Named Entity Recognition in Swedish Health Records with Character-Based Deep Bidirectional LSTMs

Simon Almgren[1], **Sean Pavlov**[1], **Olof Mogren***
Chalmers University of Technology, Sweden
*olof@mogren.one

Abstract

We propose an approach for named entity recognition in medical data, using a character-based deep bidirectional recurrent neural network. Such models can learn features and patterns based on the character sequence, and are not limited to a fixed vocabulary. This makes them very well suited for the NER task in the medical domain. Our experimental evaluation shows promising results, with a 60% improvement in F_1 score over the baseline, and our system generalizes well between different datasets.

1 Introduction

Named Entity Recognition (NER) is the task of finding mentions of named entities in a text. In non-medical NER, entity classes are typically people, organizations, and locations. It is one of the fundamental Natural Language Processing (NLP) tasks and has been studied extensively.

In this paper, we approach the problem of finding medical entities such as (1) *disorders and findings*, (2) *pharmaceutical drugs*, and (3) *body structure*. Our proposed method uses deep bidirectional character-based recurrent neural networks (RNNs), trained in an end-to-end fashion to perform both boundary detection and classification at the same time.

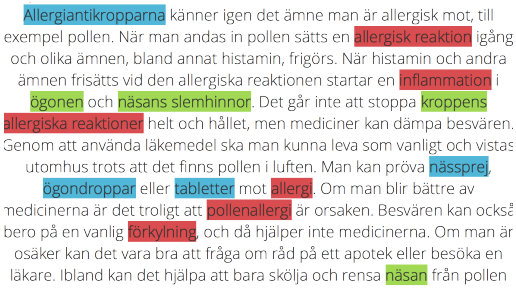

Figure 1: A Swedish medical example text with NER tags illustrated with colour.

There are a number of properties that make this problem especially challenging in biomedical text (Zhou et al., 2004). Firstly, names composed of multiple words are frequently used to describe an entity, highlighting the requirement of good boundary detection on an NER system. Secondly, one noun can be part of a mention of several entities at the same time E.g: "91 and 84 kDa proteins" consists of two entity names: "91 kDa proteins" and "84 kDa proteins". Thirdly, it is common to write the same biomedical entity in different ways, e.g: "N-acetylcysteine", "N-acetyl-cysteine", "NAcetylCysteine". Lastly, ambiguous mentions are common, including abbreviations that refer to different things in different contexts. (The examples above are from Zhou et al. (2004)).

Our proposed method has a number of benefits over previous work: Firstly, the model can simultaneously recognize and classify entity mentions. Secondly, using an end-to-end neural network approach eliminates the need for feature engineering. All features needed are learned by the model during training. Thirdly, because our model works on the raw character sequence, it does not suffer from out-of-vocabulary terms, it can learn that different character patterns represent the same thing, and it can learn the typical character-based features often used in traditional machine learning based solutions to NER.

We evaluate the model on Swedish health records in the Stockholm EPR corpus and obtain promising results. We also note that the method generalizes well between different datasets.

1. Equal contribution.

Proceedings of the Fifth Workshop on Building and Evaluating Resources for Biomedical Text Mining (BioTxtM 2016), pages 30–39, Osaka, Japan, December 12th 2016.

2 Background

A recurrent neural network (RNN) is a feedforward artificial neural network that can model a sequence of arbitrary length, using weight sharing between each position in the sequence. In a language setting, it is common to model sequences of words, in which case each input x_t is the vector representation of a word. In the basic RNN variant, the transition function is a linear transformation of the hidden state and the input, followed by a pointwise nonlinearity:

$$h_t = \tanh(Wx_t + Uh_{t-1} + b),$$

where W and U are trainable weight matrices, b is a bias term, and tanh is the nonlinearity.

Basic RNNs struggle with learning long dependencies and suffer from the vanishing gradient problem. This makes RNN models difficult to train (Hochreiter, 1998; Bengio et al., 1994), and provoked the development of the Long Short Term Memory (LSTM) (Schmidhuber and Hochreiter, 1997), that to some extent solves these shortcomings. An LSTM is an RNN where the cell at each step t contains an internal memory vector c_t, and three gates controlling what parts of the internal memory will be kept (the forget gate f_t), what parts of the input that will be stored in the internal memory (the input gate i_t), as well as what will be included in the output (the output gate o_t). In essence, this means that the following expressions are evaluated at each step in the sequence, to compute the new internal memory c_t and the cell output h_t. Here "$\odot$" represents element-wise multiplication.

$$\begin{aligned}
i_t &= \sigma(W^{(i)}x_t + U^{(i)}h_{t-1} + b^{(i)}),\\
f_t &= \sigma(W^{(f)}x_t + U^{(f)}h_{t-1} + b^{(f)}),\\
o_t &= \sigma(W^{(o)}x_t + U^{(o)}h_{t-1} + b^{(o)}),\\
u_t &= \tanh(W^{(u)}x_t + U^{(u)}h_{t-1} + b^{(u)}),\\
c_t &= i_t \odot u_t + f_t \odot c_{t-1},\\
h_t &= o_t \odot \tanh(c_t).
\end{aligned} \tag{1}$$

Most RNN based models work on word level. Words are coded as a one-hot vector, and each word is associated with an internally learned embedding vector. In this work, we propose a character-level model that is able to learn features based on arbitrary parts of the character sequence.

LSTM networks have been used successfully for language modelling, sentiment analysis (Tang et al., 2015), textual entailment (Rocktäschel et al., 2016), and machine translation (Sutskever et al., 2014). In the following sections, we will see that the learned features are also suitable for recognizing and classifying mentions of medical entities in health record data.

3 Named Entity Recognition with Character-Based Deep Bidirectional LSTMs

In this paper, we propose a character based RNN model with deep bidirectional LSTM cells (BiLSTM) to do Named Entity Recognition in the medical domain (see Figure 2). The model is trained and evaluated on medical texts in Swedish. It has a softmax output layer with four outputs corresponding to each position in the input sequence, representing the three different entity labels, and a special label for all non-entity characters.

The model is trained end-to-end using backpropagation and the Adam optimizer (Diederik Kingma, 2015) to perform entity classification on a character-by-character basis. A neural network learns to internally represent data with representations that are useful for the task. This is an effect of using backpropagation, and allows us to eliminate all manual feature engineering, enabling quick deployment of our system.

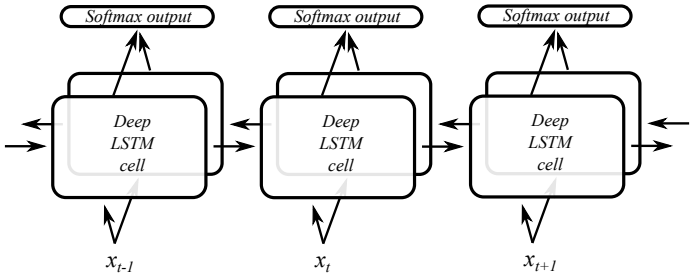

Figure 2: A deep bidirectional LSTM network. At each input x_t, the model is trained to output a prediction y_t of the correct entity class. In this paper, each block is a deep LSTM cell (see Figure 3), and the network is trained using backpropagation through time (BPTT).

3.1 Character classification

Our model works on the raw character sequence of the input document. This is an approach that has proven to work well in some other NLP applications, (see e.g. Luong and Manning (2016), **?**)).

Compared to a word-based sequence model, this means that we can use a much smaller vocabulary for the input tokens. Traditional (non-neural) entity recognition systems typically rely heavily on hand-engineered character-based features, such as capitalization, numerical characters, word prefixes and suffixes (Ratinov and Roth, 2009). Having the capacity of learning this kind of features automatically is what motivated us to use this kind of model. A character-based model does not rely on words being in its vocabulary: any word can be represented, as long as it is written with the given alphabet.

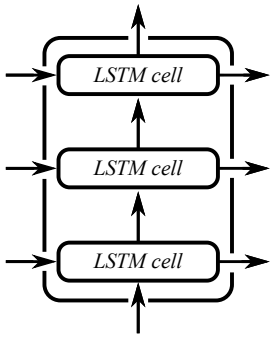

Figure 3: A deep LSTM cell, consisting of 3 internally stacked LSTM cells.

The character sequence model computes one classification output per input character. The label is one of: (1) *disorders and findings*, (2) *pharmaceutical drugs*, (3) *body structure*, (4) *non-entity term*. Using these labels (including the special "non-entity" label), we can simultaneously recognize and classify entity mentions by computing one label per character in the input text. This means that we can interpret each connected subsequence with the same classification as an entity mention.

However, there are some special cases: Firstly, to handle the situation when sporadic characters are classified inconsistently, we treat the character classifications as a voting mechanism for each word, and the majority class is chosen. Secondly, if a space between two tokens is classified consistently with the two tokens, both tokens are interpreted as belonging to the same entity mention. If the space is classified as a non-entity character, the two tokens are treated as two different entity mentions.

4 Experimental setup

This section explains the set-up of the empirical study of our model.

4.1 Model layout

We used a deep bidirectional recurrent neural network with LSTM cells. The depth of the LSTM cells was set to 3, and we used 128 hidden units in the LSTM cells. The model was implemented using Tensorflow. Learning rate: 0.002, decay rate: 0.975. Using drop-out on activations from the input embedding layers as well as on the LSTM output activations were evaluated, but was left out in the final version. See Section 4.4 for details on hyperparameters. The source code of our model is available on Github[1].

[1] `https://github.com/withtwist/medical-ner/`

4.2 Seed-term collection

Seed-terms are used both to build the datasets (see Section 4.3), and to build up the representations for the classification centroids in the BOW baseline method. Seed-terms were extracted from two taxonomies, SweMeSH[2], a taxonomy of Swedish medical terms and Snomed CT[3], consisting of Swedish medical concept terms. Using the hierarchical structure of the two taxonomies, all terms that was descendants of each of our predefined categories was extracted and considered seed-terms. The following predefined categories was used for the extraction: *disorder & finding* (*sjukdom & symtom*), *pharmaceutical drug* (*läkemedel*) and *body structure* (*kroppsdel*). The choice of these main entity classes was aligned with Skeppstedt et al. (2014).

4.3 Datasets

We use an approach similar to Mintz et al. (2009) to obtain the data needed for training and evaluation. The datasets that we prepared for training, validating and testing our model are available for download at `https://github.com/olofmogren/biomedical-ner-data-swedish/`.

The *Läkartidningen* corpus was originally presented by Kokkinakis and Gerdin (2010), and contains articles from the Swedish journal for medical professionals. This was annotated for NER as a part of this work. All occurrences of seed-terms were extracted (see Section 4.2), along with a context window of 60 characters (approximately ten words). The window is positioned so that the entity mention is located randomly within the sequence. In addition, negative training examples were extracted in order to prevent the model from learning that classified entities always occur in every sequence. All the characters in these negative training examples had the same "non-entity" label. Neural models typically benefit from large amounts of training data. To increase the amount of training data, each occurrence of seed-terms were extracted three more times, where the window was shifted by a random number of steps. The resulting data is a total of 775,000 of sequences with 60 characters each. 10% of the data is negative data, where every character has the "non-entity" label.

Another dataset was built from medical articles on the *Swedish Wikipedia*. Firstly, an initial list of medical domain articles were chosen manually and fetched. Secondly, articles were fetched that were linked from the initial articles. Finally, the seed-terms list (see Section 4.2) was used to create the labels and extract training examples of 60 character sequences, in the same way as was done with *Läkartidningen*.

1177 Vårdguiden iis a web site provided by the Swedish public health care authorities, containing information, counselling, and other health-care services. The corpus consists of 15 annotated documents downloaded during May 2016. This dataset was manually annotated with the seed-terms list as support (see Section 4.2). The resulting dataset has 2740 annotations, out of which 1574 are *disorder and finding*, 546 are *pharmaceutical drug*, and 620 are *body structure*.

The *Stockholm Electronic Patient Record (EPR) Clinical Entity Corpus* (Dalianis et al., 2012) is a dataset with health records of over two million patients at Karolinska University Hospital in Stockholm encompassing the years 2006-2014. It consists of 7946 documents containing real-world anonymized health records with annotations in 4 categories: *disorder*, *finding*, *drug* and *body structure*. Since we have a category where "*disorder*" and "*finding*" are bundled together they were considered the same.

Läkartidningen, *Swedish Wikipedia*, and *1177 Vårdguiden* are all datasets with rather high quality text, most of it even professionally edited. This is in stark contrast to the text in *Stockholm EPR* where misspellings are common, there are reduntant parts in many records, and writing style is highly diverse (Dalianis et al., 2009).

[2] `http://mesh.kib.ki.se/swemesh/swemesh_se.cfm`
[3] `http://www.socialstyrelsen.se/nationellehalsa/snomed-ct`

4.4 Hyperparameter search

A number of settings for hyperparameters were explored during development. In the variations listed below, one hyperparameter at the time is varied and evaluated, and if we saw an improvement, the change of setting was retained. A more thorough hyperparameter investigatin is left for future work. For the three first experiments, dropout was used on the activations from the embedding layer, as well as on the activations on the LSTM outputs. (See Section 4.1 for details).

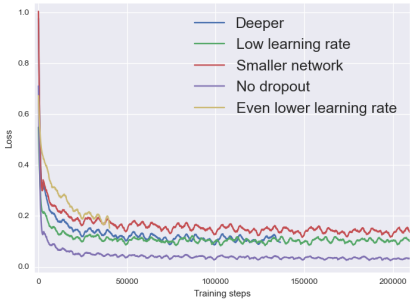

Figure 4: Validation loss.

Deeper: A model using 4 stacked LSTM cells. Learning rate: 0.05, decay rate: 0.975, drop probability: 0.5. *Low learning rate*: LSTK depth: 3, learning rate: 0.002, decay rate: 0.975. Lowering the learning rate proved useful and 0.002 became the default setting for learning rate. Drop probability: 0.5. *Smaller network*: 64 hidden units in each LSTM cell. LSTM depth: 3: learning rate: 0.002, decay rate: 0.975, drop probability: 0.5. *No dropout*: This model left all the settings as default but removed dropout entirely. 128 hidden units in each LSTM cell, LSTM depth: 3, learning rate: 0.002 and decay rate: 0.975. This setting proved to be the best, which meant that the default settings subsequently never used dropout. *Even lower learning rate*: Learning rate: 0.0002 and decay rate: 0.975. No drop-out.

See Figure 4 for the validation performance of the different settings. The resulting model used in the final experiments reported in Section 5 had 3 stacked LSTM layers with 128 hidden units in each. Learning rate: 0.002, decay rate 0.975, and no drop-out.

4.5 Baselines

Two baselines were implemented and used. The **dictionary baseline** simply consist of dictionary look-ups of the encountered words in the list of seed-terms. The **BOW (Bag-Of-Words) baseline** is based on Zhang and Elhadad (2013). The original version was developed for and evaluated on medical texts in English. The approach considers each noun-phrase as an entity candidate, and represents each candidate using a weighted bag-of-words-vector for the context. The required preprocessing is tokenization, sentence splitting, part-of-speech-tagging, and noun phrase chunking. The first steps was done using GATE (Cunningham et al., 2011) and OpenNLP[4], while noun phrase chunking was done using Swe-SPARK (Aycock, 1998). An IDF threshold is used to filter out uncommon or unspecific noun phrases. Then for each category the algorithm builds an average context vector representing the mentions in a training corpus. We used a triangular window for the context vectors, giving the central words a weight of 20, and context words the weight of $1/k$, where k is the distance from the central word. Mentions with a cosine similarity lower than 0.005 to any of the category vectors was discarded. Candidate mentions that have a difference between the top two scoring categories that is lower than 0.7 are also discarded.

Zhang and Elhadad (2013) used one bag-of-words vector for the internal words of an entity mention, and one for the context words of the mention. The two vectors were then concatenated, resulting in a vector which is twice the size of their vocabulary. Since the bag-of-words-vectors are already sparse to begin with, we added them together instead and made it possible to use a larger vocabulary size.

4.6 Training

Development and training were performed using text from *Läkartidningen* (Kokkinakis and Gerdin, 2010). Validation was done using the *Medical Wikipedia* dataset. Training was done using the Adam optimizer (Diederik Kingma, 2015).

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

4.7 Evaluation

Evaluation of the proposed model was performed on two different datasets: *Stockholm EPR corpus* (Dalianis et al., 2012), with anonymized health record data in Swedish, and *1177 Vårdguiden*.

We report F_1 scores for total named entity recognition, as well as only entity classification (given correct boundary detection, we report scores of the entity classification performed by the system).

In the BOW baseline the entities are determined before hand while the Char-BiLSTM model recognizes and classifies them as it traverses the document.

5 Results

This section presents the results of the experimental evaluation of the proposed model. Table 1 shows the results of running the dictionary baseline model on *Stockholm EPR corpus* (Dalianis et al., 2012). The baseline model achieves a precision of over 0.70 on *disorder & Finding* and *body structure*, but is substantially lower for *pharmaceutical drug*. It has a higher precision than recall in general due to the fact that if a match is found it is probably correct. The algorithm got a precision of 0.67, a recall of 0.12 and an F_1 score of 0.20.

Category	**P**	**R**	**F_1**
Disorder & finding	0.76	0.12	0.20
Pharmaceutical drug	0.25	0.04	0.07
Body structure	0.70	0.29	0.41
Total	0.67	0.12	0.20

Table 1: Dictionary baseline performance on the *Stockholm EPR corpus*. Although total precision is reasonably good (0.67), the precision (0.12) is not.

The evaluation of the Char-BiLSTM model was performed on 733 real-world examples of health records from *Stockholm EPR corpus* (Dalianis et al., 2012). Since the data is very sensitive the evaluation was not performed by ourselves but instead the holder of the data performed the evaluations.

Char-BiLSTM overall results: The results in Table 2 shows the results of the Char-BiLSTM model. Both *disorder & finding* and *body structure* have a much higher precision than recall, while *pharmaceutical drug* is better balanced.

Category	**P**	**R**	**F_1**
Disorder & findings	0.72	0.18	0.29
Pharmaceutical drugs	0.69	0.43	0.53
Body structure	0.46	0.28	0.35
Total	0.67	0.24	0.35

Table 2: Results for Char-BiLSTM on *Stockholm EPR corpus*. The model obtains a total precision that matches the dictionary baseline (0.67), and a recall that is much higher than the baseline (0.24).

Char-BiLSTM results, classification only: Given the boundaries for the entities in *Stockholm EPR corpus*, the performance of the Char-BiLSTM model (performing only classification of the given entities) is given in Table 3. The table shows promising results for both the category *disorder & finding* and *pharmaceutical drug* which has an F_1 score of 0.81 and 0.74 respectively. *body structure* shows a weaker F_1 score of 0.47. The model got an overall F_1 score of 0.75.

We compare our system with the two baselines using the *1177 Vårdguiden corpus*. Since the BOW baseline detects boundaries using whole noun phrases, we re-ran the experiments, adjusting the evaluation data, so that the boundaries were complete noun-phrases.

Category	P	R	F_1
Disorder & findings	0.92	0.73	0.81
Pharmaceutical drugs	0.64	0.87	0.74
Body structure	0.36	0.68	0.47
Total	0.75	0.75	0.75

Table 3: Entity classification results of Char-BiLSTM on *Stockholm EPR corpus*. Given the entity boundaries, we can see that the classification of entities work very well, obtaining a total F_1 score of 0.75.

Comparing the results of the models in Table 4, we see that the BOW baseline does not perform well due to the wider boundaries that it detects. This can clearly be seen in the experiments with adjusted data, where it performs around 47% better. All models have around 0.50 in precision, except for the adjusted BOW baseline which comes in at 0.32. Recall is much lower (between 0.08 and 0.17), except for the BiLSTM model which has a recall of 0.21. This is the main reason why the BiLSTM model has the highest F_1 score with 0.29. At the second place comes the adjusted BOW baseline at 0.22, followed by the dictionary baseline model at 0.22 and lastly the BOW baseline with 0.15. Even though the dictionary baseline model and the adjusted BOW baseline have similar performance scores, we can see that their precision and recall are vastly different. The dictionary baseline model has a high precision and a low recall, while the adjusted BOW baseline is more balanced between precision and recall.

Model	P	R	F_1
Dictionary	0.54	0.14	0.22
BOW	**0.55**	0.09	0.15
BOW (adj.)	0.32	0.17	0.22
Char-BiLSTM	0.48	**0.21**	**0.29**

Table 4: Comparison of the results between each model on *1177 Vårdguiden*.

6 Related work

Supervised NER has been thoroughly explored in the past. Finkel et al. (2005) used Conditional Random Fields (CRF), a technique often used for NER. Zhou et al. (2004) used Hidden Markov Models (HMMs) along with extensive feature engineering to perform NER on medical texts. State-of-the-art in the medical domain have been achieved by Wang and Patrick (2009) with a combination of CRF, Support Vector Machines (SVM) and Maximum Entropy (ME) to recognize and classify entities.

Skeppstedt et al. (2014) currently holds the state-of-the-art in Swedish for the medical domain, based on CRF.

Recently, a number of papers have proposed using RNNs for sequence labelling tasks. Cícero Nogueira dos Santos (2015) presented a model that learns word embeddings along with character embeddings from a convolutional layer, which are used in a window-based fixed feed forward neural network. Huang et al. (2015) proposed a bidirectional LSTM model, but it used traditional feature engineering, and the classification was performed using a CRF layer in the network. In contrast, our proposed model learns all its features, and can be trained efficiently with simple backpropagation and stochastic gradient descent. Ma and Hovy (2016) presented a model that uses a convolutional network to compute representations for parts of words. Then the representations are combined with some character-level features and fed into a bidirectional LSTM network, and finally a CRF performs the labelling. Chiu and Nichols (2016) presented a similar model but with a softmax output instead of the CRF layer. Like our system, the models are trained end-to-end and obtains good results on standard NER evaluations, however our system is conceptually simpler, and learns all of its features directly from the character stream. Lam-

ple et al. (2016) presented two different architechtures, one using LSTMs and CRFs, and one using a shift-reduce approach. Gillick et al. (2016) presented a character-based model with LSTM units similar to a translation model, but instead of decoding into a different language, the state from the encoder is decoded into a sequence of tags.

Learning representations for text is important for many other tasks within natural language processing. A common way of representing sequences of words is to use some form of word embeddings (Mikolov et al., 2013; Pennington et al., 2014; Collobert and Weston, 2008), and for each word in the sequence, do an element-wise addition (Mitchell and Lapata, 2010). This approach works well for many applications, such as phrase similarity, multi-document summarization (Mogren et al., 2015), and word sense induction (Kågebäck et al., 2015), even though it disregards the order of the words. In contrast, RNNs and LSTMs (Hochreiter and Schmidhuber, 1997) learn representations for text that takes the order into account. They have been successfully been applied to sentiment analysis (Tang et al., 2015), question answering systems (Hagstedt P Suorra and Mogren, 2016), and machine translation (Sutskever et al., 2014).

Character-based neural sequence models have recently been presented to tackle the problem of out-of-vocabulary terms in neural machine translation (Luong and Manning, 2016; Chung et al., 2016) and for language modelling (Kim et al., 2016).

7 Discussion

The results of the empirical evaluation of the proposed system show some interesting points, suggesting that this approach should be researched further.

We have evaluated our model on the Stockholm EPR corpus of Swedish health records, but we did not compare our scores with other approaches that was evaluated on the same dataset. The reason is that we were unable to do a fair comparison since our model was trained on other data. We believe that our scores are competitive, and indicates that the model is promising. While systems that were trained on data from the same corpus show better performance in the evaluation on *Stockholm EPR* (Skeppstedt et al., 2014), we note that our solution can be trained on a dataset that is quite diferent from the test set. This can be explained in part with the documented robustness of character-based recurrent neural models to misspellings and out-of-vocabulary terms.

We are convinced that our solution would be able to obtain even better scores if able to train on the same data.

8 Conclusions

In this paper, we have proposed a character-based deep bidirectional LSTM model (Char-BiLSTM) to perform named entity recognition in Swedish health records. We beat two different baseline methods on the task, and show that this is a promising research direction. The proposed model obtains an F_1 score of 0.35 which is about 60% better than the BOW baseline (Zhang and Elhadad, 2013). Our model learns all the features it needs, and therefore eliminates the need for feature engineering. We have seen that a character-based neural model adapts well to this domain, and in fact that it is able to generalize from relatively well-written training data to test-data with lesser quality text.

Acknowledgments

This work has been done within the project "Data-driven secure business intelligence", grant IIS11-0089 from the Swedish Foundation for Strategic Research (SSF).

References

John Aycock. 1998. Compiling little languages in python. In *Proceedings of the 7th International Python Conference*, pages 69–77.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions on*, 5(2):157–166.

Jason Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. 2016. A character-level decoder without explicit segmentation for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1693–1703, Berlin, Germany, August. Association for Computational Linguistics.

Victor Guimarães Cícero Nogueira dos Santos. 2015. Boosting named entity recognition with neural character embeddings. *Proceedings of NEWS 2015 The Fifth Named Entities Workshop*.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, Valentin Tablan, Niraj Aswani, Ian Roberts, Genevieve Gorrell, Adam Funk, Angus Roberts, Danica Damljanovic, Thomas Heitz, Mark A. Greenwood, Horacio Saggion, Johann Petrak, Yaoyong Li, and Wim Peters. 2011. *Text Processing with GATE (Version 6)*.

Hercules Dalianis, Martin Hassel, and Sumithra Velupillai. 2009. The stockholm epr corpus—characteristics and some initial findings. In *Proceedings of the 14th International Symposium on Health Information Management Research - iSHIMR*.

Hercules Dalianis, Martin Hassel, Aron Henriksson, and Maria Skeppstedt. 2012. Stockholm epr corpus: A clinical database used to improve health care. *Swedish Language Technology Conference*, pages 17–18.

Jimmy Ba Diederik Kingma. 2015. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

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.

Dan Gillick, Cliff Brunk, Oriol Vinyals, and Amarnag Subramanya. 2016. Multilingual language processing from bytes. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1296–1306, San Diego, California, June. Association for Computational Linguistics.

Jacob Hagstedt P Suorra and Olof Mogren. 2016. Assisting discussion forum users using deep recurrent neural networks. *Proceedings of the 1st Workshop on Representation Learning for NLP at ACL 2016*, page 53.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Sepp Hochreiter. 1998. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Mikael Kågebäck, Fredrik Johansson, Richard Johansson, and Devdatt Dubhashi. 2015. Neural context embeddings for automatic discovery of word senses. In *Proceedings of NAACL-HLT*, pages 25–32.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Dimitrios Kokkinakis and Ulla Gerdin. 2010. Läkartidningens arkiv i en ny skepnad - en resurs för forskare, läkare och allmänhet. *Språkbruk*, pages 22–28.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. 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*, pages 260–270, San Diego, California, June. Association for Computational Linguistics.

Minh-Thang Luong and Christopher D. Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1054–1063, Berlin, Germany, August. Association for Computational Linguistics.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany, August. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *ICLR*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive science*, 34(8):1388–1429.

Olof Mogren, Mikael Kågebäck, and Devdatt Dubhashi. 2015. Extractive summarization by aggregating multiple similarities. In *Recent Advances in Natural Language Processing*, page 451.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–43.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155. Association for Computational Linguistics.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. 2016. Reasoning about entailment with neural attention. In *International Conference on Learning Representations*.

Jürgen Schmidhuber and Sepp Hochreiter. 1997. Long short-term memory. *Neural computation*, 7(8):1735–1780.

Maria Skeppstedt, Maria Kvist, H. Nilsson Gunnar, and Dalianis Hercules. 2014. Automatic recognition of disorders, findings, pharmaceuticals and body structures from clinical text: An annotation and machine learning study. *Journal of Biomedical Informatics*, 49:148–158.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432.

Yefeng Wang and Jon Patrick. 2009. Cascading classifiers for named entity recognition in clinical notes. In *Proceedings of the workshop on biomedical information extraction*, pages 42–49. Association for Computational Linguistics.

Shaodian Zhang and Noémie Elhadad. 2013. Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts. *Journal of biomedical informatics*, 46(6):1088–1098.

Guodong Zhou, Jie Zhang, Jian Su, Dan Shen, and Chewlim Tan. 2004. Recognizing names in biomedical texts: a machine learning approach. *Bioinformatics*, 20(7):1178–1190.