A large-scale cross-corpus performance analysis of pre-training on NER in the biomedical domain

Exposé

Jan-Christopher Pien

07.02.2019

1 Motivation and background

In 2017 alone, 813,598 citations were indexed in MEDLINE[1]. This vast amount of information cannot be processed by humans alone. Thus, researchers have continuously tried to design and optimize automated systems that perform information extraction (IE) in the biomedical domain. Named entity recognition (NER) constitutes an important part of natural language processing (NLP), especially in the biomedical domain. It can for example be used to extract names of chemicals, diseases, species, genes/proteins and cell lines from abstracts or full texts [5].

While continuous performance gains have been reported using a single corpus for training and evaluating (see section 3), there has been little research on cross-corpus evaluating. Cross-corpus is in this context understood as training a model on one or more corpora and evaluating it on a different corpus. Cross-corpus evaluation usually leads to significant performance losses, because the corpora differ in annotation style or entity distribution [14], or because the model overfits to one corpus and does not generalize well to other corpora [19].

2 Research goal

The research goal of this work is to examine whether using pre-training can improve performance of cross-corpus NER in the biomedical domain. Specifically, we will study using different pre-trained context-sensitive word embeddings as well as pre-training on gold- and silver-standard corpora and the impact these different pre-training methods have on the performance of training NER on one corpus and evaluating on other corpora in the biomedical domain.

[1] `https://www.nlm.nih.gov/bsd/index_stats_comp.html`

The evaluation corpora used by Habibi et al. [5] for their large-scale evaluation of NER in the biomedical domain provide a large data base for performing this examination, because they consist of multiple corpora for each of the five different entity types (see also section 5).

3 Related work

3.1 Neural architectures for NER

In [9], Lample et al. have developed a bi-directional Long Short-term Memory Network (LSTM) with a Conditional Random Fields (CRF) layer on top to perform NER on general-domain texts across different languages. The CRF model, developed by Lafferty, McCallum, and Pereira [8], is a probabilistic model that can be used for sequence labeling problems. Lample et al. have been one of the first to propose a NN model without handcrafted features to perform NER. Instead, they use word embeddings composed of a character-based word representation and a distributional representation based on the *word2vec* model [11].

3.2 NER in the biomedical domain

While rule-based systems have been used for many years in NER in the biomedical domain [13], recently neural architectures similar to the one above have successfully been implemented.

In [5], Habibi et al. have extended the above model to the biomedical domain, using word embeddings either trained on domain specific texts or on a combination of domain specific and general domain texts as an input to the LSTM models. They have achieved an average performance improvement of 5% compared to their chosen baseline models (entity-specific NER tools).

In [4], Giorgi and Bader have used a similar architecture, but extended the method to first train a NN on a large domain-specific silver-standard corpus (SSC) and then transfer this network to a smaller gold-standard corpus (GSC). They have achieved an average error rate reduction of 11% compared to their baseline (no pre-training on a SSC).

3.3 Transfer learning

In computer vision, transfer learning has had overwhelming success. The ImageNet, originally developed by Deng et al. [3], is a large and unspecific collection of labelled images. Using this database, many authors have improved performance in specialized tasks in computer vision [15]. In [19], Yosinski et al. have examined how transferring the features from deep neural networks trained on natural images to a target task can improve performance because it enables transfer of knowledge and can reduce the risk of overfitting. They concluded that transferring low-level features improves performance because they capture general properties of the images. At higher-level layers of the model, however, the specificity towards the task to be performed increases, thus limiting the transferability of these features.

Mou et al. have taken a similar approach and examined whether the same holds true for NLP tasks by conducting two series of experiments: sentence classification using a LSTM architecture and sentence-pair classification using two CNNs for each sentence. They have reported similar observations, however they have observed the limitation that transferability mainly depends on the semantical similarity between the source and the target task [12].

Howard and Ruder have trained a language model (LM) using an architecture developed by Merity, Keskar, and Socher [10] on a large corpus based on 28,595 preprocessed general-domain Wikipedia articles (ULMFiT). They use the word representations this LM produces as input to a subsequent classification layer to perform their text classification target task. They have reported significant reduces in error rates using their specific fine-tuning techniques.

3.4 Transfer learning for NER

With flair, Akbik, Blythe, and Vollgraf [1] have released a framework that uses a similar LM architecture as ULMFiT, but uses characters and not words as input. They have used the word representations of this LM as input to a CRF layer in a similar fashion as Lample et al. [9] to perform NER. They have achieved significant performance improvements on English and German NER, reporting a new state-of-the-art on the CoNLL03 shared task.

Sachan et al. have trained a bi-directional LM on a large unlabeled medical domain corpus and transferred its weights to a NER model with the same architecture as the LM [16]. They have observed small improvements in F1-score compared to using no pre-training.

Weber et al. have examined how pre-training a neural NER model for the biomedical domain similar to [5] on silver- or gold-standard corpora improves performance by about 2.5%. They have created a silver-standard corpus by training a CRF NER tagger on the union of all training sets of their gold-standard corpora (the same we use in this work). They then use this model to annotate a large corpus of PubMed abstracts and use this as the pre-training corpus for their neural NER model. They contrast this method with just using the union of all training sets of their gold-standard corpora as a pre-training corpus.

3.5 Cross-corpus NER

Nothman, Murphy, and Curran [14] have created a large corpus using Wikipedia articles and used this corpus to improve cross-corpus NER on the MUC-7 Named Entity Task, the English CoNLL-03 Shared Task and the BBN Pronoun Coreference and Entity Type Corpus. They have reported performance gains of up to 11% compared to using just a single gold-standard corpus. Additionally, they have performed a thorough corpus analysis, giving explanations for poor performance in cross-corpus NER.

4 Intended approach

The architecture of the NER model will follow [5]. They feed word embeddings of the tokens of the sentence to be predicted into a single-layer bi-directional LSTM and use a CRF layer to classify each token. This has been shown a successful method for single corpus NER in the biomedical domain [5, 4]. We will use this model to produce baseline results on which we can report improvements.

For cross-corpus evaluation, we will evaluate each of the five entity classes separately. A model will be trained on the training set of each of corpora in an entity class, selected on the validation set of this corpus and evaluated on all the test stets of all other corpora in this entity class.

4.1 Pre-training using contextual word embeddings

Following the approach developed by [7], who are using a LM pretrained using a technique developed by Merity, Keskar, and Socher [10], we will pretrain both a forward and a backward LM on a large base corpus (see section 5).

Merity, Keskar, and Socher showed that vanilla LSTMs with specific regularization techniques and adaptations to the optimization algorithm are able to outperform much more complicated LM architectures. The specific weight-drop mechanism being used is called DropConnect and was developed by Wan et al. [17]. Moreover, instead of a simple stochastic gradient descent (SGD) optimization algorithm, Merity, Keskar, and Socher introduce a variant called "Non-monotonically Triggered Averaged SGD" (NT-ASGD) [10], which dynamically averages the gradients when the validation metric does not improve for a predefined number of epochs. They also employ a number of other regularization techniques which are described in detail in [10].

Howard and Ruder have shown that this LM can capture enough information to perform well on text classification tasks [7]. They have used the concatenation of the hidden state of the LM at the last time step of the input sequence, as well as the max-pooled and mean-pooled representation of the hidden states of the LM of the input sequence as input to a linear block that classifies the text.

For our purposes, we will use a similar architecture as [1]: we will use the representations of each token of the input sequence output by the LM as word embeddings in our baseline model. Since these word embeddings depend on the context of the whole input sequence, we call these "contextual word embeddings". We will use both the forward and the backward LM and concatenate the outputs of both models.

Following the approach in [1], we will also train a forward and backward character-level LM on our base corpus and use the outputs of these models in a similar fashion.

4.2 Silver- and gold-standard pre-training

As outlined in subsection 3.4, we will use a silver- and gold-standard pre-training corpus to pre-train our model. Instead of fine-tuning and evaluating on the same corpus as in [18], we will fine-tune on one corpus and evaluate on all different corpora for each entity type. To avoid information leakage, the respective corpora will be removed from the gold-standard pre-training corpus.

4.3 Corpus and error analysis

For a thorough corpus and error analysis, we will follow methods developed by [14] to examine inconsistencies in the different corpora belonging to one entity class. For example, they find all n-grams appearing multiple times in one corpus that have different tags for some sub-sequence of the n-gram. Also, they perform simple entity type frequency analysis, grouped by POS tags or "wordtypes", a technique described by Collins [2] to classify words based on their lexicographical structure.

Moreover, we will examine how often out-of-vocabulary (OOV) errors occur in the different corpora and how overlapping the entity vocabulary between different corpora is.

5 Corpora

For our LM base corpus, we will use a large corpus provided by Hakala et al. [6]. This corpus consists of 26 million abstracts and 1.4 million full articles, covering the whole of PubMed and PubMed Central Open Access articles up to publication in 2015.

As target corpora for NER we will use the same corpora used by [5]. The 34 different corpora are manually annotated with one of five entity classes: chemicals, diseases, species, genes/proteins or cell line. The corpora differ greatly in size, ranging from 215 to 173,808 sentences. We use 6 corpora for the chemical entity class, 6 corpora for the disease entity class, 6 corpora for the species entity class, 12 corpora for the genes/proteins entity class and 4 corpora for the cell lines entity class. For a complete description of these corpora see [5].

References

[1] Alan Akbik, Duncan Blythe, and Roland Vollgraf. "Contextual String Embeddings for Sequence Labeling". In: *COLING 2018, 27th International Conference on Computational Linguistics*. 2018, pp. 1638–1649.

[2] Michael Collins. "Ranking algorithms for named-entity extraction". In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*. Ed. by Pierre Isabelle. Morristown, NJ, USA: Association for Computational Linguistics, 2002, p. 489. DOI: 10.3115/1073083.1073165.

[3] Jia Deng et al. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[4] John M. Giorgi and Gary D. Bader. "Transfer learning for biomedical named entity recognition with neural networks". In: *Bioinformatics* 34 (2018), pp. 1–8. DOI: 10.1093/bioinformatics/bty449.

[5] Maryam Habibi et al. “Deep learning with word embeddings improves biomedical named entity recognition”. In: *Bioinformatics* 33.14 (2017), pp. i37–i48. DOI: `10.1093/bioinformatics/btx228`.

[6] Kai Hakala et al. “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*. Ed. by Kevin Bretonnel Cohen et al. Stroudsburg, PA, USA: Association for Computational Linguistics, 2016, pp. 102–107. DOI: `10.18653/v1/W16-2913`.

[7] Jeremy Howard and Sebastian Ruder. “Universal Language Model Fine-tuning for Text Classification”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, 2018, pp. 328–339.

[8] John Lafferty, Andrew McCallum, and Fernando C.N. Pereira. “Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data”. In: *ICML ’01 Proceedings of the Eighteenth International Conference on Machine Learning*. Ed. by Carla E. Brodley and Andrea Pohoreckyj Danyluk. San Francisco, CA, USA: Morgan Kaufmann Publishers, 2001, pp. 282–289.

[9] Guillaume Lample et al. “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*. Association for Computational Linguistics, 2016, pp. 260–270. DOI: `10.18653/v1/N16-1030`.

[10] Stephen Merity, Nitish Shirish Keskar, and Richard Socher. *Regularizing and Optimizing LSTM Language Models*. 2017. arXiv.org: `1708.02182v1`.

[11] Tomas Mikolov et al. “Distributed Representations of Words and Phrases and Their Compositionality”. In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*. NIPS’13. USA: Curran Associates Inc, 2013, pp. 3111–3119.

[12] Lili Mou et al. *How Transferable are Neural Networks in NLP Applications?* 2016. arXiv.org: `1603.06111v2`.

[13] Meenakshi Narayanaswamy, K. E. Ravikumar, and K. Vijay-Shanker. “A biological named entity recognizer”. In: *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing* 8 (2003), pp. 427–438. ISSN: 2335-6928.

[14] Joel Nothman, Tara Murphy, and James R. Curran. “Analysing Wikipedia and gold-standard corpora for NER training”. In: *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics on - EACL ’09*. Ed. by Alex Lascarides, Claire Gardent, and Joakim Nivre. Morristown, NJ, USA: Association for Computational Linguistics, 2009, pp. 612–620. DOI: `10.3115/1609067.1609135`. (Visited on 02/06/2019).

[15] Ali Sharif Razavian et al. *CNN Features off-the-shelf: an Astounding Baseline for Recognition*. 2014. arXiv.org: `1403.6382v3`.

[16] Devendra Singh Sachan et al. *Effective Use of Bidirectional Language Modeling for Transfer Learning in Biomedical Named Entity Recognition.* 2017. arXiv.org: `1711.07908v3`.

[17] Li Wan et al. "Regularization of Neural Networks using DropConnect". In: *Proceedings of the 30th International Conference on Machine Learning.* Ed. by Sanjoy Dasgupta and David McAllester. Vol. 28. Proceedings of Machine Learning Research. Atlanta, GA, USA: PMLR, 2013, pp. 1058–1066.

[18] Leon Weber et al. "Pretraining Improves Deep Learning for BiomedicalNamed Entity Recognition". In: *Bioinformatics* unpublished (2019).

[19] Jason Yosinski et al. "How transferable are features in deep neural networks?" In: *Advances in Neural Information Processing Systems 27* (2014).