cross-lingual sequence tagging from scratch. *arXiv* **2016**, arXiv:1603.06270.
22. Zhang, Y.; Weiss, D. Stack-propagation: Improved Representation Learning for Syntax. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, Berlin, Germany, 7–12 August 2016; Volume 1. [CrossRef]
23. Johansson, R. Training parsers on incompatible treebanks. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Atlanta, GA, USA, 9–14 June 2013; pp. 127–137.
24. Søgaard, A.; Goldberg, Y. Deep multi-task learning with low level tasks supervised at lower layers. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, Berlin, Germany, 7–12 August 2016; Volume 2. [CrossRef]
25. Hashimoto, K.; Xiong, C.; Tsuruoka, Y.; Socher, R. A Joint Many-Task Model: Growing a Neural Network for Multiple NLP Tasks. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 9–11 September 2017; pp. 1923–1933. [CrossRef]
26. Luong, M.; Le, Q.V.; Sutskever, I.; Vinyals, O.; Kaiser, L. Multi-task Sequence to Sequence Learning. In Proceedings of the 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, 2–4 May 2016.
27. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? In Proceedings of the Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, Montreal, QC, Canada, 8–13 December 2014; pp. 3320–3328.
28. Savini, E.; Caragea, C. Intermediate-Task Transfer Learning with BERT for Sarcasm Detection. *Mathematics* **2022**, *10*, 844. [CrossRef]
29. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative Pre-Training. Available online: https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf (accessed on 1 February 2023).

30. Oquab, M.; Bottou, L.; Laptev, I.; Sivic, J. Learning and Transferring Mid-level Image Representations Using Convolutional Neural Networks. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, 23–28 June 2014; pp. 1717–1724. [CrossRef]
31. Al-Stouhi, S.; Reddy, C.K. Transfer learning for class imbalance problems with inadequate data. *Knowl. Inf. Syst.* **2016**, *48*, 201–228. [CrossRef]
32. Yang, J.; Zhang, Y.; Dong, F. Neural Word Segmentation with Rich Pretraining. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, BC, Canada, 30 July–4 August 2017; Volume 1, pp. 839–849.
33. Zoph, B.; Yuret, D.; May, J.; Knight, K. Transfer Learning for Low-Resource Neural Machine Translation. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, TX, USA, 1–4 November 2016; pp. 1568–1575.
34. Crichton, G.; Pyysalo, S.; Chiu, B.; Korhonen, A. A neural network multi-task learning approach to biomedical named entity recognition. *BMC Bioinform.* **2017**, *18*, 368. [CrossRef]
35. Wang, X.; Zhang, Y.; Ren, X.; Zhang, Y.; Zitnik, M.; Shang, J.; Langlotz, C.; Han, J. Cross-type biomedical named entity recognition with deep multi-task learning. *Bioinformatics* **2019**, *35*, 1745–1752. [CrossRef]
36. Dugas, F.; Nichols, E. DeepNNNER: Applying BLSTM-CNNs and Extended Lexicons to Named Entity Recognition in Tweets. In Proceedings of the 2nd Workshop on Noisy User-Generated Text (WNUT), Osaka, Japan, 11 December 2016; pp. 178–187.
37. Segura-Bedmar, I.; Suárez-Paniagua, V.; Martínez, P. Exploring Word Embedding for Drug Name Recognition. In Proceedings of the Sixth International Workshop on Health Text Mining and Information Analysis, Louhi@EMNLP 2015, Lisbon, Portugal, 17 September 2015; pp. 64–72. [CrossRef]
38. Huang, Z.; Xu, W.; Yu, K. Bidirectional LSTM-CRF Models for Sequence Tagging. *arXiv* **2015**, arXiv:1508.01991.
39. Limsopatham, N.; Collier, N. Learning Orthographic Features in Bi-directional LSTM for Biomedical Named Entity Recognition. In Proceedings of the Fifth Workshop on Building and Evaluating Resources for Biomedical Text Mining, BioTxtM@COLING 2016, Osaka, Japan, 12 December 2016; pp. 10–19.
40. dos Santos, C.; Guimaraes, V.; Niterói, R.; de Janeiro, R. Boosting Named Entity Recognition with Neural Character Embeddings. In Proceedings of the NEWS 2015 The Fifth Named Entities Workshop, Beijing, China, 31 July 2015; p. 25.
41. Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* **2011**, *12*, 2493–2537.
42. Yang, J.; Liang, S.; Zhang, Y. Design Challenges and Misconceptions in Neural Sequence Labeling. In Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, NW, USA, 20–26 August 2018; pp. 3879–3889.
43. Li, J.; Liu, C.; Gong, Y. Layer Trajectory LSTM. In Proceedings of the Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2–6 September 2018; pp. 1768–1772. . 2018-1485. [CrossRef]
44. Hattori, M. A biologically inspired dual-network memory model for reduction of catastrophic forgetting. *Neurocomputing* **2014**, *134*, 262–268. [CrossRef]
45. Ramasesh, V.V.; Dyer, E.; Raghu, M. Anatomy of Catastrophic Forgetting: Hidden Representations and Task Semantics. In Proceedings of the 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, 3–7 May 2021.
46. French, R.M. Catastrophic forgetting in connectionist networks. *Trends Cogn. Sci.* **1999**, *3*, 128–135. [CrossRef]
47. Ma, X.; Hovy, E.H. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, Berlin, Germany, 7–12 August 2016; Volume 1. [CrossRef]
48. Sugianto, N.; Tjondronegoro, D.; Sorwar, G.; Chakraborty, P.; Yuwono, E.I. Continuous Learning without Forgetting for Person Re-Identification. In Proceedings of the 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Taipei, Taiwan, 18–21 September 2019; pp. 1–8.
49. Mikolov, T.; Grave, E.; Bojanowski, P.; Puhrsch, C.; Joulin, A. Advances in Pre-Training Distributed Word Representations. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, 7–12 May 2018.
50. Pennington, J.; Socher, R.; Manning, C. Glove: Global Vectors for Word Representation. In Proceedings of the Conference On Empirical Methods In Natural Language Processing (EMNLP), Doha, Qatar, 26–28 October 2014; pp. 1532–1543. [CrossRef]
51. Zhai, Z.; Nguyen, D.Q.; Akhondi, S.A.; Thorne, C.; Druckenbrodt, C.; Cohn, T.; Gregory, M.; Verspoor, K. Improving Chemical Named Entity Recognition in Patents with Contextualized Word Embeddings. In Proceedings of the 18th BioNLP Workshop and Shared Task, BioNLP@ACL 2019, Florence, Italy, 1 August 2019; pp. 328–338. [CrossRef]
52. Lester, B. iobes: A Library for Span-Level Processing. *arXiv* **2020**, arXiv:2010.04373.
53. Yu, J.; Bohnet, B.; Poesio, M. Named Entity Recognition as Dependency Parsing. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, 5–10 July 2020; pp. 6470–6476. [CrossRef]
54. Giorgi, J.M.; Bader, G.D. Transfer learning for biomedical named entity recognition with neural networks. *Bioinformatics* **2018**, *34*, 4087–4094. [CrossRef]
55. Sheldon, M.R.; Fillyaw, M.J.; Thompson, W.D. The use and interpretation of the Friedman test in the analysis of ordinal-scale data in repeated measures designs. *Physiother. Res. Int.* **1996**, *1*, 221–228. [CrossRef]
56. Zimmerman, D.W.; Zumbo, B.D. Relative power of the Wilcoxon test, the Friedman test, and repeated-measures ANOVA on ranks. *J. Exp. Educ.* **1993**, *62*, 75–86. [CrossRef]

57. Peng, Y.; Yan, S.; Lu, Z. Transfer Learning in Biomedical Natural Language Processing: An Evaluation of BERT and ELMo on Ten Benchmarking Datasets. In Proceedings of the 18th BioNLP Workshop and Shared Task, BioNLP@ACL 2019, Florence, Italy, 1 August 2019; pp. 58–65. [CrossRef]
58. Lee, J.; Yoon, W.; Kim, S.; Kim, D.; Kim, S.; So, C.H.; Kang, J. BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* **2020**, *36*, 1234–1240. [CrossRef]
59. Tsai, R.T.; Wu, S.; Chou, W.; Lin, Y.; He, D.; Hsiang, J.; Sung, T.; Hsu, W. Various criteria in the evaluation of biomedical named entity recognition. *BMC Bioinform.* **2006**, *7*, 92. [CrossRef]
60. Sang, E.F.T.K.; Meulder, F.D. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Edmonton, AB, Canada, 31 May–1 June 2003; pp. 142–147.
61. Pyysalo, S.; Ananiadou, S. Anatomical entity mention recognition at literature scale. *Bioinformatics* **2014**, *30*, 868–875. [CrossRef]
62. Francis, S.; Van Landeghem, J.; Moens, M.F. Transfer Learning for Named Entity Recognition in Financial and Biomedical Documents. *Information* **2019**, *10*, 248. [CrossRef]
63. Krallinger, M.; Leitner, F.; Rabal, O.; Vazquez, M.; Oyarzabal, J.; Valencia, A. CHEMDNER: The drugs and chemical names extraction challenge. *J. Cheminform.* **2015**, *7*, S1. [CrossRef]
64. Wei, C.H.; Peng, Y.; Leaman, R.; Davis, A.P.; Mattingly, C.J.; Li, J.; Wiegers, T.C.; Lu, Z. Overview of the BioCreative V chemical disease relation (CDR) task. In Proceedings of the Fifth BioCreative Challenge Evaluation Workshop, Sevilla, Spain, 31 August 2015; Volume 14.
65. Kim, J.D.; Ohta, T.; Pyysalo, S.; Kano, Y.; Tsujii, J. Overview of BioNLP'09 shared task on event extraction. In Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task, Boulder, CO, USA, 5 June 2009; pp. 1–9.
66. Ohta, T.; Pyysalo, S.; Tsujii, J. Overview of the epigenetics and post-translational modifications (EPI) task of BioNLP shared task 2011. In Proceedings of the BioNLP Shared Task 2011 Workshop, Portland, OR, USA, 24 June 2011; pp. 16–25.
67. Pyysalo, S.; Ohta, T.; Miwa, M.; Tsujii, J. Towards exhaustive protein modification event extraction. In Proceedings of the BioNLP 2011 Workshop, Portland, OR, USA, 24 June 2011; pp. 114–123.
68. Pyysalo, S.; Ohta, T.; Rak, R.; Sullivan, D.; Mao, C.; Wang, C.; Sobral, B.; Tsujii, J.; Ananiadou, S. Overview of the infectious diseases (ID) task of BioNLP Shared Task 2011. In Proceedings of the BioNLP Shared Task 2011 Workshop, Portland, OR, USA, 24 June 2011; pp. 26–35.
69. Nédellec, C.; Bossy, R.; Kim, J.D.; Kim, J.J.; Ohta, T.; Pyysalo, S.; Zweigenbaum, P. Overview of BioNLP shared task 2013. In Proceedings of the BioNLP Shared Task 2013 Workshop, Sofia, Bulgaria, 8–9 August 2013; pp. 1–7.
70. Pyysalo, S.; Ohta, T.; Ananiadou, S. Overview of the cancer genetics (CG) task of BioNLP Shared Task 2013. In Proceedings of the BioNLP Shared Task 2013 Workshop, Sofia, Bulgaria, 8–9 August 2013; pp. 58–66.
71. Kim, J.D.; Kim, J.j.; Han, X.; Rebholz-Schuhmann, D. Extending the evaluation of Genia Event task toward knowledge base construction and comparison to Gene Regulation Ontology task. *BMC Bioinform.* **2015**, *16*, S3. [CrossRef]
72. Kim, J.D.; Wang, Y.; Yasunori, Y. The genia event extraction shared task, 2013 edition-overview. In Proceedings of the BioNLP Shared Task 2013 Workshop, Sofia, Bulgaria, 8–9 August 2013; pp. 8–15.
73. Ohta, T.; Pyysalo, S.; Rak, R.; Rowley, A.; Chun, H.W.; Jung, S.J.; Choi, S.P.; Ananiadou, S.; Tsujii, J. Overview of the pathway curation (PC) task of bioNLP shared task 2013. In Proceedings of the BioNLP Shared Task 2013 Workshop, Sofia, Bulgaria, 8–9 August 2013; pp. 67–75.
74. Basher, A.R.M.; Purdy, A.S.; Birol, I. Event extraction from biomedical literature. *bioRxiv* **2015.** [CrossRef]
75. Mi, H.; Thomas, P. PANTHER pathway: An ontology-based pathway database coupled with data analysis tools. In *Protein Networks and Pathway Analysis*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 123–140.
76. Bada, M.; Eckert, M.; Evans, D.; Garcia, K.; Shipley, K.; Sitnikov, D.; Baumgartner, W.A.; Cohen, K.B.; Verspoor, K.; Blake, J.A.; et al. Concept annotation in the CRAFT corpus. *BMC Bioinform.* **2012**, *13*, 161. [CrossRef]
77. Kim, J.D.; Ohta, T.; Tsuruoka, Y.; Tateisi, Y.; Collier, N. Introduction to the bio-entity recognition task at JNLPBA. In Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and Its Applications, Geneva, Switzerland, 28–29 August 2004; pp. 73–78.
78. Nguyen, N.T.; Gabud, R.S.; Ananiadou, S. COPIOUS: A gold standard corpus of named entities towards extracting species occurrence from biodiversity literature. *Biodivers. Data J.* **2019**, *7*, e29626. [CrossRef]
79. Doğan, R.I.; Leaman, R.; Lu, Z. NCBI disease corpus: A resource for disease name recognition and concept normalization. *J. Biomed. Inform.* **2014**, *47*, 1–10. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.