

RESEARCH ARTICLE

Open Access



# Multiple-level biomedical event trigger recognition with transfer learning

Yifei Chen

## Abstract

**Background:** Automatic extraction of biomedical events from literature is an important task in the understanding biological systems, allowing for faster update of the latest discoveries automatically. Detecting trigger words which indicate events is a critical step in the process of event extraction, because following steps depend on the recognized triggers. The task in this study is to identify event triggers from the literature across multiple levels of biological organization. In order to achieve high performances, the machine learning based approaches, such as neural networks, must be trained on a dataset with plentiful annotations. However, annotations might be difficult to obtain on the multiple levels, and annotated resources have so far mainly focused on the relations and processes at the molecular level. In this work, we aim to apply transfer learning for multiple-level trigger recognition, in which a source dataset with sufficient annotations on the molecular level is utilized to improve performance on a target domain with insufficient annotations and more trigger types.

**Results:** We propose a generalized cross-domain neural network transfer learning architecture and approach, which can share as much knowledge as possible between the source and target domains, especially when their label sets overlap. In the experiments, MLEE corpus is used to train and test the proposed model to recognize the multiple-level triggers as a target dataset. Two different corpora having the varying degrees of overlapping labels with MLEE from the BioNLP'09 and BioNLP'11 Shared Tasks are used as source datasets, respectively. Regardless of the degree of overlap, our proposed approach achieves recognition improvement. Moreover, its performance exceeds previously reported results of other leading systems on the same MLEE corpus.

**Conclusions:** The proposed transfer learning method can further improve the performance compared with the traditional method, when the labels of the source and target datasets overlap. The most essential reason is that our approach has changed the way parameters are shared. The vertical sharing replaces the horizontal sharing, which brings more sharable parameters. Hence, these more shared parameters between networks improve the performance and generalization of the model on the target domain effectively.

**Keywords:** Event trigger recognition, Transfer learning, Neural networks

## Background

Recently, as interest in biomedical research grows, an overwhelming amount of literature has been published online. As a result, there are incremental studies in applying Text Mining (TM) techniques for automatic recognizing and tracking of the new discoveries and theories in these biomedical articles. These biomedical TM applications include named entity (e.g. gene and protein

mentions) recognition, relation (e.g. protein-protein interactions) extraction between entities, and event (e.g. gene transcriptions and regulations) extraction, etc [1–3].

Event extraction refers to automatically extracting structured representations of biomedical relations, functions and processes from text [3]. Since the BioNLP'09 [4] and BioNLP'11 [5] Shared Tasks, event extraction has become a research focus. The structure of each event is defined as an arbitrary number of participants to indicate functions and processes on molecular level, such as “regulation” and “phosphorylation”. When a certain protein regulates the expression of a certain gene and its products are in turn

Correspondence: [yifeichen91@nau.edu.cn](mailto:yifeichen91@nau.edu.cn)  
School of Information Engineering, Nanjing Audit University, 86 West Yushan Road, Nanjing, China



involved in some phosphorylation processes, the “regulation” and “phosphorylation” events come into being. Event extraction task usually contains two main steps: identifying the event triggers and then identifying the event arguments according to the triggers [6]. Event trigger recognition, aiming at detecting those expressions from text that indicate certain events, is the first and crucial step of event extraction. Event extraction performance depends entirely on the recognized triggers. This point was clearly shown by Björne et al. [7]. They found that between using the gold standard and predicted triggers, the performance declined by more than 20 points. Many Machine Learning (ML) based methods, including Conditional Random Field (CRF) [8, 9], Support Vector Machine (SVM) [7, 10–13], and Deep Neural Network (DNN) [14–16] models have been successfully applied to event trigger recognition.

These machine learning based approaches rely on large quantity and high quality annotated training data. Their performance may deteriorate when certain training instances are insufficient. However, acquiring manually annotated datasets is both time consuming and costly. Up to now, the manual annotations of biological events mainly focus on genes and proteins. In the corpora of the Shared Tasks of BioNLP'09, 9 types of frequently used biomolecular events are annotated. Biomolecular events involving proteins and genes are an important part of the picture of biological systems, but still only a small part. Hence, in order to obtain a more comprehensive understanding of biological systems, the scope of event extraction has been broadened from molecular-level reactions to cellular-, tissue- and organ-level effects, and to organism-level outcomes [17]. It is not trivial to keep up to date with the annotations of the expanding event types across multiple levels. For example, in the MLEE corpus [10] multiple levels of events from the molecular level to the whole organism have been annotated. The number of event types has been extended to 19. But at the same time, the number of annotated instances for each event type has been greatly reduced. Thus, it will be useful that the annotated dataset from a related domain (such as biomolecular event annotations from the BioNLP'09 corpus) can help to alleviate the shortage of training data problem in the target domain (such as multiple-level event recognition from the MLEE corpus). Recently, transfer learning (TL) techniques have been proposed to address this need [18].

The concept of transfer learning comes from the observed fact that when learning in a new related domain, humans can usually benefit from what they have learned before [19]. This idea has been employed in data mining and machine learning fields [20–22] as a transfer learning schema. Pan and Yang [18] define transfer learning as using some knowledge learned from a source dataset to perform a task on a target dataset. And, transfer learning

has been successfully applied to many fields, including text mining [23, 24].

Here, we focus on the research of transfer learning for DNNs, due to their successful application in many text mining tasks over the last few years. Ideally, transfer learning can achieve higher performance by reducing the amount of annotated data needed, and improving generalization of the model on the target dataset. Normally, in the setting of TM and Natural Language Processing (NLP), according to the difference between the source and target datasets, transfer learning approaches of DNN models have three common categories: cross-lingual transfer, cross-domain transfer and cross-task transfer. Due to different languages, cross-lingual transfer is mostly limited to the use of additional language resources to transfer knowledge [25, 26] between the source and target datasets. It cannot extend to our biomedical event trigger recognition applications across multiple levels.

Sharing the same language, both cross-domain and cross-task transfer learning modes can take advantage of more relevance between source and target datasets. In these two modes, parameters of DNN models are used to transfer knowledge between source and target datasets. Some parameters of one model learned from a source dataset can be converted to initialize some parameters of another related model for optimizing on a target dataset. Usually, how many parameters can be shared depends on the degree of the relevance of the source and target datasets. Yang [27] examined the effects of transfer learning for deep hierarchical recurrent networks on several different sequence labelling tasks, including the cross-domain, cross-task and cross-lingual transfer learning models. And it was reported that significant improvement can be obtained. In the case of cross-domain transfer, the datasets of two domains are consistent when their label sets are identical or mappable to each other. Otherwise, the datasets of two domains are inconsistent. If the two domains are consistent, they can share the parameters of all the layers between the source and target DNN models. But, if they are inconsistent, the parameter sharing is restricted to the fewer layers of the DNN models. The cross-task transfer can be simply considered as the case of the cross-domain transfer using inconsistent label sets due to the fact that different tasks do not share the same tags. Hence, the same parameter sharing strategy is effective for them [27]. In the work of Meftah [28], both cross-task and cross-domain (with inconsistent source and target tags) transfer learning was implemented to address the problem of the need in annotated data of social media texts. And the validity and genericity of the models were demonstrated on the Part-Of-Speech (POS) tagging tasks. More studies on transfer learning have been successfully performed in the NLP sequence labelling tasks. Dong [29] proposed a multichannel DNN model to

transfer knowledge cross-domain in Chinese social media. In order to ensure the consistency of the source and target domains, some tags are merged in their paper. The experiments showed that the model achieved the best advanced performance. Lee [24] used cross-domain transfer learning for Named Entity Recognition (NER) with consistent tags, showing that transfer learning improved upon the state-of-the-art results on a target dataset with a small number of instances. Giorgi [30] demonstrated that transferring a DNN model significantly improved the latest leading results for biomedical NER, when the source and target domains are consistent.

Our aim in this study is to transfer the trigger recognition knowledge from the source molecular level domain to the target multiple-level domain. This can be seen as an exploratory step towards the more effective automatic extraction of targets from a complex and multifarious domain based on an available simple and singular domain. This situation often occurs in certain fields when research is extended from a familiar area to an unfamiliar and broader area. For instance, after the 9 types of molecular level event relationships between genes and proteins from the biomedical literature have been studied, the research focus will shift to other levels, and the event types will be expanded. The source and target domains, event triggers from different levels, are highly related. Under this circumstance, their label sets may overlap more or less. Nevertheless the annotations from the source and target domains are inconsistent, because their label sets are not identical and mappable. However, among all the above transfer learning studies, there is no model designed to solve how to share network parameters in the case of overlapping label sets. They just simplify the problem to the case of having different label sets between the source and target domains.

We present a new generalized transfer learning approach based on a DNN model, which attempts to share the knowledge to the extent possible between the related source and target domains. The transfer learning approach is modified and generalized to share more network parameters to improve trigger recognition performance across multiple levels on the target domain. Our approach mainly addresses transfer learning between the domains with overlapping label sets. In this paper, a source domain with plentiful annotations of biomolecular event triggers (the BioNLP corpus) is used to improve performance on a target domain of multiple-level event triggers with fewer available annotations (the MLEE corpus). To our knowledge, no reported research has applied transfer learning to make the best use of overlapping label sets to find the shared knowledge.

The rest of this paper is organized as follows. In “[Methods](#)” section, detailed descriptions of the proposed generalized transfer learning method and Multiple-Level

Trigger recogNizer (MLTrigNer) system are provided. “[Results](#)” section describes the used biomedical corpora, experimental settings, and all the experimental results. And this is followed by the in-depth analysis in “[Discussion](#)” section. We present the conclusions and future work in “[Conclusions](#)” section.

## Results

### Corpus description

An in-depth investigation is carried out to compare the performance of our proposed Multiple-Level event Trigger recogNizer, MLTrigNer, which is built based on the generalized cross-domain transfer learning BiLSTM-CRF model. The dataset  $Data_{MLEE}$  is used as the target domain dataset. With varying degrees of label overlapping,  $Data_{ST09}$  and  $Data_{EPI11}$  are used as the source domain datasets, respectively. Named entity and trigger types annotated in these corpora are illustrated in Table 1. In the trigger types of  $Data_{MLEE}$ , the labels overlapped with  $Data_{ST09}$  are marked using ‘\*’, and the labels overlapped with  $Data_{EPI11}$  are marked using ‘+’. We can see that  $Data_{MLEE}$  and  $Data_{ST09}$  are highly related because of the nine overlapping trigger labels. However, there are some overlapping labels that have gone beyond the molecular level in  $Data_{MLEE}$ , which annotate events across multiple levels. For example, “Localization” is the event type extracted from both cells and biomolecules in  $Data_{MLEE}$ .  $Data_{MLEE}$  and  $Data_{EPI11}$  are loosely related with only two overlapping trigger labels. More details of these datasets are introduced in the following.

### $Data_{MLEE}$

The MLEE corpus [10] is used to train and test our MLTrigNer on multiple-level trigger word identification as a target dataset. The corpus is taken from 262 PubMed abstracts focusing on tissue-level and organ-level processes, which are highly related to certain organism-level pathologies. In  $Data_{MLEE}$ , 19 event types are chosen from the GENIA ontology, which can be classified into four groups: anatomical, molecular, general and planned. Our task is to identify the correct trigger type of each event. Hence, there are 20 tags in the target label set, including a negative one. All the statistics in the training, development and test sets are shown in Table 2.

### $Data_{ST09}$

This corpus is taken from the Shared Task (ST) of BioNLP challenge 2009 [4] and contains training and development sets, including 950 abstracts from PubMed. It is used to train our MLTrigNer as a source dataset. In this corpus, 9 event types are chosen from the GENIA ontology involving molecular-level entities and processes, which can be categorized into 3 different groups: simple events, binding events and regulation events. The training

**Table 1** Named entity and trigger types in *Data<sub>MLEE</sub>*, *Data<sub>ST09</sub>* and *Data<sub>EPI11</sub>*, respectively

Corpus	Named entity type	Trigger type
<i>Data<sub>ST09</sub></i>	Protein	Gene expression*
		Transcription*, Binding*
		Protein catabolism*
		Phosphorylation*
		Localization*, Regulation*
		Positive regulation*
<i>Data<sub>EPI11</sub></i>	Protein	Hydroxylation, Dehydroxylation
		Phosphorylation+, Deglycosylation
		Dephosphorylation+, Catalysis
		Ubiquitination, Acetylation
		Deubiquitination
		DNA methylation
		DNA demethylation
		Glycosylation, Deacetylation
		Methylation, Demethylation
		<i>Data<sub>MLEE</sub></i>
Drug or compound	Development, Binding*	
Developing anatomical structure	Blood vessel develop	
Organ, Tissue	Growth, Death, Regulation*	
Immaterial anatomical entity	Breakdown, Remodeling	
Anatomical system	Synthesis, Localization*	
Organism, Cell	Gene expression*	
Pathological formation	Transcription*	
Organism subdivision	Protein catabolism*	
Multi-tissue structure	Phosphorylation*+	
Cellular component	Dephosphorylation+	
Organism substance	Positive regulation*	
	Negative regulation*	

In the trigger types of *Data<sub>MLEE</sub>*, the labels overlapped with *Data<sub>ST09</sub>* are marked using '\*', and the labels overlapped with *Data<sub>EPI11</sub>* are marked using '+'

**Table 2** Statistics of documents, words and events in the dataset *Data<sub>MLEE</sub>*, including the training set, the development set, and the test set, respectively

Item	Training	Development	Test
Document	131	44	87
Words	27,875	9610	19,103
Event	3296	1175	2206

and development sets are combined as a source domain dataset *Data<sub>ST09</sub>*. All of the detailed statistics of *Data<sub>ST09</sub>* are shown in Table 3.

#### *Data<sub>EPI11</sub>*

This corpus is taken from the Epigenetics and Post-translational Modifications (EPI) task of BioNLP challenge 2011 [5] and contains training and development sets, including 800 abstracts relating primarily to protein modifications drawn from PubMed. It is also used to train our MLTrigNer as a source dataset. In this corpus, 14 protein entity modification event types and their catalysis are chosen. Hence there are 15 event types totally. The training and development sets are combined as a source domain dataset *Data<sub>EPI11</sub>*. All of the detailed statistics in *Data<sub>EPI11</sub>* are shown in Table 4. The number of annotated events in *Data<sub>EPI11</sub>* is less than that in the *Data<sub>ST09</sub>*, annotating the more event types.

#### Performance assessment

We measure the performance of the trigger recognition system in terms of the *F1* measure. The *F1* is determined by a combination of precision and recall. Precision is the ratio of the number of correctly classified triggers within a category to the total number of recognized ones. Recall is the ratio of the number of correctly classified triggers within a category to the total number of triggers. They are defined as follows:

$$F1 - measure = \frac{2Precision \times Recall}{Precision + Recall} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

where *TP* is the number of the triggers that are correctly classified to a category, *FP* is the number of the triggers that are misclassified to a category, and *FN* is the number of the triggers misclassified to other categories.

#### Implementation details

All of the experiments described in the following are implemented using the Tensorflow library [31]. Hyperparameters are tuned using the training and development sets through cross-validation and then the final model

**Table 3** Statistics of documents, words and events in the training set, the development set and their combination as *Data<sub>ST09</sub>*, respectively

Item	Training	Development	<i>Data<sub>ST09</sub></i>
Abstract	800	150	950
Words	176,146	33,937	210,083
Event	8597	1809	10,406

**Table 4** Statistics of documents, words and events in the training set, the development set and their combination as  $Data_{EPI11}$ , respectively

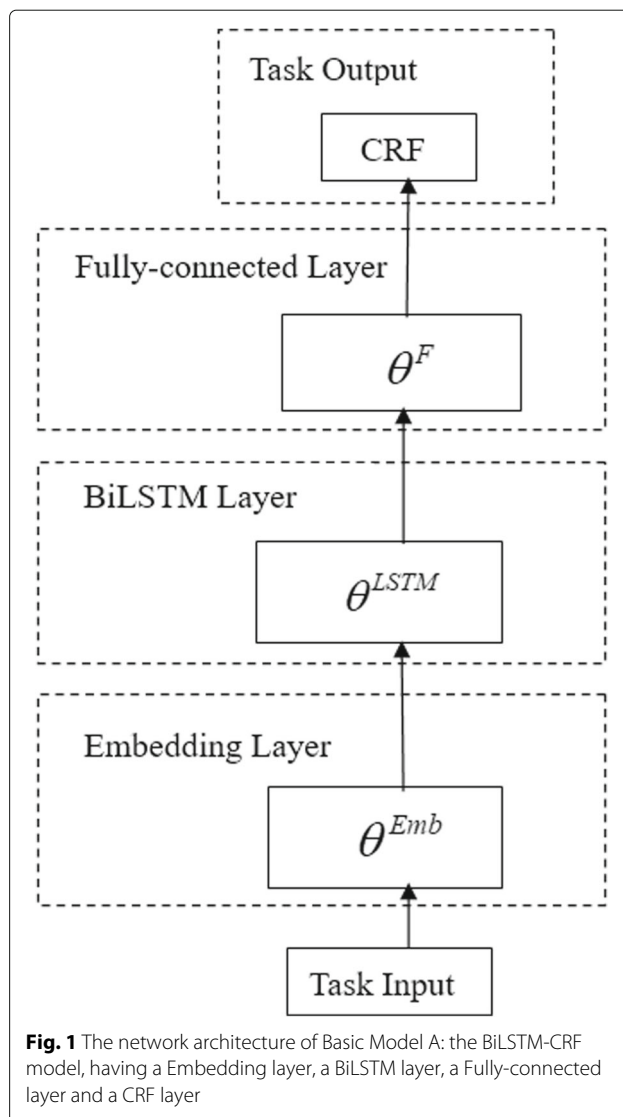
Item	Training	Development	$Data_{EPI11}$
Abstract	600	200	800
Words	127,312	43,497	170,809
Event	1852	601	2453

is trained on the combined set of the optimal ones. We tune the pre-trained word embedding vector  $E^w$  to 200 dimensions, character embedding vector  $E^c$  to 100 dimensions, named entity type embedding vector  $E^e$  to 10 for source domain while 50 dimensions for target domain, POS embedding vector  $E^p$  to 50 dimensions, pre-trained dependency tree-based word embedding vector  $E^d$  to 300 dimensions. Then, the BiLSTM layer with a hidden state dimension of 300, and the fully-connected layer with 600 dimensions. In order to avoid overfitting, dropout with a probability 0.5 is used before the input to the BiLSTM and fully-connected layers.

#### Transfer learning performance

The effectiveness of our proposed is approach illustrated based on the performance comparison of the three neural network models described in “Methods” section. First, the Basic Model A (Fig. 1) is trained only on the training and development sets of  $Data_{MLEE}$  (without transfer learning) as a baseline measurement, and its results are shown in the second column of Table 5. Then,  $Data_{ST09}$  is used as the source dataset in the transfer learning models. The TL Model C (Fig. 2) and the MLTrigNer model (Fig. 3) are jointly trained on  $Data_{ST09}$  and the training and development sets of the target dataset  $Data_{MLEE}$  using different transfer learning approaches, respectively. The three models are tested on the test set of  $Data_{MLEE}$ . The results are shown in the third and fourth columns of Table 5. Among the models described in “Methods” section, the TL Model B (Fig. 4) cannot be used in the trigger recognition task since the domain-dependent input feature sets are employed, which are inconsistent in the source and target domains.

From the results of the Basic Models A and the TL Model C, we can see that the transfer learning improves the  $F1$  measure 1.76%. Generalizing the transfer learning schema in the MLTrigNer Model improves the trigger recognition performance a further 1.78%. This improvement is due to the fact that in our approach, more parameters are transferred from the source network to the target one than usual, signifying more effective knowledge sharing. It is worth noting there are improvements in both precision and recall, which refers to the ability of the MLTrigNer to identify more positive triggers.



Higher precision and recall signify identification of more potential biomedical events during the subsequent processing phase, which is important for the ultimate event extraction application. Compared with the TL Model C, beside “Negative regulation” and “Localization”, the  $F1$  values of the other trigger types overlapping with the source dataset are improved. Among these overlapping labels, some of them have gone beyond the molecular level in  $Data_{MLEE}$  to annotate events across multiple levels. Moreover, the  $F1$  values of the 7 non-overlapping trigger types are also improved, except for “Growth”, “Dephosphorylation” and “Planned process”. Hence, our proposed approach can improve the recognition performance across multiple levels through transfer more knowledge from a single level domain.

Then,  $Data_{EPI11}$  is used as the source dataset alternatively. Basic Model A (Fig. 1) was also trained only



**Table 5** Detailed results achieved by the proposed MLTrigNer Model, Basic Model A and TL Model C on *Data<sub>MLEE</sub>*

Trigger type	Basic Model A			TL Model C			MLTrigNer Model		
	P	R	F1	P	R	F1	P	R	F1
Cell proliferation	85.37	81.40	83.33	80.95	79.07	80.00	81.40	81.40	81.40
Development	66.37	76.53	71.09	76.29	75.51	75.90	78.00	79.59	<b>78.79</b>
Blood vessel develop	97.33	94.19	95.74	97.98	93.87	95.88	100.0	94.52	<b>97.18</b>
Growth	96.00	85.71	90.57	90.74	87.50	89.09	92.31	85.71	88.89
Death	73.68	75.68	74.67	67.39	83.78	74.70	69.57	86.49	<b>77.11</b>
Breakdown	82.35	63.64	71.79	75.00	68.18	71.43	87.50	63.64	<b>73.68</b>
Remodeling	71.43	50.00	58.82	55.55	50.00	52.63	85.71	66.67	<b>75.00</b>
Synthesis	50.00	25.00	33.33	0.0	0.0	0.0	100.0	100.0	<b>100.0</b>
Gene expression	91.67	83.33	87.30	91.80	84.85	88.19	94.44	90.15	<b>92.25</b>
Transcription	0.0	0.0	0.0	50.00	16.67	25.00	100.0	66.67	<b>80.00</b>
Protein Catabolism	0.0	0.0	0.0	0.0	0.0	0.0	100.0	60.00	<b>75.00</b>
Phosphorylation	75.00	100.0	85.71	60.00	100.0	75.00	75.00	100.0	85.71
Dephosphorylation	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Localization	78.83	81.20	80.00	82.09	82.71	82.40	82.17	79.70	80.92
Binding	86.96	70.18	77.67	83.02	77.19	80.00	91.49	75.43	<b>82.69</b>
Regulation	59.80	58.93	59.37	60.27	63.77	61.97	61.71	66.18	<b>63.87</b>
Positive regulation	80.88	81.90	81.39	84.82	81.59	83.17	84.81	85.07	<b>84.94</b>
Negative regulation	84.73	65.71	74.02	80.75	78.78	79.75	78.96	75.10	76.99
Planned process	78.69	48.98	60.38	74.15	55.61	63.56	78.76	58.67	67.25
TOTAL	81.63	74.26	77.77	81.52	77.66	79.53	83.31	79.40	<b>81.31</b>

The Basic Model A is trained only on the training and development sets of *Data<sub>MLEE</sub>* without transfer learning. The TL Model C and the MLTrigNer model are jointly trained on the source dataset *Data<sub>ST09</sub>* and the training and development sets of the target dataset *Data<sub>MLEE</sub>* using different transfer learning approaches, respectively. The three models are tested on the test set of *Data<sub>MLEE</sub>*. In the results of MLTrigNer Model, the improved F1 values are marked in bold

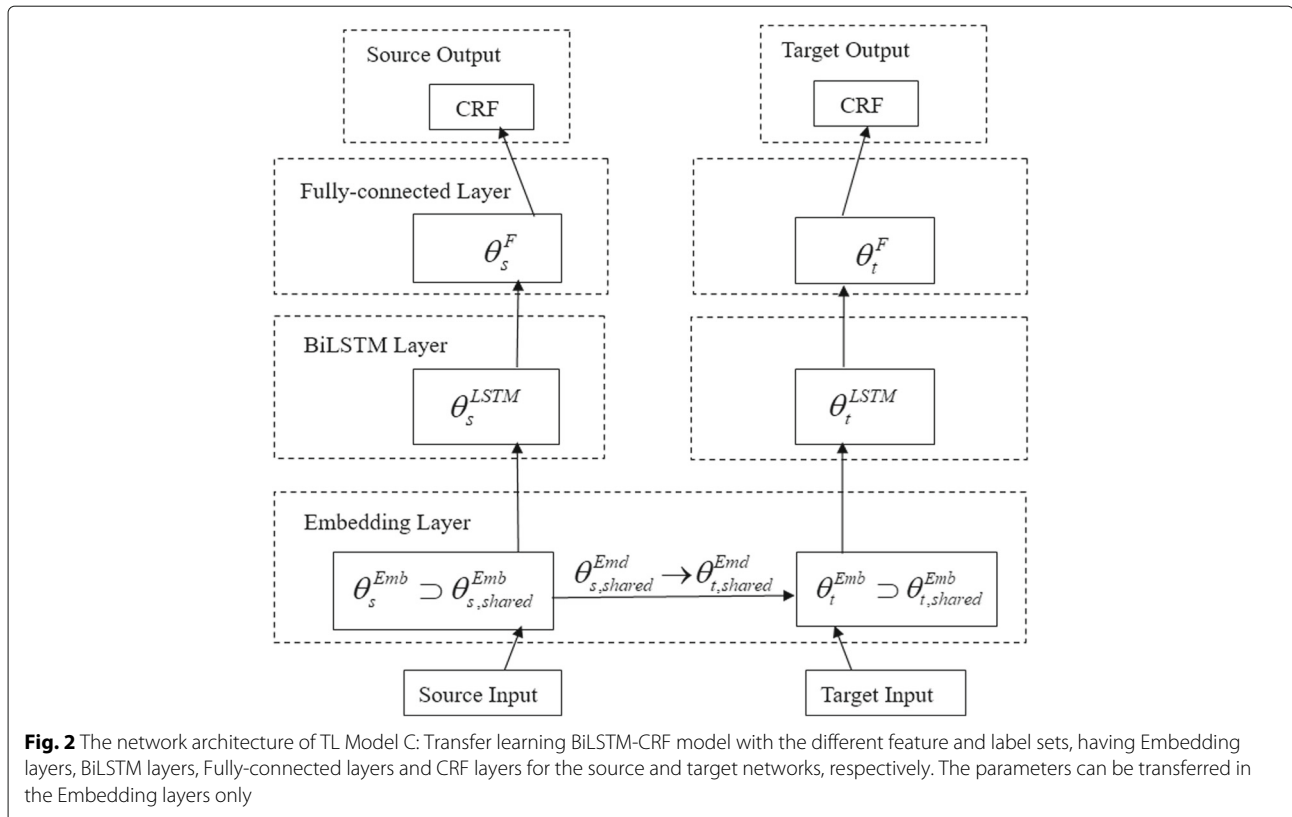
on the training and development sets of *Data<sub>MLEE</sub>* (without transfer learning) as a baseline measurement, and its results are shown in the second column of Table 6. The TL Model C (Fig. 2) and the MLTrigNer Model (Fig. 3) are then jointly trained on the source dataset *Data<sub>EPI11</sub>* and the training and development sets of the target dataset *Data<sub>MLEE</sub>* using different transfer learning approaches. The results are shown in the third and fourth columns of Table 6, respectively. The three models are tested on the test set of *Data<sub>MLEE</sub>*.

From the results of the Basic Model A and the TL Model C, we can see that the transfer learning improves the F1 measure 0.87%. The MLTrigNer Model improves the performance a further 1.04%, and the improvements are also both in precision and recall. Using *Data<sub>EPI11</sub>* as the source dataset, the MLTrigNer Model brings less performance improvement. This is due to the decreased correlation between the source and target domains. In the transfer learning models, less parameters can be transferred from the source to the target networks.

However, our MLTrigNer Model still can improve the performance further compared with the basic transfer learning approach. Hence, our proposed method is effective when the overlapping is more or less. Compared with the TL Model C, the recognition performance of the overlapping trigger “Phosphorylation” is not improved, and its F1 measure is 100.0 in both models, which cannot be improved further. Moreover, the performance of the 13 non-overlapping trigger types are all improved.

#### MLTrigNer compared with other trigger recognition systems

We compare the performance of the proposed transfer learning based trigger recognition system, MLTrigNer, with other leading systems on the same *Data<sub>NMLEE</sub>* dataset. Since *Data<sub>ST09</sub>* as the source dataset shows the better performance from the results in Tables 5 and 6, we utilized *Data<sub>ST09</sub>* to train the MLTrigNer Model as the source dataset. The detailed F1 measure results are illustrated in Table 7.



Peysalo et al. [10] defined an SVM-based classifier with rich hand-crafted features to recognize triggers in the text. Zhou et al. [13] also defined an SVM-based classifier with word embeddings and hand-crafted features. Nie et al. [14] proposed a word embedding-assisted neural network model to model semantic and syntactic information in event trigger identification (the results were converted to 19 categories). Wang et al. [15] defined a window-based convolution neural network (CNN) classifier. Rahul et al. [16] proposed a method that uses a recurrent neural network (RNN) to extract higher-level sentence features in trigger identification.

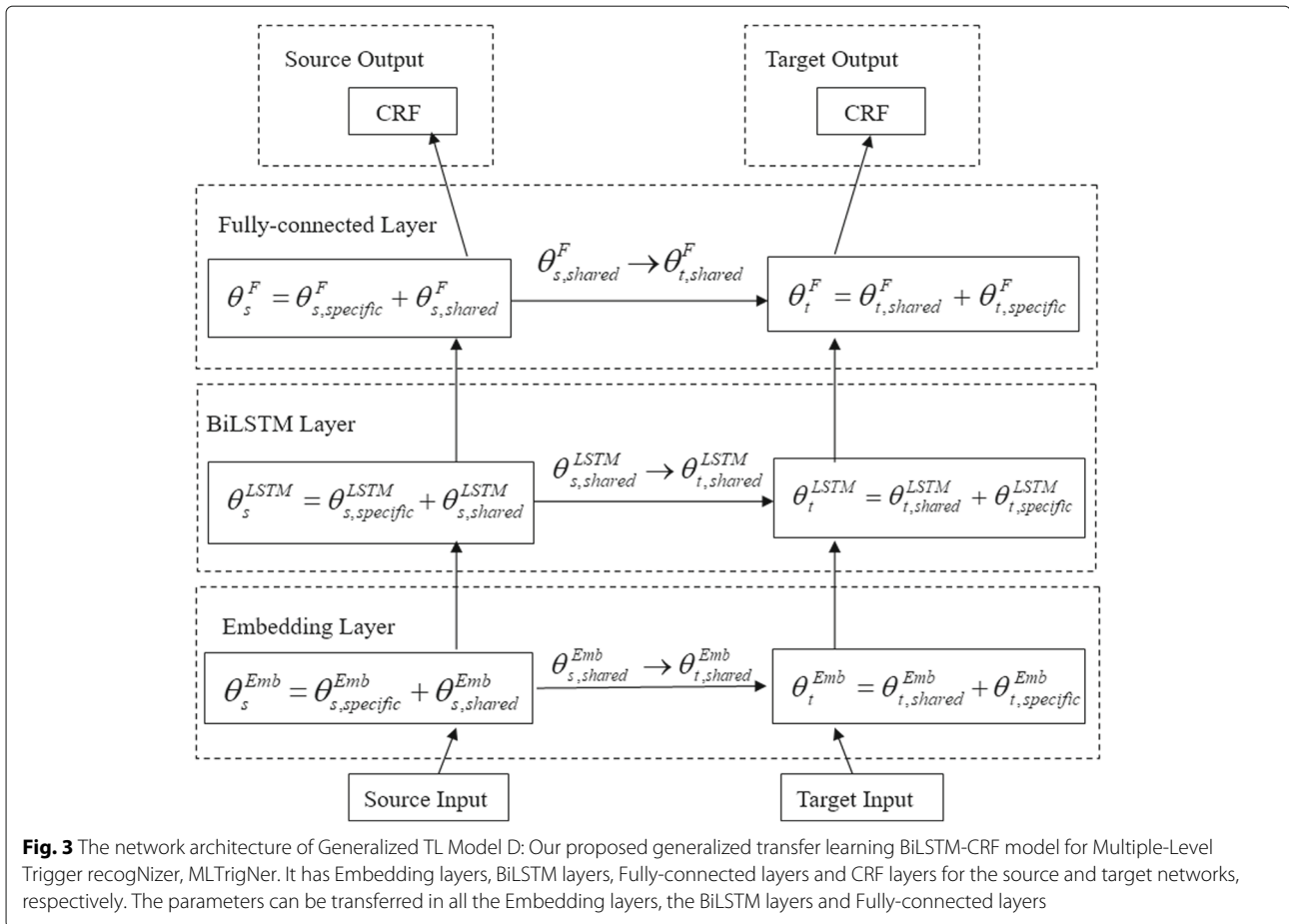
From Table 7, we can draw two conclusions. First, our generalized transfer learning approach achieves the best result on the dataset  $Data_{MLEE}$ , which indicates that our MLTrigNer can still improve the performance of biomedical trigger word recognition. Second, from Table 5, the TL Model C achieves competitive results compared to these leading systems, which means that the improvement of our generalized transfer learning approach is achieved on a relatively strong basis.

## Discussion

### Transfer performance analysis on highly related domains

We conduct an in-depth study and detailed comparison on the highly related domains of  $Data_{ST09}$  and  $Data_{MLEE}$

to show the learning ability of our proposed approach. In our study, there are two datasets with the different overlapping degrees of the labels used as source domains to transfer knowledge, respectively. Between them,  $Data_{ST09}$  is highly related with the target domain. Its trigger types are nested in those of the target domain dataset from Table 1. Hence, we can simply put the  $Data_{ST09}$  and the training and development sets of  $Data_{MLEE}$  together to train the BiLSTM-CRF model without transfer learning (Basic Model A), and then the model is tested on the test set of  $Data_{MLEE}$ . Its performance is shown in Table 8 in the line of "Basic Model A ( $Data_{MLEE} + Data_{ST09}$ )". For the purpose of comparison, in the line of "Basic Model A ( $Data_{MLEE}$ )", the performance of Basic Model A trained on the training and development sets of  $Data_{MLEE}$  and tested on the test set of  $Data_{MLEE}$  is listed. And in the last line, the performance of our MLTrigNer Model is shown, which uses  $Data_{ST09}$  and  $Data_{MLEE}$  as the source and target datasets, respectively. From the results we can see that the performance even declines when just simply mixing nested datasets together. On the other hand, the performance can be improved using our transfer learning approach. In the process of trigger recognition, the shared knowledge brought by the transfer learning is more important than the data itself.



### Ratio effect analysis on source data

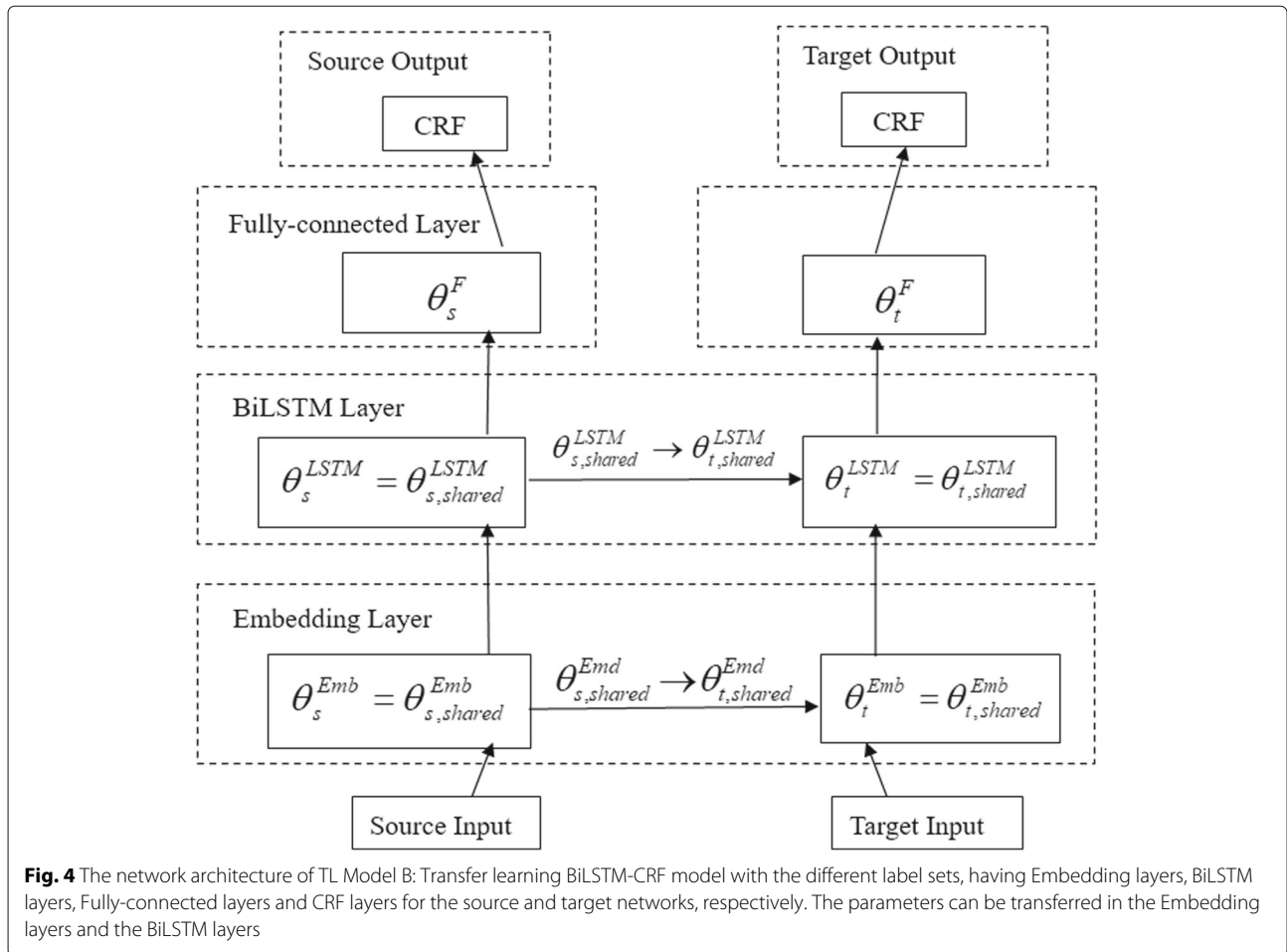
It is important to analyze the effect of the ratio of source domain data. First, we use *Data<sub>ST09</sub>* as the source dataset, which is more than 3.6 times the size of the target domain dataset. We keep the size of target data unchanged, and gradually change the size of source data. The changes in the MLTrigNer Model results are shown as a curve in Fig. 5, with the source ratio as 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% and 100%. We can see that *F1* first goes up continuously as the source data is added. Then it reaches a maximum of 81.31 when the source ratio is 80%. Finally, it trends downwards even as more source data is added, reaching 80.46 with 100% data in *Data<sub>ST09</sub>*. The results verify that more data from source domain does not always lead to better performance in target domain. In our study, the optimal source/target ratio is about 2.9 : 1 when maximum performance achieved in *Data<sub>MLEE</sub>*. In order to optimize the performance of the model under different datasets, we set the ratio of source domain data to be one of the important hyperparameters of the MLTrigNer model, which is tuned on the training and development sets using cross-validation.

Then, we use *Data<sub>EPI11</sub>* as the source dataset alternatively, which is about 3.1 times the size of the target domain dataset. We also keep the size of the target data unchanged, and gradually change the size of the source data. The changes in the MLTrigNer Model results are shown as a curve in Fig. 6, with the source ratio as 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% and 100%. Similar trends are found in the Figs. 5 and 6. The values of *F1* measure first goes up continuously as source training data is added, and reaches a maximum of 79.68 when the source ratio is 90%. Then, it trends downwards even as more source data is added, reaching 79.45 with 100% data in *Data<sub>EPI11</sub>*. After tuned on the training and development sets using cross-validation, the optimal source/target ratio is about 2.7 : 1 when maximum performance achieved in *Data<sub>MLEE</sub>*.

### Error analysis

From the metrics in Tables 5 and 6 we can notice that the results of the trigger type “Dephosphorylation” are all zeroes regardless of the models. From a more detailed list of types and sizes of trigger words of the *Data<sub>MLEE</sub>* in Table 9, we can see that there are only 6





“Dephosphorylation” instances in the  $Data_{MLEE}$ . Without adequate training instances, the recognition results of the Basic Model A and TL Model C are very poor. Moreover, with our transfer learning approach, its recognition results of the MLTrigNer model are still zeroes under the situation that “Dephosphorylation” is an overlapping trigger type. This is a limitation of our transfer learning approach that it cannot transfer enough knowledge from other triggers for labelling the rare trigger types.

## Conclusions

In this paper we develop a novel transfer learning approach for multiple-level event trigger recognition based on a DNN model. We design a more general transfer learning approach to set the cross-domain transfer, which can share as much knowledge as possible between the source and target datasets, particularly encompassing the case of overlapping label sets. In the experiments, the source datasets having varying degrees of overlapping labels with the target dataset are utilized to verify the effectiveness of our proposed MLTrigNer model. Compared with the basic transfer learning model,

our approach improves the performance on the target domain further. Moreover, its performance exceeds other leading trigger recognition systems on the same MLEE corpus. Hence this study contributes to the effective recognition of biomedical trigger words from text across multiple levels. Through analysis, it is found that there are three essential factors mattering to our cross-domain transfer learning approach: the degree of overlapping of the source and target domains; the number of sharable parameters in each layer of a network; and an appropriate size of the source and target datasets. In the future work, more source datasets from different biomedical event levels with varying degrees of overlapping label tags can be utilized together to improve the performance further.

## Methods

In this section, we introduce our proposed transfer learning approach. Our solution for trigger recognition is based on a Bidirectional LSTM-CRF model (BiLSTM-CRF) [32], which uses a deep neural network, Long Short Term Memory (LSTM) [33], to extract higher-level abstract

**Table 6** Detailed results achieved by the proposed MLTrigNer Model, Basic Model A and TL Model C on  $Data_{MLEE}$ 

Trigger type	Basic Model A			TL Model C			MLTrigNer Model		
	P	R	F1	P	R	F1	P	R	F1
Cell proliferation	85.37	81.40	83.33	83.33	81.40	82.35	81.40	81.40	81.40
Development	66.37	76.53	71.09	74.51	77.55	76.00	78.35	77.55	<b>77.95</b>
Blood vessel develop	97.33	94.19	95.74	98.64	93.87	96.20	98.99	94.84	<b>96.87</b>
Growth	96.00	85.71	90.57	88.89	85.71	87.27	92.45	87.50	89.91
Death	73.68	75.68	74.67	66.67	81.08	73.17	66.67	81.08	73.17
Breakdown	82.35	63.64	71.79	73.68	63.64	68.29	87.50	63.64	<b>73.68</b>
Remodeling	71.43	50.00	58.82	75.00	30.00	42.86	66.67	40.00	50.00
Synthesis	50.00	25.00	33.33	33.33	25.00	28.57	20.00	25.00	22.22
Gene expression	91.67	83.33	87.30	85.51	89.39	87.41	89.05	92.42	<b>90.71</b>
Transcription	0.0	0.0	0.0	50.00	16.67	25.00	100.0	16.67	<b>28.57</b>
Protein Catabolism	0.0	0.0	0.0	0.0	0.0	0.0	33.33	20.00	<b>25.00</b>
Phosphorylation	75.00	100.0	85.71	100.0	100.0	100.0	100.0	100.0	100.0
Dephosphorylation	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Localization	78.83	81.20	80.00	77.14	81.20	79.12	82.17	79.70	<b>80.92</b>
Binding	86.96	70.18	77.67	83.02	77.19	80.00	79.25	73.68	76.36
Regulation	59.80	58.93	59.37	65.13	61.65	63.34	65.17	63.29	<b>64.22</b>
Positive regulation	80.88	81.90	81.39	81.11	82.91	82.00	81.96	82.22	<b>82.09</b>
Negative regulation	84.73	65.71	74.02	77.18	75.61	76.39	80.72	73.47	<b>76.92</b>
Planned process	78.69	48.98	60.38	66.86	57.65	61.92	71.07	57.65	<b>63.66</b>
TOTAL	81.63	74.26	77.77	79.69	77.62	78.64	81.76	77.71	<b>79.68</b>

The Basic Model A is trained only on the training and development sets of  $Data_{MLEE}$  without transfer learning. The TL Model C and the MLTrigNer model are jointly trained on the source dataset  $Data_{EPI11}$  and the training and development sets of the target dataset  $Data_{MLEE}$  using different transfer learning approaches, respectively. The three models are tested on the test set of  $Data_{MLEE}$ . In the results of MLTrigNer Model, the improved F1 values are marked in bold

features to train a CRF [34]. We design a transfer learning approach to allow for joint training with a source dataset, which uses an input feature set and a output label set that overlap with the target dataset, respectively.

We first introduce and describe the architecture of the BiLSTM-CRF model as Basic Model A. We then introduce the cross-domain transfer learning BiLSTM-CRF model with inconsistent label sets as TL Model B, and in addition with inconsistent input feature sets as TL Model C.

**Table 7** Detailed performance results achieved by the proposed MLTrigNer and the other leading trigger recognition systems, respectively

Trigger Recognition System	Precision	Recall	F1-Measure
Our MLTrigNer system	83.31	79.40	<b>81.31</b>
SVM-based System [10]	81.44	69.48	75.67
SVM-based System [13]	75.56	81.29	78.32
Neural Network based System [14]	71.04	84.60	77.23
CNN-based System [15]	80.67	76.76	78.67
RNN-based System [16]	79.78	78.45	79.11

In these results, the best F1 value of our MLTrigNer system is marked in bold

Finally, our proposed generalized transfer learning model, Generalized TL Model D, is described in detail. The different architectures of the four models are shown in Figs. 1, 4, 2 and 3, respectively.

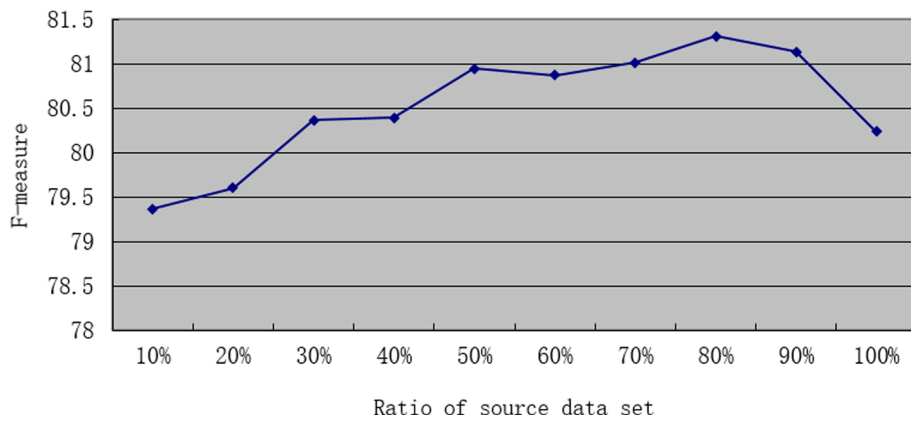
#### Basic model a: biLSTM-CRF model

We present our trigger recognition task based on the BiLSTM-CRF model as Basic Model A, whose architecture is shown in Fig. 1. In Basic Model A,  $\theta$ s denote all the

**Table 8** Detailed performance results on highly related domains with different training modes, including the Basic Model A (trained on the training and development sets of  $Data_{MLEE}$ ), the Basic Model A (trained on the combination of  $Data_{ST09}$  and the training and development sets of  $Data_{MLEE}$ ), and our MLTrigNer Model (using  $Data_{MLEE}$  as the target dataset and  $Data_{ST09}$  as the source dataset)

Trigger Recognition System	Precision	Recall	F1-Measure
Basic Model A ( $Data_{MLEE}$ )	81.63	74.26	77.77
Basic Model A ( $Data_{MLEE} + Data_{ST09}$ )	78.78	73.92	76.28
Our MLTrigNer Model ( $Data_{MLEE} + Data_{ST09}$ )	83.31	79.40	<b>81.31</b>

In these results, the best F1 value of our MLTrigNer model is marked in bold



**Fig. 5** The ratio effect of source domain data *Data<sub>5T09</sub>* to our transfer learning model, MLTrigNer, with the ratio as 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% and 100%

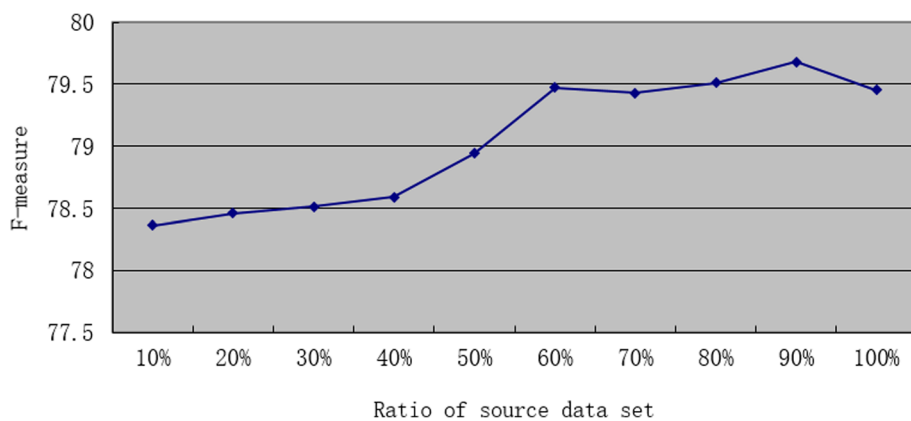
trainable parameters in each network layer. This model detects trigger words and annotates their types, and its performance serves as the baseline. For a given input sentence  $\{word_1, word_2, \dots, word_n\}$ , the aim of trigger recognition is to output a tag sequence  $\{tag_1, tag_2, \dots, tag_n\}$ , where  $word_i$  is a word (or a token) in the sentence and  $tag_i$  denotes its corresponding type label. The value of  $tag_i$  belongs to the label set, which is a biomedical event type or negative if it does not indicate any event. The BiLSTM-CRF model feeds a set of features for an input embedding layer (with parameters  $\theta^{Emb}$ ), extracts higher-level abstract features in subsequence BiLSTM (with parameters  $\theta^{LSTM}$ ) and fully-connected (with parameters  $\theta^F$ ) layers, and trains a CRF layer for the final sequence labelling. The main layers of the BiLSTM-CRF model for trigger recognition are described below.

**Embedding layer**

In order to express both syntactic and semantic information in input sentences, besides each word,  $word_i$ , we also

extract other four features from character, POS, named entity type and dependency parse tree. Through lookup tables, the embedding layer converts each input feature into one of the following representation vectors:

- 1 Word embedding vector  $E^w$ : Each word in an input sentence is mapped to a word embedding vector, which contains semantic information from its linear contexts. In this paper, we use a pre-trained word lookup table  $LT^w$  learned from PubMed articles using the word2vec model [35].
- 2 Character embedding vector  $E^c$ : We use an extra LSTM network to extract the orthographic information from the sequence of characters in each input word. Its parameters  $LT^c$  are weights and biases of the LSTM, which are initialized randomly and trained to output a character-level embedding vector.
- 3 POS embedding vector  $E^p$ : We train a POS lookup table  $LT^p$  to extend the word embedding. It maps the POS tag of each word in an input sentence to a POS



**Fig. 6** The ratio effect of source domain data *Data<sub>EP11</sub>* to our transfer learning model, MLTrigNer, with the ratio as 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% and 100%

**Table 9** List of types and sizes of trigger words in the  $Data_{MLEE}$ , where “Dephosphorylation” is a rare trigger type

	Trigger type	size in $Data_{MLEE}$
Anatomical	Cell proliferation	133
	Development	316
	Blood vessel develop	855
	Growth	169
	Death	97
	Breakdown	69
	Remodeling	33
Molecular	Synthesis	17
	Gene expression	435
	Transcription	37
	Protein catabolism	26
	Phosphorylation	33
	Dephosphorylation	6
General	Localization	450
	Binding	184
	Regulation	773
	Positive regulation	1327
	Negative regulation	921
Planned	Planned process	643

embedding vector, which extracts syntactic information from the input word.  $LT^p$  is initialized randomly and trained to obtain a mapping lookup table.

- 4 Named entity type embedding vector  $E^e$ : We train a lookup table  $LT^e$  to map named entity type of each word in an input sentence to an embedding vector to extract domain-dependent information. The named entities were provided by the task data.  $LT^e$  is initialized randomly and trained to output a mapping lookup table.
- 5 Dependency tree-based word embedding vector  $E^d$ : In order to extend features from linear word contexts to non-linear syntactic contexts, each word from an input sentence is mapped to a dependency tree-based word embedding vector, which contains rich non-linear functional and syntactic information. We use a pre-trained word lookup table  $LT^d$  learned from English Wikipedia using the skip-gram model [36].

In the embedding layer, trainable parameter set can be expressed as  $\theta^{Emb} = \{LT^c, LT^p, LT^e\}$ .

#### **BiLSTM layer**

This layer takes a concatenation of the output embedding vectors of the previous embedding layer as input,

$x_i = [E_i^w; E_i^c; E_i^p; E_i^e; E_i^d]$ . Due to the ability to learn long-distance dependencies in a sequence through designed memory cells, LSTM is a powerful tool for sequence labelling tasks [33]. Suppose that an input sequence to a LSTM layer is  $\{x_1, x_2, \dots, x_T\}$ , and it yields an output sequence of  $\{h_1, h_2, \dots, h_T\}$  by employing the following implementation strategy during training [32], where both sequences have the same length  $T$ :

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (4)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (5)$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (6)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (7)$$

$$h_t = o_t \tanh(c_t) \quad (8)$$

where  $\sigma$  denotes the logistic sigmoid function,  $\tanh$  is the hyperbolic tangent activation function, and all weights ( $W$ s) and biases ( $b$ s) make up the parameter set ( $\theta^{LSTM}$ ) of the LSTM layer. More details about the LSTM can be referred to in [32]. In sequence labelling tasks, it is better to be able to process both the past (from the left side) and the future (from the right side) context dependencies in the sequence. Therefore, another commonly used version of the LSTM is employed, called the Bidirectional LSTM (BiLSTM) [32, 37]. In the BiLSTM, for each word the forward LSTM captures the features from the left side and the backward LSTM captures the features from the right side. Each word effectively encodes information about the whole sentence.

#### **Fully-Connected layer**

The output of the BiLSTM layer at each time step  $t$ , obtained by concatenating the outputs of the forward and backward LSTMs  $h_t = [h_t^F; h_t^B]$ , is mapped to a linear and fully-connected network layer using ReLU activation functions as follows:

$$y_t = \max(0, W_t h_t + b_t) \quad (9)$$

where all weights ( $W$ s) and biases ( $b$ s) make up the parameter set ( $\theta^F$ ) of the fully-connected layer.

#### **CRF layer**

On the top of the fully-connected layer, a final CRF layer generates a sequence of labels for corresponding words. The CRF layer can learn the strong dependencies across output labels and come into the most likely sequence of the predicted tags [38].

### Transfer learning approach

The goal of cross-domain transfer in this study is to learn a sequence labelling model for triggers which transfers knowledge from a source domain to a related target domain.

#### TL model b

When the label sets of the source and target domains are inconsistent, including overlapping, it is treated as the case of the domains having completely different label sets in the basic idea of transfer learning. In this situation, the architecture of TL Model B is an extension of the basic BiLSTM-CRF model. And the source and target domains share the same input feature sets in the model. The TL Model B in Fig. 4 gives an overview of how to transfer parameters ( $\theta$ s) of each neural network layer between both datasets within a certain range.

Let  $s$  and  $t$  represent the source domain and the target domain, respectively. And the parameter sets of each model layer  $l$  are  $\theta_s^l$  and  $\theta_t^l$  for the source and target domains, including the embedding layers ( $\theta_s^{Emd}$  and  $\theta_t^{Emd}$ ), the BiLSTM layers ( $\theta_s^{LSTM}$  and  $\theta_t^{LSTM}$ ), and the fully-connected layers ( $\theta_s^F$  and  $\theta_t^F$ ). The transfer learning process consists of learning the parameters ( $\theta_s^{Emd}$ ,  $\theta_s^{LSTM}$  and  $\theta_s^F$ ) of a neural network on a source dataset, then transferring a part of them to another neural network and optimizing parameters ( $\theta_t^{Emd}$ ,  $\theta_t^{LSTM}$  and  $\theta_t^F$ ) on a target dataset. In TL Model B, without the same label sets, only the parameters of the embedding and BiLSTM layers can be transferred and shared, as illustrated below:

$$\theta_s^{Emd} = \theta_{s,shared}^{Emd}, \theta_t^{Emd} = \theta_{t,shared}^{Emd}, \text{with } \theta_{s,shared}^{Emd} \rightarrow \theta_{t,shared}^{Emd} \quad (10)$$

$$\theta_s^{LSTM} = \theta_{s,shared}^{LSTM}, \theta_t^{LSTM} = \theta_{t,shared}^{LSTM}, \text{with } \theta_{s,shared}^{LSTM} \rightarrow \theta_{t,shared}^{LSTM} \quad (11)$$

where the subscript *shared* means the parameters that can be shared and transferred between the source and target domains. After training on the source domain, all the embedding and BiLSTM layer parameters,  $\theta_s^{Emd}$  and  $\theta_s^{LSTM}$ , are mapped to initialize the parameters of the corresponding layers on the target dataset,  $\theta_t^{Emd}$  and  $\theta_t^{LSTM}$ . Hence we have  $\theta_{s,shared}^{Emd} \rightarrow \theta_{t,shared}^{Emd}$  and  $\theta_{s,shared}^{LSTM} \rightarrow \theta_{t,shared}^{LSTM}$ . It also means that the parameters of the fully-connected layer,  $\theta_s^F$  and  $\theta_t^F$ , should be trained separately because of the inconsistent label sets.

#### TL model c

When with their own domain-dependent features, such as named entity type, the input feature sets of the source and target domains are inconsistent. The BiLSTM layers will have the different parameter dimensions and

structures due to the different feature sets. Hence, the parameters of this layer cannot be shared neither. In this situation, the only parameters that can be transferred are from the embedding layer as shown in Eq. 12. More specifically, the shared parameters are those lookup tables trained for domain-independent features,  $\theta_{s,shared} = \{TL^w, TL^c, TL^p, TL^d\}$ , where  $TL^w$  and  $TL^d$  are pre-trained. The TL Model C in Fig. 2 gives an overview of how to transfer the parameters between the neural network layers of both datasets.

$$\theta_s^{Emd} \supset \theta_{s,shared}^{Emd}, \theta_t^{Emd} \supset \theta_{t,shared}^{Emd}, \text{with } \theta_{s,shared}^{Emd} \rightarrow \theta_{t,shared}^{Emd} \quad (12)$$

#### Generalized tL model d (MLTrigNer): our transfer learning approach

This study uses the corpus with biomolecular trigger annotations as the source domain dataset and the corpus with multiple-level biomedical event triggers as the target domain dataset. Because of their inconsistent input feature and output label sets, we just can choose the TL Model C shown in Fig. 2 to build a trigger recognizer, without sharing the parameters of the fully-connected and BiLSTM layers. This ignores the information hidden in the overlapping features and labels. It is known in transfer learning that the more parameters are shared, the better generalization can be achieved in the target domain. For this purpose, we propose a generalized transfer learning architecture and approach to share as many parameters as possible to explore the transferability of each layer in a neural network, especially when the feature and label sets are overlapping.

As we discussed that parameters stand for the abstract features learned from a neural network. In the basic transfer learning architectures, TL Model B and C, the parameters are chosen to be transferred according to the network layers horizontally. When the label sets of the source and target domains are consistent, parameters from the upper (fully-connected) and middle (BiLSTM) layers can be transferred. Otherwise, when the label sets are inconsistent, the parameters of the whole upper layer closest to the output are discarded in TL Model B. Moreover, when the source and the target domains have inconsistent extracted feature sets, the parameters of the whole middle layer should be discarded in TL Model C. After careful study of the lower (embedding) layer of TL Model C, we find out that all these parameters learned from the source domain can be split into two parts: a source-specific part and a source-target-shared part. Correspondingly, the parameters of the target domain also can be split into two parts: a target-specific part and a source-target-shared part. This kind of divide is vertical within a network layer, and the source-target-shared part of the parameters can transfer the information carried by



the overlapping of feature and label sets in the middle and upper layers. The main benefit is that we can include more domain-dependent features in the lower layer. For instance, in our trigger recognition task, there is a different and richer named entity type feature set in the target domain.

Figure 3 shows how we generalize the basic transfer learning approach to share as many parameters as possible. As mentioned, the parameters are split into two parts, domain-specific and domain-shared parameters:

$$\theta_s^l = \theta_{s,specific}^l + \theta_{s,shared}^l, \theta_t^l = \theta_{t,specific}^l + \theta_{t,shared}^l \quad (13)$$

where  $\theta_{s,shared}^l$  and  $\theta_{t,shared}^l$  are the parameters shared and mapped through the transfer learning in each layer  $l$ , and the domain specific parameters  $\theta_{s,specific}^l$  and  $\theta_{t,specific}^l$  are trained for each domain exclusively.

The degree of parameters to be transferred from the source network to the target network is determined according to the overlapping degrees of the input feature and output label sets between the source and target domains. Figure 3 shows the parameter sharing situation of the MLTrigNer. In general, suppose  $\{x_1^l, x_2^l, \dots, x_j^l, \dots\}$  are the inputs of each layer  $l$ ,  $\{y_1^l, y_2^l, \dots, y_j^l, \dots\}$  are the outputs, and parameters  $\theta$  of this layer are all weights ( $W^l$ 's) and biases ( $b^l$ 's). Since parameters can be divided into the domain-shared and domain-specific parts, their connected inputs and outputs can also be divided accordingly.

For the middle layers, such as the BiLSTM layers, of the source and target networks in Fig. 3, they have domain-specific and shared inputs of feature embedding vectors as  $[x_{specific}^l, x_{shared}^l]$ . Hence the corresponding domain-specific and shared connection weights for each output  $y_j^l$  are  $[W_{j,specific}^l, W_{j,shared}^l]$ , and each output  $y_j^l$  has its own bias  $b_j^l$ . The shared parameters in Eq. 13,  $\theta_{s,shared}^l$  and  $\theta_{t,shared}^l$  are  $\{W_{shared}^l, b^l\}$ . We can obtain each output  $y_j^l$  as follows:

$$y_j^l = active\_function \left( \left[ \begin{array}{c} (W_{j,specific}^l)^T, (W_{j,shared}^l)^T \\ \left[ \begin{array}{c} x_{specific}^l \\ x_{shared}^l \end{array} \right] + b_j^l \end{array} \right] \right) \quad (14)$$

For the upper layers, such as the fully-connected layers, of the source and target networks in Fig. 3, they have domain-specific and shared label outputs as  $[y_{specific}^l, y_{shared}^l]$ . Hence the domain-specific and shared parameters for the corresponding outputs are  $\{W_{j,specific}^l, b_{j,specific}^l\}$  and  $\{W_{j,shared}^l, b_{j,shared}^l\}$ , respectively. The shared parameters in Eq. 13,  $\theta_{s,shared}^l$  and  $\theta_{t,shared}^l$

are  $\{W_{shared}^l, b_{shared}^l\}$ . We can obtain each domain-specific output  $y_{j,specific}^l$  and shared output  $y_{j,shared}^l$  as follows:

$$y_{j,specific}^l = active\_function \left( \left( W_{j,specific}^l \right)^T x + b_{j,specific}^l \right) \quad (15)$$

$$y_{j,shared}^l = active\_function \left( \left( W_{j,shared}^l \right)^T x + b_{j,shared}^l \right) \quad (16)$$

If the feature sets are the exactly same on both domains, there are no source-specific and target-specific parts of the parameters for the BiLSTM layers,  $\theta_{s,specific}^{LSTM} = \emptyset$ ,  $\theta_{t,specific}^{LSTM} = \emptyset$ . Moreover, under this circumstance, if the label sets are completely different from each other on both domains, there are no source-target-shared parameters for the fully-connected layer,  $\theta_{s,shared}^F = \theta_{t,shared}^F = \emptyset$ , which is the TL Model B. On the other hand, if the label sets and the feature sets are inconsistent, we have  $\theta_{s,shared}^{LSTM} = \theta_{t,shared}^{LSTM} = \emptyset$  and  $\theta_{s,shared}^F = \theta_{t,shared}^F = \emptyset$ , which is the TL Model C.

The training takes place over the following three main phases. First, the network is trained on the dataset from the source domain. Both  $\theta_{s,specific}^l$  and  $\theta_{s,shared}^l$  are learned. Then the shared parameters of each layer are transferred to the target domain,  $\theta_{s,shared}^l \rightarrow \theta_{t,shared}^l$ , to initialize the corresponding parts of the target model parameters. Finally, the network is trained on the dataset from the target domain. Both  $\theta_{t,specific}^l$  and  $\theta_{t,shared}^l$  are tuned and optimized.

**Abbreviations**

BiLSTM-CRF: Bidirectional LSTM-CRF model; BiLSTM: Bidirectional LSTM; CNN: Convolution neural network; CRF: Conditional random field; DNN: Deep neural network; EPI: Epigenetics and post-translational modifications; LSTM: Long short term Memory; ML: Machine learning; MLTrigNer: Multiple-level trigger recognizer system; NER: Named entity recognition; NLP: Natural language processing; POS: Part-of-speech; RNN: Recurrent neural network; ST: Shared task; SVM: Support vector machine; TL: Transfer learning; TM:Text mining

**Acknowledgements**

The author wants to thank the anonymous reviewers for their helpful comments and suggestions.

**Author's contributions**

YC designed research, performed research, analyzed data, and wrote the paper. The author read and approved the final manuscript.

**Funding**

This work is supported by the Natural Science Foundation of Jiangsu Province under No.BK20171495. The funding did not have a role in the design of the study and collection, analysis, and interpretation of data and in writing the manuscript.

**Availability of data and materials**

The MLEE corpus analysed during the current study is available in <http://www.nactem.ac.uk/MLEE/#availability> [10]. Accessed on 20 May 2019. The corpus from the BioNLP'09 Shared Tasks is available in <http://www.geniaproject.org/shared-tasks/bionlp-shared-task-2009> [4]. Accessed on 20 May 2019. The corpus from the BioNLP'11 Shared Tasks is available in <http://weaver.nlplab.org/~bionlp-st/BioNLP-ST/downloads/downloads.shtml> [5]. Accessed on 20 May 2019.

**Ethics approval and consent to participate**

Not applicable.

**Consent for publication**

Not applicable.

**Competing interests**

The author declares that she has no competing interests.

Received: 11 January 2019 Accepted: 16 August 2019

Published online: 06 September 2019

**References**

- Ananiadou S. In: McNaught J, editor. Text mining for biology and biomedicine. London: Artech House Publishers; 2005.
- Zweigenbaum P, Demner-Fushman D, Yu H, Cohen KB. Frontiers of biomedical text mining: current progress. *Brief Bioinform*. 2007;8(5):358–75.
- Ananiadou S, Pyysalo S, Tsujii J, Kell DB. Event extraction for systems biology by text mining the literature. *Trends Biotechnol*. 2010;28(7):381–90.
- Kim JD, Ohta T, Pyysalo A, 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. Morristown: Association for Computational Linguistics; 2009. p. 1–9.
- Kim JD, Pyysalo S, Ohta T, Bossy R, Nguyen N, Tsujii J. Overview of BioNLP Shared Task 2011. In: Proceedings of the BioNLP Shared Task 2011 Workshop. Portland, Oregon: Association for Computational Linguistics; 2011. p. 1–6.
- Kim JD, Tsujii J. Corpus annotation for mining biomedical events from literature. *BMC Bioinformatics*. 2008;10(9):10–34.
- Björne J, Heimonen J, Ginter F, Airola A, Pahikkala T, Salakoski T. Extracting complex biological events with rich graph-based feature sets. In: Proceeding of BioNLP Shared Task 2009 Workshop. Boulder: Association for Computational Linguistics; 2009. p. 10–8.
- Yang C, Lin H, Yang Z, Wang J, Li Y. Biomolecular event trigger detection using neighborhood hash feature. *J Theor Biol*. 2013;218:22–8.
- Campos D, Bui QC, Matos S, Oliveira JL. TrigNER: automatically optimized biomedical event trigger recognition on scientific documents. *Source Code Biol Med*. 2014;9:1.
- Pyysalo S, Ohta T, Miwa M, Cho HC, Tsujii J, Ananiadou S. Event Extraction across Multiple Levels of Biological Organization. *Bioinformatics*. 2012;28(18):i575–81.
- Miwa M, Sætre R, Kim JD, Tsujii J. Event extraction with complex event classification using rich features. *J Bioinform Comput Biol*. 2010;8:131–46.
- Björne J, Ginter F, Salakoski T. University of Turku in the BioNLP'11 shared task. *BMC Bioinforma*. 2014;13(Suppl 11):S4.
- Zhou D, Zhong D, He Y. Event Trigger Identification for Biomedical Events Extraction Using Domain Knowledge. *Bioinformatics*. 2014;30(11):1587–94.
- Nie Y, Rong W, Zhang Y, et al. Embedding Assisted Prediction Architecture for Event Trigger Identification. *J Bioinform Comput Biol*. 2015;13(3):i575–7.
- Wang J, Li H, An Y, Lin H, Yang Z. Biomedical event trigger detection based on convolutional neural network. *Int J Data Min Bioinforma*. 2016;15(3):195–213.
- Rahul PV, Sahu SK, Anand A. Biomedical Event Trigger Identification Using Bidirectional Recurrent Neural Network Based Models. Vancouver: Association for Computational Linguistics; 2017, pp. 316–21.
- Kitano H. Systems biology: A brief overview. *Science*. 2002;295:1662–4.
- Pan SJ Yang. A survey on transfer learning. *IEEE Trans Knowl Data Eng*. 2010;22(10):1345–59.
- Shinner BF. Science and human behavior. New York: Colliler Macmillian; 1953.
- Thrun S. Is learning the N-th thing any easier than learning the first? In: Proceedings of NIPS-96; 1996. p. 640–6.
- Baxter J. A Bayesian/information theoretic model of learning to learn via multiple task sampling. *Mach Learn*. 1997;28(1):7–39.
- Wang D, Zheng TF. Transfer learning for speech and language processing. Hong Kong: Proceeding of Signal and Information Processing Association Annual Summit and Conference (APSIPA); 2015. <https://doi.org/10.1109/APSIPA.2015.7415532>.
- Mou L, Meng Z, Yan R, Li G, Xu Y, Zhang L, Jin Z. How transferable are neural networks in nlp applications? *CoRR*. 2016. <https://doi.org/10.18653/v1/D16-1046>.
- Lee JY, Deroncourt F, Szolovits P. Transfer learning for named entity recognition with neural networks. *CoRR*. 2017. abs/1705.06273.
- Ziriky A, Hagiwara M. Cross-lingual transfer of named entity recognizers without parallel corpora. In: ACL. Stroudsburg: Association for Computational Linguistics; 2015. <https://doi.org/10.3115/v1/P15-2064>.
- Wang M, Manning CD. Cross-lingual pseudo-projected expectation regularization for weakly supervised learning. *TACL*. 2013. arXiv:1310.1597.
- Yang Z, Salakhutdinov R, Cohen WW. Transfer learning for sequence tagging with hierarchical recurrent networks. In: ICLR. Toulon: OpenReview.net; 2017. arXiv:1703.06345.
- Meftah S, Semmar N, Sadat F. A Neural Network Model for Part-Of-Speech Tagging of Social Media Texts. In: Processing of Eleventh International Conference on Language Resources and Evaluation. Miyazaki: European Language Resources Association (ELRA); 2018.
- Dong C, Wu H, Zhang J, Zong C. Multichannel LSTM-CRF for Named Entity Recognition in Chinese Social Media. In: Sun M, Wang X, Chang B, Xiong D, editors. Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data. NLP-NABD 2017, CCL 2017. Lecture Notes in Computer Science. Cham: Springer; 2017. p. 197–208.
- Giorgi JM, Bader GD. Transfer learning for biomedical named entity recognition with neural networks. *Bioinformatics*. 2018;34(23):4087–94.
- Abadi M, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *CoRR*. 2016. abs/1603.04467.
- Huang Z, Xu W, Yu K. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*. 2015. abs/1508.01991.
- Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput*. 1997;9(8):1735–80.
- Lafferty JD, McCallum A, Pereira FCN. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01. San Francisco: Morgan Kaufmann Publishers Inc.; 2001. p. 282–9.
- Pyysalo S, Ginter F, Moen H, Salakoski T, Ananiadou S. Distributional semantics resources for biomedical text processing. In: Proceedings International Symposium on Languages in Biology and Medicine (LBM). Tokyo: Database Center for Life Science; 2013. p. 39–44.
- Levy O, Goldberg Y. Dependency-Based Word Embeddings. In: ACL. Stroudsburg: Association for Computational Linguistics; 2014. p. 302–8.
- Graves A. Generating sequences with recurrent neural networks. *CoRR*. 2013. abs/1308.0850.
- Lample G, Ballesteros M, Subramanian S, Kawakami K, Dyer C. Neural architectures for named entity recognition. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Stroudsburg: Association for Computational Linguistics; 2016. p. 260–70.

**Publisher's Note**

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Ready to submit your research? Choose BMC and benefit from:**

- fast, convenient online submission
- thorough peer review by experienced researchers in your field
- rapid publication on acceptance
- support for research data, including large and complex data types
- gold Open Access which fosters wider collaboration and increased citations
- maximum visibility for your research: over 100M website views per year

At BMC, research is always in progress.

Learn more [biomedcentral.com/submissions](https://biomedcentral.com/submissions)

