NeuroNER: an easy-to-use program for named-entity recognition based on neural networks

Franck Dernoncourt* MIT francky@mit.edu

Ji Young Lee* MIT jjylee@mit.edu

Peter Szolovits MIT psz@mit.edu

Abstract

Named-entity recognition (NER) aims at identifying entities of interest in a text. Artificial neural networks (ANNs) have recently been shown to outperform existing NER systems. However, ANNs remain challenging to use for non-expert users. In this paper, we present NeuroNER, an easy-to-use named-entity recognition tool based on ANNs. Users can annotate entities using a graphical web-based user interface (BRAT): the annotations are then used to train an ANN, which in turn predict entities' locations and categories in new texts. NeuroNER makes this annotation-training-prediction flow smooth and accessible to anyone.

1 Introduction

Named-entity recognition (NER) aims at identifying entities of interest in the text, such as location, organization and temporal expression. Identified entities can be used in various downstream applications such as patient note de-identification and information extraction systems. They can also be used as features for machine learning systems for other natural language processing tasks.

Early systems for NER relied on rules defined by humans. Rule-based systems are time-consuming to develop, and cannot be easily transferred to new types of texts or entities. To address these issues, researchers have developed machine-learning-based algorithms for NER, using a variety of learning approaches, such as fully supervised learning, semi-supervised learning, unsupervised learning, and active learning. NeuroNER is based on a fully supervised learning algorithm, which is the most studied approach (Nadeau and Sekine, 2007).

Fully supervised approaches to NER include support vector machines (SVM) (Asahara and Matsumoto, 2003), maximum entropy models (Borthwick et al., 1998), decision trees (Sekine et al., 1998) as well as sequential tagging methods such as hidden Markov models (Bikel et al., 1997), Markov maximum entropy models (Kumar and Bhattacharyya, 2006), and conditional random fields (CRFs) (McCallum and Li, 2003; Tsai et al., 2006; Benajiba and Rosso, 2008; Filannino et al., 2013). Similar to rule-based systems, these approaches rely on handcrafted features, which are challenging and time-consuming to develop and may not generalize well to new datasets.

More recently, artificial neural networks (ANNs) have been shown to outperform other supervised algorithms for NER (Collobert et al., 2011; Lample et al., 2016; Lee et al., 2016; Labeau et al., 2015; Dernoncourt et al., 2016). The effectiveness of ANNs can be attributed to their ability to learn effective features jointly with model parameters directly from the training dataset, instead of relying on handcrafted features developed from a specific dataset. However, ANNs remain challenging to use for non-expert users.

Contributions NeuroNER makes state-of-the-art named-entity recognition based on ANN available to anyone, by focusing on usability. To enable users to create or modify annotations for a new or existing corpus, NeuroNER interfaces with the web-based annotation program BRAT (Stenetorp et al., 2012). NeuroNER makes the annotation-training-prediction flow smooth and accessible to anyone, while leveraging the state-of-the-art prediction capabilities of ANNs. NeuroNER is open source and freely available online[1].

* These authors contributed equally to this work.

[1] NeuroNER is available at: https://github.com/Franck-Dernoncourt/NeuroNER

2 Related Work

Existing publicly available NER systems geared toward non-experts do not use ANNs. For example, Stanford NER (Finkel et al., 2005), ABNER (Settles, 2005), the MITRE Identification Scrubber Toolkit (MIST) (Aberdeen et al., 2010), (Boag et al., 2015), BANNER (Leaman et al., 2008) and NERsuite (Cho et al., 2010) rely on CRFs. GAPSCORE uses SVMs (Chang et al., 2004). Apache cTAKES (Savova et al., 2010) and Gate's ANNIE (Cunningham et al., 1996; Maynard and Cunningham, 2003) use mostly rules. NeuroNER, the first ANN-based NER system for non-experts, is more generalizable to new corpus due to the ANNs' capability to learn effective features jointly with model parameters.

Furthermore, in many cases, the NER systems assume that the user already has an annotated corpus formatted in a specific data format. As a result, users often have to connect their annotation tool with the NER systems by reformatting annotated data, which can be time-consuming and error-prone. Moreover, if users want to manually improve the annotations predicted by the NER system (e.g., if they use the NER system to accelerate the human annotations), they have to perform additional data conversion. NeuroNER streamlines this process by incorporating BRAT, a widely-used and easy-to-use annotation tool.

3 System Description

NeuroNER comprises two main components: an NER engine and an interface with BRAT. NeuroNER also comes with real-time monitoring tools for training, and pre-trained models that can be loaded to the NER engine in case the user does not have access to labelled training data. Figure 1 presents an overview of the system.

3.1 NER engine

The NER engine takes as input three sets of data with gold labels: the training set, the validation set, and the test set. Additionally, it can also take as input the deployment set, which refers to any new text without gold labels that the user wishes to label. The files that comprise each set of data should be in the same format as used for the annotation tool BRAT or the CoNLL-2003 NER shared task dataset (Tjong Kim Sang and De Meulder, 2003), and organized in the corresponding folder.

The NER engine's ANN contains three layers:

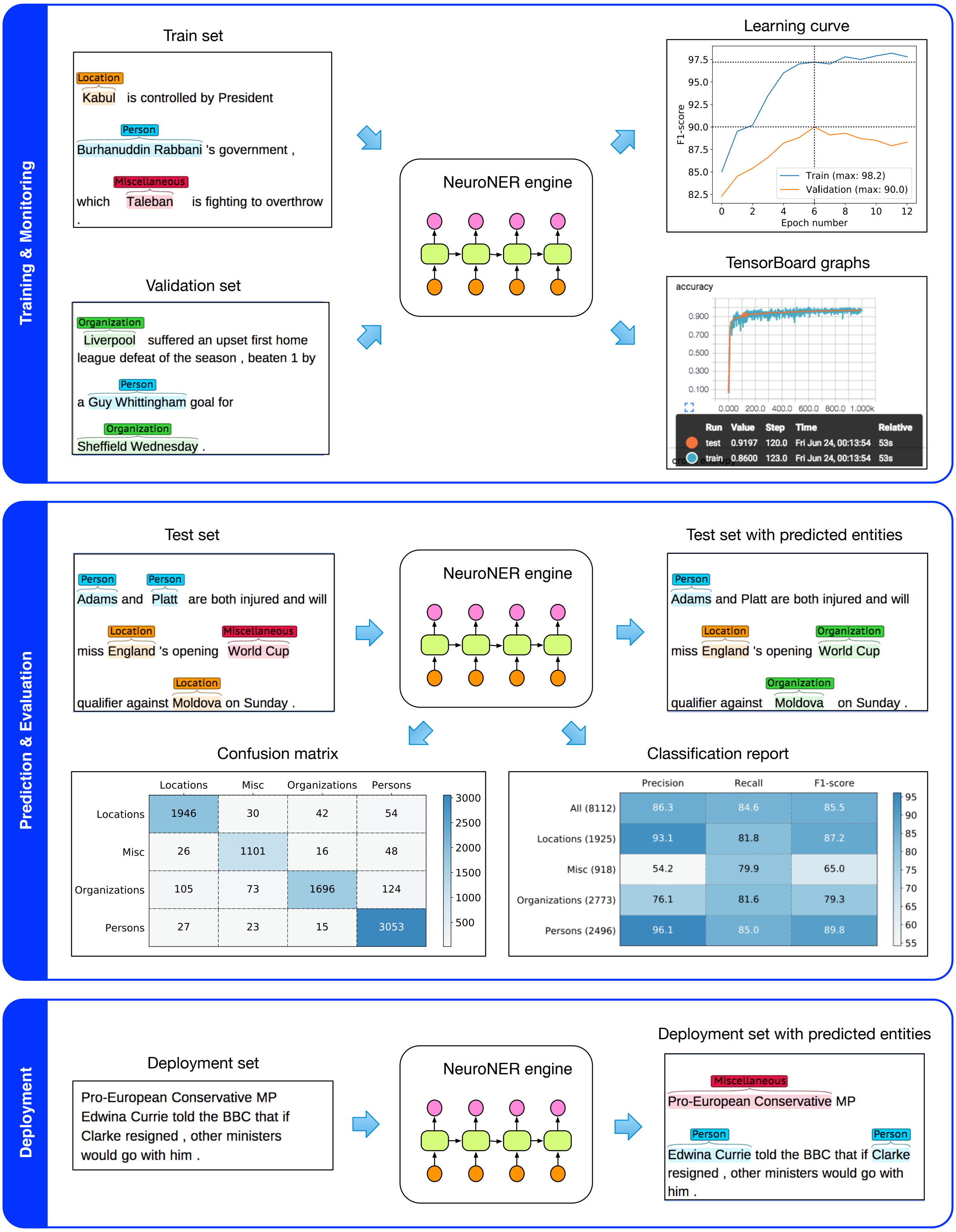
- Character-enhanced token-embedding layer,
- Label prediction layer,
- Label sequence optimization layer.

The character-enhanced token-embedding layer maps each token to a vector representation. The sequence of vector representations corresponding to a sequence of tokens is then input to label prediction layer, which outputs the sequence of vectors containing the probability of each label for each corresponding token. Lastly, the label sequence optimization layer outputs the most likely sequence of predicted labels based on the sequence of probability vectors from the previous layer. All layers are learned jointly. The model architecture is detailed in (Dernoncourt et al., 2016).

The ANN as well as the training process have several hyperparameters such as character embedding dimension, character-based token-embedding LSTM dimension, token embedding dimension, and dropout probability. All hyperparameters may be specified in a configuration file that is human-readable, so that the user does not have to dive into any code. Listing 1 presents an excerpt of the configuration file.

```
[dataset]
dataset_folder                = dat/conll

[character_lstm]
using_character_lstm          = True
char_embedding_dimension      = 25
char_lstm_dimension           = 50

[token_lstm]
token_emb_pretrained_file     = glove.txt
token_embedding_dimension     = 200
token_lstm_dimension          = 300

[crf]
using_crf                     = True
random_initial_transitions    = True

[training]
dropout                       = 0.5
patience                      = 10
maximum_number_of_epochs      = 100
maximum_training_time         = 10
number_of_cpu_threads         = 8
```

Listing 1: Excerpt of the configuration file used to define the ANN as well as the training process. Only the `dataset_folder` parameter needs to be changed by the user: the other parameters have reasonable default values, which the user may optionally tune.



Kabul is controlled by President Burhanuddin Rabbani 's government , which Taleban is fighting to overthrow .

Liverpool suffered an upset first home league defeat of the season , beaten 1 by a Guy Whittingham goal for Sheffield Wednesday .

	Run	Value	Step	Time	Relative
●	test	0.9197	120.0	Fri Jun 24, 00:13:54	53s
●	train	0.8600	123.0	Fri Jun 24, 00:13:54	53s

Adams and Platt are both injured and will miss England 's opening World Cup qualifier against Moldova on Sunday .

Adams and Platt are both injured and will miss England 's opening World Cup qualifier against Moldova on Sunday .

	Locations	Misc	Organizations	Persons
Locations	1946	30	42	54
Misc	26	1101	16	48
Organizations	105	73	1696	124
Persons	27	23	15	3053

	Precision	Recall	F1-score
All (8112)	86.3	84.6	85.5
Locations (1925)	93.1	81.8	87.2
Misc (918)	54.2	79.9	65.0
Organizations (2773)	76.1	81.6	79.3
Persons (2496)	96.1	85.0	89.8

Pro-European Conservative MP Edwina Currie told the BBC that if Clarke resigned , other ministers would go with him .

Pro-European Conservative MP Edwina Currie told the BBC that if Clarke resigned , other ministers would go with him .

Figure 1: NeuroNER system overview. In the NeuroNER engine, the training set is used to train the parameters of the ANN, and the validation set is used to determine when to stop training. The user can monitor the training process in real time via the learning curve and TensorBoard. To evaluate the trained ANN, the labels are predicted for the test set: the performance metrics can be calculated and plotted by comparing the predicted labels with the gold labels. The evaluation can be done at the same time as the training if the test set is provided along with the training and validation sets, or separately after the training or using a pre-trained model. Lastly, the NeuroNER engine can label the deployment set, i.e. any new text without gold labels.

3.2 Real-time monitoring for training

As training an ANN may take many hours, or even a few days on very large datasets, NeuroNER provides the user with real-time feedback during the training for monitoring purpose. Feedback is given through two different means: plots generated by NeuroNER, and TensorBoard.

Plots NeuroNER generates several plots showing the training progress and outcome at each epoch. Plots include the evolution of the overall F1-score over time, confusion matrices visualizing the number of correct versus incorrect predictions for each class, and classification reports showing the F1-score, precision and recall for each class.

TensorBoard As NeuroNER is based on TensorFlow , it leverages the functionalities of TensorBoard. TensorBoard is a suite of web applications for inspecting and understanding TensorFlow runs and graphs. It allows to view in real time the performances achieved by the ANN being trained. Moreover, since it is web-based, these performances can be conveniently shared with anyone remotely. Lastly, since graphs generated by TensorBoard are interactive, the user may gain further insights on the ANN performances.

3.3 Pre-trained models

Some users may prefer not to train any ANN model, either due to time constraints or unavailable gold labels. For example, if the user wants to tag protected health information, they might not be able to have access to a labeled identifiable dataset. To address this need, NeuroNER provides a set of pre-trained models. Users are encouraged to contribute by uploading their own trained models. NeuroNER also comes with several pre-trained token embeddings, either with word2vec (Mikolov et al., 2013a,b) or GloVe (Pennington et al., 2014), which the NeuroNER engine can load easily once specified in the configuration file.

3.4 Annotations

NeuroNER is designed to smoothly integrate with the freely available web-based annotation tool BRAT, so that non-expert users may create or improve annotations. Specifically, NeuroNER addresses two main use cases:

- creating new annotations from scratch, e.g. if the goal is to annotate a dataset for which no gold label is available,
- improving the annotations of an already labeled dataset: the annotations may have been done by another human or by a previous run of NeuroNER.

In the latter case, the user may use NeuroNER interactively, by iterating between manually improving the annotations and running the NeuroNER engine with the new annotations to obtain more accurate annotations.

NeuroNER can take as input datasets in the BRAT format, and outputs BRAT-formatted predictions, which makes it easy to start training directly from the annotations as well as visualize and analyze the predictions. We chose BRAT for two main reasons: it is easy to use, and it can be deployed as a web application, which allows crowdsourcing. As a result, the user may quickly gather a vast amount of annotations by using crowdsourcing marketplaces such as Amazon Mechanical Turk (Buhrmester et al., 2011) and CrowdFlower (Finin et al., 2010).

One limitation of NeuroNER is that it does not allow overlapping annotations in the BRAT format. However, NeuroNER is not restricted to named-entity recognition: it may be used for any sequence labeling, such as part-of-speech tagging and chunking.

3.5 System requirements

NeuroNER runs on Linux, Mac OS X, and Microsoft Windows. It requires Python 3.5, TensorFlow 1.0 (Abadi et al., 2016), scikit-learn (Pedregosa et al., 2011), and BRAT. A setup script is provided to make the installation straightforward. It can use the GPU if available, and the number of CPU threads and GPUs to use can be specified in the configuration file.

3.6 Performances

To assess the quality of NeuroNER's predictions, we use two publicly and freely available datasets for named-entity recognition: CoNLL 2003 and

Model	CoNLL 2003	i2b2 2014
Best published	90.9	97.9
NeuroNER	90.5	97.7

Table 1: F1-scores (%) on the test set comparing NeuroNER with the best published methods in the literature, viz. (Passos et al., 2014) for CoNLL 2003, (Dernoncourt et al., 2016) for i2b2 2014.

i2b2 2014. CoNLL 2003 (Tjong Kim Sang and De Meulder, 2003) is a widely studied dataset with 4 usual types of entity: persons, organizations, locations and miscellaneous names. We use the English version.

The i2b2 2014 dataset (Stubbs et al., 2015) was released as part of the 2014 i2b2/UTHealth shared task Track 1. It is the largest publicly available dataset for de-identification, which is a form of named-entity recognition where the entities are protected health information such as patients' names and patients' phone numbers. 22 systems were submitted for this shared task.

Table 1 compares NeuroNER with state-of-the-art systems on CoNLL 2003 and i2b2 2014. Although the hyperparameters of NeuroNER were not optimized for these datasets (the default hyperparameters were used), the performances of NeuroNER are on par with the state-of-the-art systems.

4 Conclusions

In this article we have presented NeuroNER, an ANN-based NER tool that is accessible to non-expert users and yields state-of-the-art results. Addressing the need of many users who want to create or improve annotations, NeuroNER smoothly integrates with the web-based annotation tool BRAT.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

John Aberdeen, Samuel Bayer, Reyyan Yeniterzi, Ben Wellner, Cheryl Clark, David Hanauer, Bradley Malin, and Lynette Hirschman. 2010. The mitre identification scrubber toolkit: design, training, and assessment. *International journal of medical informatics*, 79(12):849–859.

Masayuki Asahara and Yuji Matsumoto. 2003. Japanese named entity extraction with redundant morphological analysis. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 8–15. Association for Computational Linguistics.

Yassine Benajiba and Paolo Rosso. 2008. Arabic named entity recognition using conditional random fields. In *Proc. of Workshop on HLT & NLP within the Arabic World, LREC*, volume 8, pages 143–153. Citeseer.

Daniel M Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. 1997. Nymble: a high-performance learning name-finder. In *Proceedings of the fifth conference on Applied natural language processing*, pages 194–201. Association for Computational Linguistics.

William Boag, Kevin Wacome, Tristan Naumann, and Anna Rumshisky. 2015. Cliner: A lightweight tool for clinical named entity recognition. *AMIA Joint Summits on Clinical Research Informatics (poster)*.

Andrew Borthwick, John Sterling, Eugene Agichtein, and Ralph Grishman. 1998. Nyu: Description of the mene named entity system as used in muc-7. In *In Proceedings of the Seventh Message Understanding Conference (MUC-7)*. Citeseer.

Michael Buhrmester, Tracy Kwang, and Samuel D Gosling. 2011. Amazon's mechanical turk a new source of inexpensive, yet high-quality, data? *Perspectives on psychological science*, 6(1):3–5.

Jeffrey T Chang, Hinrich Schütze, and Russ B Altman. 2004. Gapscore: finding gene and protein names one word at a time. *Bioinformatics*, 20(2):216–225.

HC Cho, N Okazaki, M Miwa, and J Tsujii. 2010. Nersuite: a named entity recognition toolkit. *Tsujii Laboratory, Department of Information Science, University of Tokyo, Tokyo, Japan*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Hamish Cunningham, Yorick Wilks, and Robert J Gaizauskas. 1996. Gate: a general architecture for text engineering. In *Proceedings of the 16th conference on Computational linguistics-Volume 2*, pages 1057–1060. Association for Computational Linguistics.

Franck Dernoncourt, Ji Young Lee, Ozlem Uzuner, and Peter Szolovits. 2016. De-identification of patient notes with recurrent neural networks. *Journal of the American Medical Informatics Association (JAMIA)*.

Michele Filannino, Gavin Brown, and Goran Nenadic. 2013. Mantime: Temporal expression identification and normalization in the tempeval-3 challenge. *arXiv preprint arXiv:1304.7942*.

Tim Finin, Will Murnane, Anand Karandikar, Nicholas Keller, Justin Martineau, and Mark Dredze. 2010. Annotating named entities in twitter data with crowdsourcing. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 80–88. Association for Computational Linguistics.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 363–370. Association for Computational Linguistics.

N Kumar and Pushpak Bhattacharyya. 2006. Named entity recognition in Hindi using MEMM. *Techbical Report, IIT Mumbai*.

Matthieu Labeau, Kevin Löser, and Alexandre Allauzen. 2015. Non-lexical neural architecture for fine-grained POS tagging. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 232–237, Lisbon, Portugal. Association for Computational Linguistics.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Robert Leaman, Graciela Gonzalez, et al. 2008. Banner: an executable survey of advances in biomedical named entity recognition. In *Pacific symposium on biocomputing*, volume 13, pages 652–663.

Ji Young Lee, Franck Dernoncourt, Ozlem Uzuner, and Peter Szolovits. 2016. Feature-augmented neural networks for patient note de-identification. *COLING Clinical NLP*.

Diana Maynard and Hamish Cunningham. 2003. Multilingual adaptations of annie, a reusable information extraction tool. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 2*, pages 219–222. Association for Computational Linguistics.

Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 188–191. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Alexandre Passos, Vineet Kumar, and Andrew McCallum. 2014. Lexicon infused phrase embeddings for named entity resolution. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 78–86, Ann Arbor, Michigan. Association for Computational Linguistics.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, 12:1532–1543.

Guergana K Savova, James J Masanz, Philip V Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C Kipper-Schuler, and Christopher G Chute. 2010. Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513.

Satoshi Sekine et al. 1998. Nyu: Description of the japanese ne system used for met-2. In *Proc. Message Understanding Conference*.

Burr Settles. 2005. ABNER: An open source tool for automatically tagging genes, proteins, and other entity names in text. *Bioinformatics*, 21(14):3191–3192.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. Brat: a web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107. Association for Computational Linguistics.

Amber Stubbs, Christopher Kotfila, and Özlem Uzuner. 2015. Automated systems for the de-identification of longitudinal clinical narratives: Overview of 2014 i2b2/uthealth shared task track 1. *Journal of biomedical informatics*, 58:S11–S19.

Erik F Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 142–147. Association for Computational Linguistics.

Richard Tzong-Han Tsai, Cheng-Lung Sung, Hong-Jie Dai, Hsieh-Chuan Hung, Ting-Yi Sung, and Wen-Lian Hsu. 2006. Nerbio: using selected word conjunctions, term normalization, and global patterns to improve biomedical named entity recognition. *BMC bioinformatics*, 7(5):S11.