Weber, L., Münchmeyer, J., Rocktäschel, T., Habibi, M., Leser, U. (2020): HUNER: Improving Biomedical NER with Pretraining. - Bioinformatics, 36, 1, 295-302.

https://doi.org/10.1093/bioinformatics/btz528

Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

OXFORD

Subject Section

HUNER: Improving Biomedical NER with Pretraining

Leon Weber [1,†,*], Jannes Münchmeyer [1,2,†,*], Tim Rocktäschel [3], Maryam Habibi [1] and Ulf Leser [1]

[1]Computer Science Department, Humboldt-Universität zu Berlin, Berlin 10099, Germany
[2]Helmholtzzentrum Potsdam, Deutsches GeoForschungsZentrum GFZ, Potsdam 14473, Germany
[3]University College London, Department of Computer Science, Gower Street, London WC1E 6BT, UK

*To whom correspondence should be addressed.
† These authors contributed equally to this work.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Several recent studies showed that the application of deep neural networks advanced the state-of-the-art in named entity recognition (NER), including biomedical NER. However, the impact on performance and the robustness of improvements crucially depends on the availability of sufficiently large training corpora, which is a problem in our field with its often rather small gold standard corpora.
Results: We evaluate different methods for alleviating the data sparsity problem by pre-training a deep neural network (LSTM-CRF), followed by a rather short fine-tuning phase focusing on a particular corpus. Experiments were performed using 34 different corpora covering five different biomedical entity types, yielding an average increase in F1-score of 2.5% compared to learning without pre-training. We experimented both with supervised and semi-supervised pre-training, leading to interesting insights into the precision/recall trade-off. Based on our results, we created the stand-alone NER tool HUNER incorporating fully trained models for five entity types. On the independent CRAFT corpus, which was not used for creating HUNER, it outperforms the state-of-the-art tools GNormPlus and tmChem by 5%-10% on the entity types chemicals, species, and genes.
Availability: HUNER is freely available at `https://hu-ner.github.io`. HUNER comes in containers, making it easy to install and use, and it can be applied off-the-shelf to arbitrary texts. We also provide an integrated tool for obtaining and converting all 34 corpora used in our evaluation, including fixed training, development and test splits to enable fair comparisons in the future.
Contact: weberple@hu-berlin.de, munchmej@informatik.hu-berlin.de
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Named entity recognition in the biomedical domain (BioNER) is a central task for information extraction to better cope with the vast amount of biomedical literature. The current state of the art in BioNER are LSTM-CRFs, a method originally proposed by (Lample *et al.*, 2016). This architecture combines a recurrent neural network with a long short term memory (Hochreiter and Schmidhuber, 1997) for learning (possibly long-ranging) correlations of features over the input texts and a CRF (Lafferty *et al.*, 2001) for predicting the tag sequence which identifies the entities. Unfortunately, training such deep architectures requires large amounts of annotated gold standard data. This poses a problem to applications in biomedicine, where corpora sometimes contain less than 500 sentences and rarely exceed a few thousands (see Table 1).

One approach to mitigate this problem is transfer learning. The general idea is to include prior knowledge into a model using data which has similarities to the actual target data but cannot be considered as gold standard (Pan *et al.*, 2010). The recently very popular word embeddings can be seen as a form of transfer learning, as the knowledge on statistical

co-occurrences of words derived from arbitrary unlabeled texts (sharing the same language) is given to the model as prior knowledge (Mikolov *et al.*, 2013); we recently showed that this approach can significantly improve the performance of BioNER (Habibi *et al.*, 2017).

Another form of transfer learning, which is very prominent especially for neural network models and which we study in the present work, is pre-training the entire model (and not only the word vectors). Here, the typical random model initialization is replaced by a phase where model parameters are learned using corpora other but similar to the target corpus. Pre-training of language models has shown to yield major improvements in various tasks of natural language processing, including sentence and text classification (Dai and Le, 2015; Howard and Ruder, 2018), sequence-to-sequence learning (Ramachandran *et al.*, 2016), and question answering (Min *et al.*, 2017). Giorgi and Bader applied this principle to a LSTM-CRF architecture for BioNER (Giorgi and Bader, 2018) and report improvements in F1-score (compared to Habibi *et al.* (2017)) between 0.13% and 2.81% for four different entity types using 23 corpora. This work uses pre-training on a silver standard corpus sampled from the CALBC-SSC-III-Small corpus (Kafkas *et al.*, 2012), which was created by unifying the output of several biomedical NER tools.

In this work, we significantly extend the work from Giorgi and Bader. First, we explore different pre-training schemes, namely a silver-standard and a gold-standard approach. The former is very similar to the setup in Giorgi and Bader (2018). However, in contrast to using a publicly available corpus, we create a silver standard by (1) training a CRF on the union of available gold standard corpora for a given entity type, (2) applying this model on all PubMed abstracts published until 2015, (3) and filtering out all tagged entities for which recognition probability of the CRF model is below a given threshold. This setup allows us to control the trade off between diversity in the pre-training data and the number of falsely tagged entities, as a higher threshold during entity filtering leads to only high-quality training instances but with limited diversity. In contrast, a lower threshold increases the number of falsely tagged entities but increases diversity. We compare the performance of this semi-supervised approach to pre-training with a strictly supervised one, where pre-training is performed directly on other gold standard corpora for the same entity type, leading to a cross-corpus setup (Tikk *et al.*, 2010). Second, we include in our evaluation all five entity types used in Habibi *et al.* (2017) and extend the number of corpora to 34. Third, we also perform evaluations without fine-tuning, i.e., we use the pre-trained model directly as NER tool. Clearly, such an approach has problems whenever the annotation styles of the pre-training corpora differ strongly from those of the target corpus. On the other hand, it is the only realistic setup for applying NER to unseen texts in novel applications, where no specific training data is available nor specific annotation guidelines have to be obeyed. To our surprise, we found this model without fine-tuning to work astonishingly well. Hence, we bundled the gold-standard pre-trained model into an easy-to-use stand alone tool, called HUNER, which is able to perform off-the-shelf NER for five different biomedical entity types. On the CRAFT corpus, HUNER outperforms tmChem (Leaman *et al.*, 2015) and GNormPlus (Wei *et al.*, 2015), state-of-the-art biomedical NER tools that also don't require re-training, by a considerable margin.

HUNER and scripts for downloading, transforming, and splitting all corpora used in this evaluation are freely available at `https://github.com/hu-ner/huner`.

2 Methods

2.1 Corpora and Pre-Processing

Following (Habibi *et al.*, 2017), we conduct experiments on 34 corpora spanning the five different entity types chemicals, cell lines, diseases, genes, and species, as well as the two different text genres of patent documents and scientific articles, including both abstracts and full-texts. To ensure that follow-up work can be fairly compared to the proposed approach, we built a fully scripted pre-processing pipeline which downloads all corpora, converts them from their specific source format into the standard CoNLL2003 format (Tjong Kim Sang and De Meulder, 2003), and performs unified pre-processing using a custom NLP pipeline. This pipeline employs the maximum entropy models for *OpenNLP 1.5.0*[1] together with *OpenNLP 1.9.0*[2] to split texts into sentences and to annotate them with Part-of-Speech tags.

Subsequently, each resulting data set is split into train, development and test sets on a document level. This splitting is carried out in a deterministic way, to ensure that follow-up work can use the same splits as used in this work. The ratios between training, development and test set are 60:10:30. Where possible, splits were chosen to be identical to those used in Habibi *et al.* (2017). To avoid knowledge leaks in the gold standard settings, we adjusted the splits in such a way that any sentence appearing in a train set does not appear in a development or test set for any other corpus of the same entity type. This is especially important, as some corpora are based on the same documents. Statistics on the resulting set of corpora can be found in Table 1.

2.2 BioNER using Pre-Training

We use the LSTM-CRF architecture and implementation of Lample *et al.* (2016). An extensive description of this architecture can be found in Habibi *et al.* (2017). We also follow Habibi *et al.* (2017) in the choice of hyper-parameters. That is, we use vanilla Stochastic Gradient Descent with a learning rate of 0.005, a dropout rate of 0.3, a bidirectional character LSTM with 25 units for each direction and a bidirectional word LSTM with 100 units for each direction. For fine-tuning, we lower the learning rate to 0.0005. We employ 200-dimensional pre-trained word embeddings from Pyysalo *et al.* (2013), trained on a combination of PubMed abstracts[3] (nearly 23 million abstracts), PMC articles[4] (nearly 700,000 full texts) and English Wikipedia articles[5] (approximately 4 million articles). This configuration was shown to achieve good performance in BioNER (Habibi *et al.*, 2017).

We evaluated two pre-training schemes. Both schemes employ a two step training process (see Figure 1). In the first step, the pre-training step, we train a LSTM-CRF on a large pre-training corpus (details below). In the second step, we use the model weights obtained to initialize a LSTM-CRF which is then fine-tuned during a second training specific for the target corpus at hand. Pre-training in general is intended to increase the models ability to generalize from the training examples to unseen test examples. While the joint corpus does not capture the full characteristics of the target corpus, it offers a larger training set. Therefore, more characteristics of the entity type and the language can potentially be learned. The fine-tuning step then only needs to adapt to the characteristics of the target corpus and not the general task of finding entities.

[1] http://opennlp.sourceforge.net/models-1.5/

[2] https://opennlp.apache.org

[3] See https://www.ncbi.nlm.nih.gov/pubmed/

[4] See https://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/

[5] See https://dumps.wikimedia.org/

Table 1. Corpus statistics for the gold standard corpora. We abbreviate patents (Pat.), Scientific articles (Art.), abstracts (A) and full-texts (F).

Corpora	Text Genre	Text Type	Entity Type	# Sentences	# Tokens	# Unique Tokens	# Annotations	# Unique Annotations
CHEMDNER patent Krallinger *et al.* (2015b,a)	Pat.	A	Chemicals	49402	1465776	62379	63761	21305
			Genes/proteins	49429	1465776	62379	12615	5909
CHEBI[a]	Pat.	F	Chemicals	13791	313713	24358	16248	4961
BioSemantics Akhondi *et al.* (2014)	Pat.	F	Chemicals	345520	5678380	207053	327299	67716
			Disease	347213	5684374	207336	19574	4438
CHEMDNER Krallinger *et al.* (2015a)	Art.	A	Chemicals	88934	2236229	114837	79329	24640
CDR Li *et al.* (2016)	Art.	A	Chemicals	14228	323281	23068	15411	3629
			Diseases	14247	323281	23068	12630	3466
BioCreative II GM Smith *et al.* (2008)	Art.	A	Genes/proteins	20000	508257	50864	22838	16509
JNLPBA Kim *et al.* (2004)	Art.	A	Genes/proteins	18546	492551	22056	29447	9203
			Cell Lines	18546	492551	22056	10480	4270
CellFinder Neves *et al.* (2012)	Art.	F	Genes/proteins	2176	65031	7977	1348	615
			Species	2177	65031	7977	433	51
			Cell Lines	2177	65031	7977	354	71
OSIRIS Furlong *et al.* (2008)	Art.	A	Genes/proteins	1043	28697	4669	768	275
DECA Wang *et al.* (2010)	Art.	A	Genes/proteins	5470	138034	14515	5973	2375
Variome Verspoor *et al.* (2013)	Art.	F	Genes/proteins	8288	172409	12649	4382	610
			Diseases	8287	172409	12649	5508	637
			Species	8288	172409	12649	182	8
FSU-PRGE Hahn *et al.* (2010)	Art.	A	Genes/proteins	36216	960436	44559	58595	13075
IEPA Ding *et al.* (2001)	Art.	A	Genes/proteins	486	15174	2923	1089	211
BioInfer Pyysalo *et al.* (2007)	Art.	A	Genes/proteins	1100	33858	5200	4327	1501
miRNA Bagewadi *et al.* (2014)	Art.	A	Genes/proteins	2644	65998	7821	1004	410
			Diseases	2644	65998	7821	2109	671
			Species	2644	65998	7821	726	47
NCBI Disease Doğan *et al.* (2014)	Art.	A	Diseases	7140	172717	12836	6768	2322
Arizona Disease Leaman *et al.* (2009)	Art.	A	Diseases[b]	2783	73773	8302	3036	1323
SCAI Diseases Gurulingappa *et al.* (2010)	Art.	A	Diseases	5173	112340	11049	2240	1002
SCAI Chemicals Kolárik *et al.* (2008)	Art.	A	Chemicals	1170	30567	5125	1204	797
S800 Pafilis *et al.* (2013)	Art.	A	Species	7857	195197	20526	3613	1582
LocText Goldberg *et al.* (2015)	Art.	A	Genes/proteins	952	22550	4371	1839	761
			Species	952	22550	4371	273	39
Linneaus Gerner *et al.* (2010)	Art.	F	Species	21997	480813	34761	2782	406
CLL Kaewphan *et al.* (2015)	Art.	A, F	Cell Lines	402	7764	2411	341	308
Gellus Kaewphan *et al.* (2015)	Art.	A, F	Cell Lines	11809	312584	20118	650	210

[a] http://chebi.cvs.sourceforge.net/viewvc/chebi/chapati/

[b] Excluded from evaluation, as the corpora is nearly completely contained in other corpora

2.3 Creating Pre-Training Corpora

We evaluated two different ways of creating such a joint pre-training corpus.

In the gold standard pre-training (GSPT), we create aggregated training and development sets by combining all training and development sets of all corpora for each entity type. This guarantees that the annotation quality of the original corpora is preserved, meaning that we add no false positives or false negatives compared to the original corpora. On the other hand the resulting corpus does not follow a consistent annotation guideline and has only limited size, bounded by the available gold standard corpora. The resulting corpora vary in size between 19,853 sentences for cell lines and 281,883 sentences for chemicals. The number of entity mentions vary between 4,772 for species and 287,972 for chemicals (see Table 2). For the final evaluation, we use these corpora as input for a entity-specific LSTM-CRF fine-tuned for 100 epochs. As final model, we selected the one achieving the best score on the development set.

To overcome the size limitations of GSPT, we also evaluated a semi-supervised (silver standard) pre-training scheme (SSPT). We created one joint silver standard corpus for all entity types. For each entity type we used the pre-training corpus from the gold standard scheme, consisting of the union of all training sets for that entity type, to train a CRF model[6]. With these models we annotated all abstracts indexed by PubMed up to the year 2015 (as pre-processed and provided by Hakala *et al.* (2016)), and removed all sentences with a confidence lower than a given threshold. If a token was annotated as belonging to more than one entity type, we only kept the conflicting entity type with highest confidence. We added a similar number of negative sentences to the corpus, for which we chose the sentences with the highest confidence scores for not containing an entity of any type. This procedure yielded a silver standard corpus consisting of 4,292,383 sentences, containing 1,402,866 annotations for chemicals, 803,292 annotations for diseases, 23,644 annotations for genes, 26,536 annotations for cell lines and 258,313 annotations for species (see Table 2). Again, we use this corpus as input for entity-specific LSTM-CRF fine-tuning on the target-corpus for five epochs and keep the best-performing model for evaluation.

A schematic overview comparing the pre-training schemes is given in Figure 1. In contrast to GSPT, SSPT does not offer guarantees on the correctness of the contained annotations. In addition, we expect the resulting corpus to contain few hard cases, as those probably have low confidence scores in the CRF-classification and are thus discarded. On the other hand, SSPT yields a far bigger pre-training corpus. We thus expect that the SSPT corpus captures a wider variety of language possibly allowing the model to generalize better to unseen sentences at test time, but possibly a lower variety of entities. Note that the confidence threshold in the filtering step allows to regulate the size and quality of the pre-training

[6] Using CRFSuite, see http://www.chokkan.org/software/crfsuite/

Table 2. Statistics of the pre-training corpora for gold (GSPT) and silver standard pre-training (SSPT). For SSPT only sentences containing at least one entity have been counted.

	Gold standard			Silver standard		
Entity type	Sentences	Mentions of annotated entities	Unique entities	Sentences	Mentions of annotated entities	Unique entities
Chemicals	281,883	287,972	70,370	1,174,573	1,402,866	59,090
Diseases	205,042	29,929	4,799	755,768	803,292	15,663
Species	26,357	4,772	1,141	252,404	258,313	3,612
Gene/Protein	88,863	87,015	19,985	21,446	23,644	2,979
Cell line	19,853	7,000	2,105	25,666	26,536	3,284

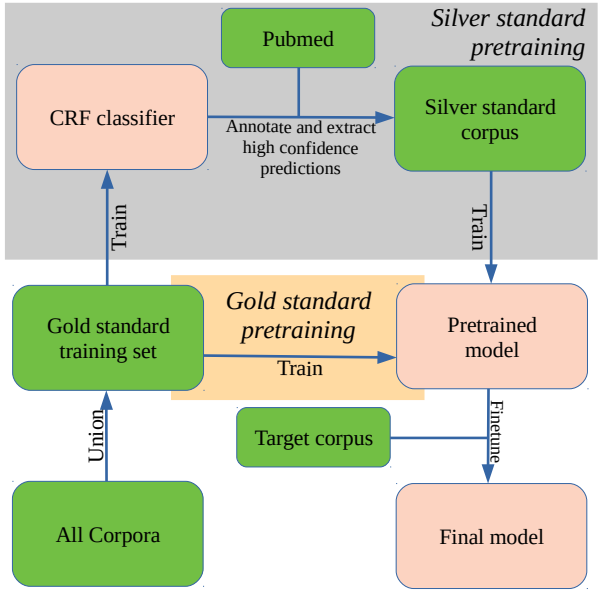

Fig. 1. Schematic overview of the two pre-training schemes. The components exclusive to silver and gold standard pre-training are highlighted. Corpora are encoded as green boxes and models as light red ones.

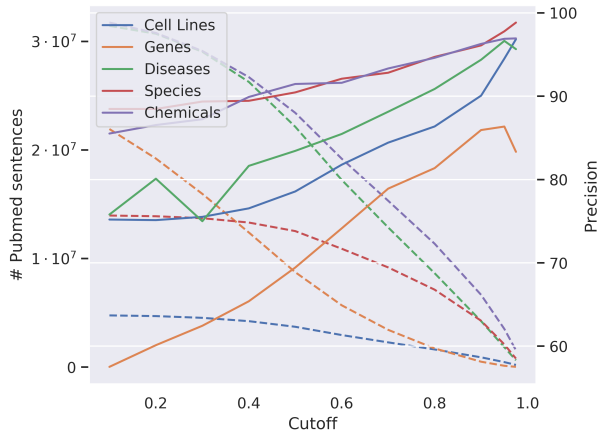

Fig. 2. Precision of the CRF classifier and number of sentences containing the entity types for the five entity types at different cutoff values. Precision is shown as solid lines, number of sentences as dashed lines of corresponding color.

corpus. While a high threshold should result in high precision at lower recall and a comparably small training set, a low threshold should yield lower precision at higher recall and a larger training set.

3 Results

3.1 Precision of the CRF

We first analyzed the precision of the CRF classifier used to generate the silver standard corpus to be able to explore the trade-off between the expected quality of the pre-training data and its size and diversity in terms of different entity names. Figure 2 shows the precision of the classifier on the test sets of all corpora of the respective entity type for different cutoff values. In all but one case precision approaches a value between 95% and 100% when the cutoff approaches 1. For genes it reaches only about 85%. As expected, higher precision comes at the cost of reduced corpus size, which shrinks nearly linearly with the cutoff for most entity types (diseases, chemicals, cell lines). For species names, the number of low confidence prediction is relatively low, leading to a considerable lower drop in number. The opposite can be observed for genes, where high confidence predictions are rare, which corresponds to the overall lower quality of gene name recognition.

Based on these results, we used a common cutoff value 0.95 for all entity types to generate our silver standard corpus for all subsequent experiments, as this value represents a reasonable trade off between corpus size and precision. This implies that we obtain a comparably small pre-training corpus for genes, while obtaining a comparably big one for species. As the pre-training step is computationally extremely expensive (one run takes on average about 8 days on an NVIDIA GTX 1080 GPU), we did not conduct further investigations on the influence of this cutoff on the performance of SSPT.

3.2 Comparison of SSPT with GSPT

We used the SSPT and GSPT schemes to pre-train entity-specific LSTM-CRF models, which were then fine tuned on the training sets of the different evaluation corpora. The final models were then evaluated on the respective test sets. We used the LSTM-CRF without pre-training as a baseline. Macro averaged results of both approaches can be found in Table 3, while results for individual corpora are listed in SM 1. In addition results are shown in Figure 3. For all five entity types, both pre-training schemes increase the average F1 score, with improvements ranging from 1.20 to 3.96 percentage points depending on the entity type. For four entity types, both schemes improve both precision and recall; the exception are cell lines where pre-training improves only recall. In general, gains in recall are more pronounced than in precision. Only for one corpus (Variome Disease), both pre-training methods lead to lower F1 scores than random initialization, with a decrease of 0.4 pp for GSPT and of 1.1 pp for SSPT. Very high gains in F1 can be achieved, for instance, on the CellFinder corpus for cell lines (+9.61pp / +9.2pp for GSPT / SSPT), the SCAI corpus for chemicals (+8.88pp / + 7.04pp), or the miRNA corpus for genes (+9.31pp / +8.42pp).

The overall and entity-type specific differences between SSPT and GSPT are small. GSPT tends to give higher improvements in terms of recall, whereas SSPT achieves higher gains in precision. GSPT leads to an increase in F1 score on 26 of the 34 corpora and SSPT on 30 of 34. This is an interesting observation, as it means that pre-training can safely re-use the biomedical corpora which are already available; taking the extra effort

Table 3. Macro averaged precision-, recall-, and F1-scores per entity type for no pre-training (Base), GSPT, and SSPT.

	Precision (%)			Recall (%)			F1-score (%)		
	Base	GSPT	SSPT	Base	GSPT	SSPT	Base	GSPT	SSPT
Chemicals	82.26	**84.39**	83.51	82.30	85.09	**85.67**	82.22	**84.68**	84.56
Diseases	81.07	**82.09**	81.77	79.48	**81.88**	81.30	80.23	**81.89**	81.43
Species	82.46	83.29	**85.30**	82.60	**87.22**	86.08	82.15	84.98	**85.60**
Gene/protein	80.30	80.60	**80.61**	77.88	**82.42**	82.03	78.86	**81.37**	81.24
Cell lines	**83.64**	82.02	82.48	69.77	75.20	**76.56**	75.09	77.96	**79.04**
Average	81.56	82.18	**82.38**	78.82	**82.79**	82.62	79.83	82.28	**82.37**

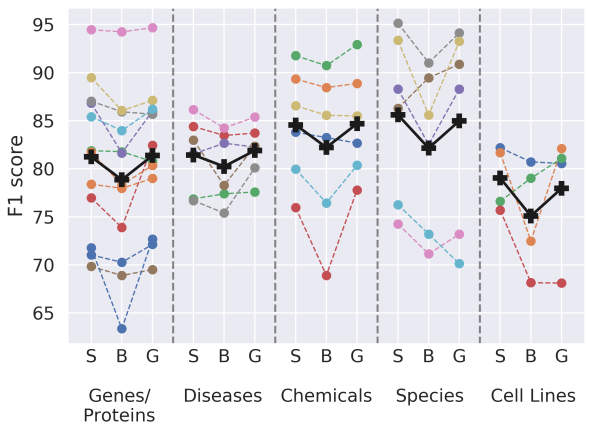

Fig. 3. F1 score for different pre-training schemes grouped by entity type for baseline (B), SSPT (S) and GSPT (G). Results for the same corpus are connected by lines. Mean values per entity type are shown in black.

to create a silver standard corpus does not seem to pay-off. On a single GTX 1080 GPU, one epoch of GSPT takes between 12 minutes (for cell lines) and 3 hours (for chemicals), while an epoch of SSPT takes around 40.5 hours. Creating the SSPT corpus took around a week on a machine with four Intel Xeon E7-4870 CPUs and a total of 80 threads.

3.3 Effects of Fine-tuning

To investigate the effects of fine-tuning on the target corpus, we also tested to directly use the SSPT and the GSPT models as NER tools, omitting the fine-tuning step. Aggregated results per entity type are presented in Table 4, while results for individual corpora can be found in SM 1. In most cases, pure SSPT leads to a model that is precise but has a low recall. This matches our expectation, as our usage of only high confidence predictions reduces diversity. Thus, the model is not good at identifying difficult entities but offers very good performance in reducing false positives. In contrast, GSPT yields a model with fairly balanced precision and recall.

It is very instructive to compare the results of GSPT with that of the traditional approach to NER (no pre-training, only learning on the target corpus; see column "no" in Table 3 and column "GSPT" in Table 4). In 12 cases, GSPT without any adaptation to the target corpus achieves a result that is within 3pp of the corpus-specific models. For 5 corpora, GSPT actually outperfoms the corpus-specific model (SCAI chemicals, CellFinder species, s800 species, LocText species, and Linneaus species). On the other hand, GSPT sometimes is drastically worse, with extreme cases being LocText genes (-33pp) and OSIRIS genes (-20pp) and DECA genes (-20pp). Performance on genes is generally much lower than corpus-specific training, which corresponds to the lower quality of the GSPT corpus itself (see above).

3.4 HUNER: An off-the-shelf biomedical NER tool

Overall, we found the results for using GSPT without fine tuning highly encouraging. Note that in applications, users often do not have specific training data at hand but require ready-to-use NER tools. Clearly, the previous section showed that the expected performance is lower than when using specifically annotated training data, but creating such resources is costly and time-consuming. Furthermore, there are many applications for NER being applied on all available texts to create a global view on the current state-of-knowledge on specific entity. Examples of such tools are PubTator (Wei *et al.*, 2013) or GeneView (Thomas *et al.*, 2012). Such applications depend on NER tools that should be as much unbiased as possible, e.g. should not depend on any specific gold standard.

To support such applications, we created HUNER, a stand-along biomedical NER tool for five different entity types. HUNER wraps the entity-specific GSPT models into easy-to-install and easy-to-use Docker containers. These containers incorporate all software dependencies of HUNER and provide a REST API that allows tagging of arbitrary sentences with the pre-trained models. Additionally, we make a Python and a command-line client for the tagging server available. To illustrate the simplicity of using HUNER, the following recipe shows the five steps necessary to apply it.

1. Install Docker
2. Download code and models from `https://github.com/hu-ner/huner`
3. Start the annotation server for a given entity type: `./start_server.sh MODEL_NAME`
4. Annotate the text: `python tagger.py text.txt text.conll --name MODEL_NAME`

HUNER expects the input file to contain one document per line and outputs the tagged text as a file in CoNLL2003 format. Pre-tokenized or Pre-sentence-split input is also supported using flags.

To further investigate the performance of HUNER, we run it on the CRAFT corpus (Bada *et al.*, 2012)[7] and compared its performance to that of GNormPlus (Wei *et al.*, 2015) and tmChem (Leaman *et al.*, 2015), the tools used to provide PubTator. We compare on all present entity types, namely chemicals, species and genes/proteins. Results are shown in Table5. HUNER achieves a considerable higher precision for chemicals and species and a considerable higher recall and F1 score on all three entity types. Differences are particularly high for F1 (+9.58pp) on genes, caused by the much improved recall (+21.78pp). HUNER is also very fast; on comparable machines (both tools are single-threaded), HUNER requires 11ms (11ms) for detecting genes (species) per sentence, whereas GNormPlus requires 390ms (50ms). In contrast, tmChem is almost four times faster than HUNER for chemicals (3ms versus 11ms per sentence).

[7] See SM 4 for some particularities of this corpus. Note that this corpus was not used for any other experiment and has no overlap with the other corpora used in this work.

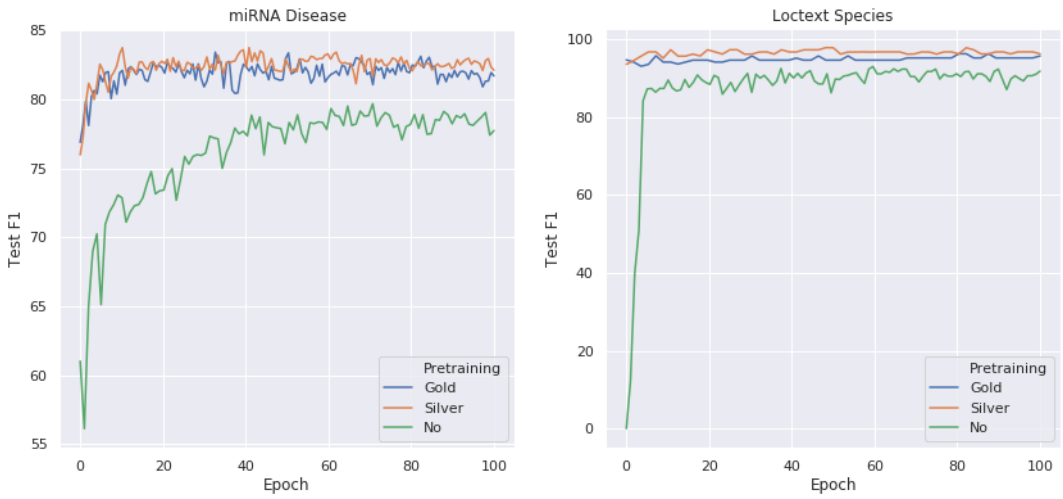

Fig. 4. Trajectories of F1-score on the test set over training time. While the randomly initialized network (No) needs at least a few epochs to reach near-optimal performance, both the models pre-trained with gold standard (Gold) and Silver Standard pre-training do so almost instantly. Note that the y-axes differ between the two diagrams.

Table 4. Macro averaged precision-, recall-, and F1-scores per entity type without fine-tuning on target corpus. The results after fine-tuning are provided in parentheses for convenience.

	Precision (%)		Recall (%)		F1-score (%)	
	GSPT	SSPT	GSPT	SSPT	GSPT	SSPT
Chemicals	**83.34** (84.39)	80.63 (83.51)	**80.26** (85.09)	70.26 (85.67)	**81.71** (84.68)	74.98 (84.56)
Diseases	75.01 (82.09)	**79.41** (81.77)	**77.71** (81.88)	65.10 (81.30)	**76.20** (81.89)	71.26 (81.43)
Species	85.37 (83.29)	**90.01** (85.30)	**79.98** (87.22)	60.29 (86.08)	**82.59** (84.98)	71.33 (85.60)
Gene/protein	**75.01** (80.60)	61.72 (80.61)	**79.16** (82.42)	13.30 (82.03)	**76.81** (81.37)	20.84 (81.24)
Cell lines	**65.09** (82.02)	64.26 (82.48)	**67.69** (75.20)	38.16 (76.56)	**66.08** (77.96)	47.34 (79.04)
Average	**79.84** (82.18)	67.8 (82.38)	**74.86** (82.79)	52.46 (82.62)	**73.33** (82.28)	57.75 (82.37)

Note that we only compare the performance in entity recognition, as HUNER only performs this step, whereas GNormPlus and tmChem also perform entity normalization. This important difference of course also must be taken into account when selecting a tool for large-scale biomedical NER. In any application where recognized genes are used for down-stream analysis where their appearances must be combined with other gene-specific values (e.g. expression data, sequence data etc.), entity normalization is mandatory and therefore HUNER in its present form is inapplicable.

4 Discussion

4.1 Effect of Pre-training

For all five entity types we evaluated, both pre-training schemes on average improved recall and F1 score when compared to a pure corpus-specific training. Also precision was improved for all entity types but cell lines. Improvements are also visible at the corpus level. Especially recall improved in most cases for both schemes, and often by a large margin. We attribute this observation mostly to the small size of most corpora. A major problem with small corpora is that (a) entities present in the test set often are not present in the training set, and (b) the number of examples in the training set is too small to learn robust abstractions. By considering additional, though not perfect, data, the pre-training schemes SSPT and GSPT see more and more diverse examples, which (a) reduces the probability of unseen entities in the test set and (b) helps to learn more general models. In contrast, pure corpus-specific training often achieves a higher precision, which is to be expected as corpus-specific guidelines play a much higher role in this setting.

4.2 Precision degradation for cell lines

We performed an in-depth analysis regarding the inferior precision of pre-training on cell lines. A first observation is that cell lines is the entity type with by far the smallest amount of data. This is equally true for number of corpora, the number of sentences, and the number of annotated entities. Furthermore, more then half of the total number of sentences are from the JNLPBA corpus, whereas CLL and CellFinder are rather small. There are also strong differences in the annotation density: The average number of entities per sentence is around 1.7 for CLL, 0.56 for JNLPBA, 0.17 for CellFinder and only 0.06 for Gellus, i.e., cell lines are highly enriched in CLL and underrepresented in Gellus (compared to the other corpora). In addition, the average length of annotated tokens varies considerably. Although only 50% of sentences are from JNLPBA, this corpus accounts for 92% of all tokens annotated as cell line mention. Accordingly, the pre-training data of both schemes is heavily dominated by the JNLPBA corpus, and precision on Gellus and, especially, CellFinder is much inferior with pre-training than without. This dominance also leads to a catastrophic performance of SSPT without fine-tuning for this entity type for all corpora except JNLPBA, with average F1-scores around 2.5%. Overall, we conclude the diversity and size of corpora for cell lines are not yet large enough to out-weight the dominance of the JNLPBA corpus and guidelines.

Table 5. Performance of HUNER and off-the-shelf state of the art tools on the CRAFT corpus

	Precision (%)			Recall (%)			F1-score (%)		
Entity type	HUNER	GNormPlus	tmChem	HUNER	GNormPlus	tmChem	HUNER	GNormPlus	tmChem
Chemicals	**53.56**	–	49.74	**35.85**	–	31.43	**42.95**	–	38.52
Species	**98.51**	87.03	–	**73.83**	69.51	–	**84.40**	77.29	–
Gene/Protein	56.67	**65.03**	–	**62.33**	40.55	–	**59.37**	49.95	–

4.3 Effects on Convergence

A side-effect of pre-training is that it considerably improve the convergence behavior of training. Exemplary trajectories of the test set performance of two corpora are displayed in Figure 4. The pre-trained models reach near-optimal performance almost instantly, while it takes the randomly initialized models a good number of epochs to converge. The plots also shows the different level at which the models converge. Variance over time is lower for the pre-trained models than for the randomly initialized ones, which is advantageous as it makes the final result less dependent of the particular choice of epochs to use.

4.4 Impact of corpus size on performance without fine-tuning

We analyzed the effect of relative corpus size on the performance in GSPT without fine-tuning. We defined the relative corpus size as the percentage of train sentences from a given corpus in the GSPT train corpus. The results are shown in Figure SM 5. In contrast to our intuition we did not observe a significant connection between relative corpus size and GSPT performance without fine-tuning. We suspect multiple reasons for this behaviour. First the relative corpus size is not the optimal measure, as corpora might have similar annotation guidelines, leading to good performance on small corpora. Second different corpora might be intrinsically differently hard to annotate due to their annotation guidelines. Third results on small corpora underlie a high variance, making the observations less precise.

4.5 Comparison to previous works

As mentioned in the introduction, Giorgi and Bader (2018) also reported results for pre-training a LSTM-CRF model for a subset of the corpora presented here. As this work used a different pre-training corpus, it is interesting to compare the overall results with that of our GSPT and SSPT schemes; see SM 2. Overall, the improvements are fairly consistent. They achieve notably (more than 3pp difference) better F1-scores compared to our silver-standard pre-training for Variome-diseases, CellFinder-genes, BioInfer-genes, miRNA-genes, and miRNA-species, but lower results for miRNA-diseases, Variome-genes, and LocText-genes. However, we note that Georgi and Bader used a different train/dev/test splits; to the best of our knowledge, these are not published. We discuss the differences between our baseline and our previous results from Habibi *et al.* (2017) in SM 3.

5 Conclusion

We propose two different pre-training schemes for BioNER and evaluate their effect on predictive performance across 34 corpora. We find that both pre-training schemes lead to improvements in average F1-score for all entity types. Furthermore, we found that the model pre-trained with GSPT shows good performance even without fine-tuning and strongly outperforms the state-of-the-art BioNER tools GNormPlus and tmChem on an held-out corpus. Therefore, we make the pre-trained models publicly available as HUNER, an easy-to-use of the shelf BioNER tool.

As this paper analyzed only basic variants of pre-training, there are plenty of options for future research. Especially more refined techniques for transfer learning could be applied. Examples would be neural adversarial domain adaptation as in (Rios *et al.*, 2018) or universal language model fine-tuning as in (Howard and Ruder, 2018). One could also try incorporating already available pre-trained representations of language like ELMO (Peters *et al.*, 2018), InferSent (Conneau *et al.*, 2017) or BERT (Devlin *et al.*, 2018).

Furthermore, we do not address the problem of Named Entity Normalization (NEN), which is an integral part of many biomedical text mining pipelines like PubTator (Wei *et al.*, 2013). It would be worthwhile to investigate whether the proposed pre-training techniques lead to improvements in NEN.

Acknowledgements

Leon Weber and Jannes Münchmeyer acknowledge the support of the Helmholtz Einstein International Berlin Research School in Data Science (HEIBRiDS). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

References

Akhondi, S. A. *et al.* (2014). Annotated chemical patent corpus: a gold standard for text mining. *PloS one*, **9**(9), e107477.

Bada, M. *et al.* (2012). Concept annotation in the craft corpus. *BMC bioinformatics*, **13**(1), 161.

Bagewadi, S. *et al.* (2014). Detecting mirna mentions and relations in biomedical literature. *F1000Research*, **3**.

Conneau, A. *et al.* (2017). Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Dai, A. M. and Le, Q. V. (2015). Semi-supervised sequence learning. In *Advances in neural information processing systems*, pages 3079–3087.

Devlin, J. *et al.* (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ding, J. *et al.* (2001). Mining medline: abstracts, sentences, or phrases? In *Biocomputing 2002*, pages 326–337. World Scientific.

Doğan, R. I. *et al.* (2014). Ncbi disease corpus: a resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, **47**, 1–10.

Furlong, L. I. *et al.* (2008). Osirisv1. 2: a named entity recognition system for sequence variants of genes in biomedical literature. *BMC bioinformatics*, **9**(1), 84.

Gerner, M. *et al.* (2010). Linnaeus: a species name identification system for biomedical literature. *BMC bioinformatics*, **11**(1), 85.

Giorgi, J. M. and Bader, G. D. (2018). Transfer learning for biomedical named entity recognition with neural networks. *Bioinformatics*.

Goldberg, T. *et al.* (2015). Linked annotations: a middle ground for manual curation of biomedical databases and text corpora. In *BMC proceedings*, volume 9, page A4. BioMed Central.

Gurulingappa, H. *et al.* (2010). An empirical evaluation of resources for the identification of diseases and adverse effects in biomedical literature. In *2nd Workshop on Building and evaluating resources for biomedical*

text mining (7th edition of the Language Resources and Evaluation Conference).

Habibi, M. *et al.* (2017). Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*, **33**(14), i37–i48.

Hahn, U. *et al.* (2010). A proposal for a configurable silver standard. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 235–242. Association for Computational Linguistics.

Hakala, K. *et al.* (2016). Syntactic analyses and named entity recognition for pubmed and pubmed central—up-to-the-minute. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, pages 102–107.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, **9**(8), 1735–1780.

Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 328–339.

Kaewphan, S. *et al.* (2015). Cell line name recognition in support of the identification of synthetic lethality in cancer from text. *Bioinformatics*, **32**(2), 276–282.

Kafkas, S. *et al.* (2012). Calbc: Releasing the final corpora. In *LREC*, pages 2923–2926.

Kim, J.-D. *et al.* (2004). Introduction to the bio-entity recognition task at jnlpba. In *Proceedings of the international joint workshop on natural language processing in biomedicine and its applications*, pages 70–75. Association for Computational Linguistics.

Kolárik, C. *et al.* (2008). Chemical names: terminological resources and corpora annotation. In *Workshop on Building and evaluating resources for biomedical text mining (6th edition of the Language Resources and Evaluation Conference)*.

Krallinger, M. *et al.* (2015a). Chemdner: The drugs and chemical names extraction challenge. *Journal of cheminformatics*, **7**(1), S1.

Krallinger, M. *et al.* (2015b). Overview of the chemdner patents task. In *Proceedings of the fifth BioCreative challenge evaluation workshop*, pages 63–75.

Lafferty, J. *et al.* (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *International Conference on Machine Learning (ICML)*, pages 282–289.

Lample, G. *et al.* (2016). Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Leaman, R. *et al.* (2009). Enabling recognition of diseases in biomedical text with machine learning: corpus and benchmark. In *Proceedings of the 2009 Symposium on Languages in Biology and Medicine*, volume 82.

Leaman, R. *et al.* (2015). tmchem: a high performance approach for chemical named entity recognition and normalization. *Journal of cheminformatics*, **7**(1), S3.

Li, J. *et al.* (2016). Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database*, **2016**.

Mikolov, T. *et al.* (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Min, S. *et al.* (2017). Question answering through transfer learning from large fine-grained supervision data. *arXiv preprint arXiv:1702.02171*.

Neves, M. *et al.* (2012). Annotating and evaluating text for stem cell research. In *Proceedings of the Third Workshop on Building and Evaluation Resources for Biomedical Text Mining (BioTxtM 2012) at Language Resources and Evaluation (LREC). Istanbul, Turkey*, pages 16–23.

Pafilis, E. *et al.* (2013). The species and organisms resources for fast and accurate identification of taxonomic names in text. *PLoS One*, **8**(6), e65390.

Pan, S. J. *et al.* (2010). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, **22**(10), 1345–1359.

Peters, M. E. *et al.* (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Pyysalo, S. *et al.* (2007). Bioinfer: a corpus for information extraction in the biomedical domain. *BMC bioinformatics*, **8**(1), 50.

Pyysalo, S. *et al.* (2013). Distributional semantics resources for biomedical text processing. In *Proceedings of the 5th International Symposium on Languages in Biology and Medicine, Tokyo, Japan*, pages 39–43.

Ramachandran, P. *et al.* (2016). Unsupervised pretraining for sequence to sequence learning. *arXiv preprint arXiv:1611.02683*.

Rios, A. *et al.* (2018). Generalizing biomedical relation classification with neural adversarial domain adaptation. *Bioinformatics*, **1**, 9.

Smith, L. *et al.* (2008). Overview of biocreative ii gene mention recognition. *Genome biology*, **9**(2), S2.

Thomas, P. *et al.* (2012). Geneview: a comprehensive semantic search engine for pubmed. *Nucleic acids research*, **40**(W1), W585–W591.

Tikk, D. *et al.* (2010). A comprehensive benchmark of kernel methods to extract protein–protein interactions from literature. *PLoS computational biology*, **6**(7), e1000837.

Tjong Kim Sang, E. F. and De Meulder, F. (2003). Introduction to the conll-2003 shared task: Language-independent named entity recognition. In W. Daelemans and M. Osborne, editors, *Proceedings of CoNLL-2003*, pages 142–147. Edmonton, Canada.

Verspoor, K. *et al.* (2013). Annotating the biomedical literature for the human variome. *Database*, **2013**.

Wang, X. *et al.* (2010). Disambiguating the species of biomedical named entities using natural language parsers. *Bioinformatics*, **26**(5), 661–667.

Wei, C.-H. *et al.* (2013). PubTator: a web-based text mining tool for assisting biocuration. *Nucleic Acids Res.*, **41**(W1), W518–W522.

Wei, C.-H. *et al.* (2015). Gnormplus: an integrative approach for tagging genes, gene families, and protein domains. *BioMed research international*, **2015**.