RESEARCH

Putting hands to rest: efficient deep CNN-RNN architecture for chemical named entity recognition with no hand-crafted rules.

Ilia Korvigo[1,2,3*], Maxim Holmatov[4,5], Anton Karazeev[1], Anatolii Zaikovskii[6] and Mikhail Skoblov[1]

Abstract

Chemical named entity recognition (NER) is an active field of research in biomedical natural language processing. To facilitate the development of new and superior chemical NER systems, BioCreative released the CHEMDNER corpus, an extensive dataset of diverse manually annotated chemical entities. Most of the systems trained on the corpus rely on complicated hand-crafted rules for data preprocessing, feature extracting and output post-processing, though modern machine learning algorithms, such as deep neural networks, can automatically design the rules with no manual intervention. Here we explored this approach by experimenting with various deep learning architectures. Our final model, based on a combination of convolutional and recurrent neural networks with attention-like loops and hybrid word- and character-level embeddings, reaches near human-level performance on the testing dataset, while having no manually asserted rules. To make our model easily accessible for standalone use and integration in third-party software, we've developed a Python package with minimalistic functional interface.

Keywords: named entities recognition; chemical; text mining; deep learning; recurrent neural network; convolutional neural network; biocreative; chemdner; conditional random fields; neural attention

Content

Background

Modern data-generation capabilities have clearly surpassed our capacity to manually analyse published data, which is ever-more evident in the era of high-throughput methods. Naturally, this fuels the development of automatic natural language processing (NLP) systems capable of extracting and transforming specific information from a body of literature with human-level precision. Among all the subtasks NLP introduces, named entity recognition (NER) – aiming to identify objects of particular semantic value (e.g. chemical compounds) – is one of the most fundamental for higher level event-focused analyses, making it an area of active research. Traditionally, chemical NER systems have relied on curated dictionaries and hand-crafted rules (e.g. regular expressions for systematic IUPAC names), which are hard to develop and maintain due to diverse morphology and rich vocabulary of biomedical literature. On the other hand, various machine learning (ML) models can automatically infer efficient rules (input transformations) from annotated corpora. Therefore, in ML terms named entity recognition is a supervised labelling problem.

To facilitate the development of new and superior NER systems, BioCreative announced the CHEMDNER machine learning challenge, which ended in 2015 [1]. As part of this task, a team of experts has produced an extensive manually annotated corpus covering various chemical entity types, including systematic and trivial names, abbreviations and identifiers, formulae and phrases. Due to many difficulties inherent to chemical entity detection and normalisation [1], even manual annotation yields the inter-annotator agreement score of 91%, which can be regarded as the golden standard for any automatic system. Twenty six teams have submitted their NER systems for the challenge, best of which have reached the F1 score of 72-88% [2, 3, 4, 5, 6, 7, 8, 9].

The systems were quite diverse in terms of text preprocessing, which is a separate NLP problem in its own right. Obviously, it's possible to represent any text as a raw sequence of character codes (e.g. byte-like sequences or Unicode character positions), yet it is more common to break the characters into word-

*Correspondence: ilia.korvigo@gmail.com
[1]Laboratory of functional analysis of the genome, Moscow Institute of Physics and Technology, Moscow, Russia
Full list of author information is available at the end of the article

like structures known as tokens, which can be further normalised and/or encoded. Although tokenisation allows reduce the number of time-steps in the sequence, thus reducing the complexity of the input space, it s d be fine enough to minimise the number of merged/overlapping entities [5, 9]. While there are many token encoding strategies, they all can be divided into two major groups: morphology aware and unaware. In the latter case, one usually builds a vocabulary of all tokens occurring in a corpus and applies a minimal frequency threshold to remove noisy entries (e.g. misspelled words and typos). Consequently, all tokens in the vocabulary get a unique identifier $t \in \mathbb{N}_+$, while all out-of-vocabulary (OOV) tokens get a spacial shared identifier. The vocabulary itself can be represented as a matrix $\mathbf{T} = (\mathbf{t}_1, \ldots, \mathbf{t}_T)$ of orthogonal unit vectors (aka one-hot encodings) ~~that~~ ~~are~~ both sparse and purely categorical: their pair-wise distances cary no underlying information about semantical similarity. In their chemical NER system, Dai et al. [9] successfully used the skip-gram embedding model to overcome these limitations. The model uses context information and a shallow neural network to embed high-dimensional one-hot encoded vectors in a lower-dimensional vector space, wherein pair-wise distances represent semantical similarity [10, 11]. Despite this strategy's increasing popularity, few CHEMDNER task participants have u it for morphology unaware encoding, relying instead on manually selected features to expand the token identifiers into feature vectors. Although, such encodings are efficient for morphologically rigid corpora (e.g. standard English texts), morphologically rich biomedical literature introduces many infrequent words and word-forms, resulting in high out-of-vocabulary (OOV) rates [12, 13]. Consequently, most CHEMNDER participants have additionally (or exclusively) used morphology aware-encodings, targeting various manually designed character-level features. Machine-learning strategies were far less diverse than the preprocessin ~~nce textual~~ data are sequential, that is a value t_i at time-step i can be conditioned on the values occurring before and after the time-step, it is e natural to use se ce models for NER problems. Although many such models exist, most of the top-scoring ML-based tools submitted for the CHEMDNER task utilised conditional random fields (CRF), which are traditionally used for sequence labelling. CRFs are graphical models related to hidden Markov models (HMMs). They take a sequence of feature vectors as inputs and generate a sequence of labels, which can be further modified during post-processing. The participants used hand-crafted post-processing rules as diverse as the preprocessing procedures.

From this brief overview of the NER systems submitted for the CHEMDNER task, it becomes quite evident that, despite the addition of machine learning models, in many ways these systems remain conceptually close to manually curated sets of rules (feature extractors). It also p explains, why LeadMine [8] (another contender), a purely rule-based system, outperforms most of the submitted ML-based s. At the same time, it is possible to reduce manual interventions to the bare minimum by treating word encoding and feature extraction as subtasks in the global machine learning task, and this is exactly the kind of problems ~~that~~ deep artificial neural networks (ANNs) excel at. e already mentioned ~~that~~ neural networks can automatically learn morphology unaware word representations the same is true about morphology aware encodings. Furthermore, deep convolutional neural networks can automatically optimise feature~~s~~ ex ors during training [14]. Most importantly, the labelling itself can be done u a recurrent neural network with time-distributed fully-connected neural classifier. Recurrent networks are naturally sequential and Turing-complete, ~~they are~~ extremely powerfu sequence-to-sequence (seq2seq) modelling (including labelling) [15]. Since, modern ANNs have reached near human-level precision in many areas, including some NLP problems, we believe they can raise the bar in chemical NER, too. Here we've experimented with different deep learning models and architectures to create an efficient end-to-end neural model for chemical named entity recognition, while avoiding any manually designed features.

Materials and methods

Problem formulation

We consider named entity recognition as a combination of two problems: sequence labelling and segmentation. Given:

- an ordered set of N character sequences $X = (X_1, \ldots, X_N)$, where $X_i = ({}^i c_1, \ldots, {}^i c_n)$ is a character sequence;
- an ordered set of N annotations $Y = (Y_1, \ldots, Y_N)$, where Y_i is a sequence $Y_i = ({}^i y_1, \ldots, {}^i y_n)$ and ${}^i y_j$ is a tuple of two boolean labels $({}^i s_j, {}^i e_j)$ showing whether the corresponding character is the beginning of a chemical entity and/or part of one, respectively;

our task ~~is to~~ is to create a *predictor* $P : X \to \hat{Y}$, where $\hat{Y}$ is a set of inferred annotations similar to Y. We also introduce a *tokeniser* $T : X \to \sim X$, where $\sim X$ is an ordered sequence of character subsequen thus slightly redefining the objective function to target per-token annotations. Provided that the *tokeniser* is

fine enough to avoid tokens with overlapping annotations, ~~our~~ redefined problem is equivalent to the original one.

Datasets

We used the CHEMDNER corpus [1] to train and validate our models. The corpus contains ten thousands abstracts from eleven chemistry-related fields of science with over 84k manually annotated chemical entities (20k of which were unique) attributed to seven groups:

- ABBREVIATION (15.55%)
- FAMILY (14.15%)
- FORMULA (14.26%)
- IDENTIFIER (2.16%)
- MULTIPLE (0.70%)
- SYSTEMATIC (22.69%)
- TRIVIAL (30.36%)

The MULTIPLE class represents phrases containing several entities of other classes separated by non-chemical words. We ignored the IDENTIFIER class, because the corpus contained too few unique entities of this class to train a competent detector for chemical identifiers. The corpus is separated into three parts: training (3.5k abstracts), development (3.5k) and testing (3k). We joined the first two, randomly shuffled them and separated 10% ~~of the resulting dataset~~ for validation during training to monitor overfitting. We used the test dataset to estimate performance upon training completion.

Text preprocessing

Our text-preprocessing routine consisted two steps: sentence segmentation and tokenisation. For the first task we used geniass [16], a highly accurate maxinum-entropy sentence segmentation model trained on a biomedical corpus. Proper tokenisation is highly important in token-level NLP tasks [5]. On the one hand, this process can isolate semantically and morphologically stable character sequences, making it easier for ~~a sequence~~ the model to focus on the data. In fact, most popular tokenisers rely on a hierarchy of handcrafter rules optimised for the standard English texts, though there are some specifically designed for biomedical and chemical literature. On the other, tokenisation may lead to overlapping annotations if the rules fail to separate several entities or non-entity characters. Consequently, instead of relying on fine-tuned rule-based tokenisers, we used a rather simple Perl-style regular expression `\w+|[^\s\w]`, yielding highly fine-grained tokens. It groups all Unicode word characters (i.e. most characters that can be seen in a word in any language, including numbers) and separates all other characters. For example, the tokeniser transforms "2-amino-1-methyl-6-phenylimidazo[4,5-b]pyridine" into tokens: "2", "-", "amino", "-", "1", "-", "methyl", "-", "6", "-", "phenylimidazo", "[", "4", ",", "5", "-", "b", "]", "pyridine". Our models used both word-level (morphology unaware) and character-level token encodings. Since the encodings were integral to the model itself, we cover them in the next subsection.

Model architecture

During development we've experimented with various deep learning designs and topologies, build from three popular architectures: one-dimensional (1D) convolutional neural networks (CNN), recurrent neural networks (RNN) and time-distributed dense (fully-connected) networks (TDD). 1-dimensional convolutional neural networks ~~can be regarded as~~ trainable feature extractors applied along the time-steps. A deep CNN [14] trains to extract time-invariant hierarchies of features at each time-step in a sequence while optimising the objective function (OF). Since texts are sequential, that is a value t_i at time-step i can be conditioned on the previous and the following time-steps, a time-invariant model alone is not sufficient. We used recurrent neural networks – highly powerful trainable state machines theoretically capable of modelling recurrent relationships of arbitrary depth – to process CNN-extracted features. These networks train by back-propagating the error through time, which in deep sequences may lead to vanishing or exploding gradients. Several RNN architectures have been developed to ~~improve deep sequence modelling~~, most notably the long short time memory network (LSTM) and gated recurrent unit network (GRU) [17]. Both architectures use trainable gates controlling the data flow and memory updates. The GRU architectures is a newer and lighter alternative to the widely adopted LSTM, using two trainable gates instead of the latter's three resulting in less parameters to optimise. Comparative studies have shown, that GRU and LSTM networks perform equally well, while the former tend to converge faster [18]. Fewer parameters are also desirable when training data are scarce. To further improve performance, it is common to use bidirectional RNNs (biRNNs) that "read" ~~the~~ sequence in both directions. We used a time-distributed fully connected network with sigmoid activation to generate label probabilities. In TDD the same "small" multi-layer perceptron (MLP) is applied to each time-step in the transformed input sequence.

All the models we trained had two input nodes for token identifiers and token strings. The token strings were encoded as raw unicode character sequences with no preprocessing. Although recurrent neural networks naturally handle inputs with heterogenous lengths, for computational efficiency and to incorporate CNNs we

joined all encoded sentences into a single array, where each row represented a single sentence. We used zero-padding to fill-in shorter sequences on the right; similarly, we joined token character strings. We added the third input node for padding masks in models with convolutional layers to force the model to ignore the zero-padded "tails". To initialise weights in word-level embedding layers, we downloaded one million random PubMed abstracts from various fields of biomedical and chemical sciences and trained Glove word-vectors [19] of different dimensionality. In our models we experimented with freezing and relaxing (i.e. letting the model adjust) the weights. For character-level encodings we used a character embedding layer, adjusting representations for each character, followed by a shallow biRNN encoding a token's matrix of character-embeddings while conditioning on the entire sentence [20]. We then experimented with different CNN and RNN topologies. ~~We considered~~ several ways to represent the target objective, including the popular IOB (inside, outside, beginning) tagging scheme. Most notably, we experimented with multiple-output networks. In a multiple-output network several output nodes share a segment of the ~~model~~ (or the entire network~~)~~ while solving there own objectives. For example, one output node can estimate the probability that a time-step contributes to a chemical entity, while another node estimates the probability that the same time-step is ~~the first time-step in~~ an entity. Furthermore the second node can use the output of the first one to attend to specific subsequences in the sequence, creating a reinforcing attention loop. Moreover, since it is possible to implement a chain CRF within a neural network [21], some of our models contained a CRF layer as the terminal output node. Throughout the network we used input dropout and recurrent dropout to control overfitting [22]. We implemented the networks in Python 3.5 and Cython using the deep-learning frameworks Keras 2 [23] and TensorFlow 1.3 [24].

Training and validation

As mentioned in the previous subsection there were many hyper-parameters to test and optimise, including frozen or relaxed word-level embeddings, the number of convolutional layers and filters in each CNN computational graph, CNN-filter widths, RNN type and context-size, between-layer dropout levels and recurrent layers, network topology and many more. We~~'ve~~ selected these hyper-parameters by simultaneously training and validating up to four models on an Ubuntu Linux machine with two Intel Xeon E5 CPUs (10 cores and 20 threads each), 512GB of RAM, three Nvidia Titan X (Maxwell) GPUs and one Nvidia GTX 1080 GPU. During training, we used the Adam update function [25] and minimised the binary cross-entropy loss on minibatches of 32 sentences. We applied gradient clipping to make the convergence more stable. Once the validation loss stoped improving, we increased the batch size to 200 samples

Results and discussion

Tokenisation, overlapping annotations and sequence lengths

In the subsection on problem formulation, we reformulated the initial objective by considering per-token instead of per-character annotations. These problems are equivalent as long as the tokenisation is fine enough to avoid overlapping annotations. To test whether this property is true for our tokeniser, we tokenised the entire CHEMINDER corpus and searched for entities with overlapping annotations. In total there were over 2.5 million tokens with around 180k tokens inside chemical entities of less than 400 had conflicting annotations, leading to an overlapping annotation rate of 0.2%. Almost all of these tokens were either amino acids with appended numeric positions in a protein (e.g. "Ser845"), wherein the numeric part was not annotated as a chemical entity, or typing errors, e.g. skipped white-spaces in "dietarydaidzein". At the same time, some of the overlapping entities were in fact annotation errors. For example in "aminoketones" the annotator didn't annotate the first root as a chemical entity. Such errors were outlined in the the introduction publication of the CHEMDNER corpus [1]. It is possible to add an additional output node with a recurrent refinement loop into the model, but due to the extremely rare occurrences of overlapping annotations this computation graph won't have enough data to train. We thus consider that the property holds and our problem is almost equivalent to per-character annotation.

Performance

We~~'ve~~ used the F1-score to estimate performance. ~~which is~~ the most common performance metric in NER problems. First of all we immediately noticed that relaxing the weights in word-level embeddings led to fast overfitting. This clearly a result of the number of parameters in the word-embedding layer, which was 10 times greater than in all the other layers combined. We thus focused on using frozen word-vectors exclusively observed no advantage in using LSTM cells over GRUs; on the contrary, GRUs trained faster and performed better ~~in out models~~. The same can be said about CRF layers, that seemed to have little to no effect on the performance other than slowing down the training process. On the other hand, convolutional layers greatly improved performance, as did the

attention-loop in the multi-output topology described earlier. Figure contains complete information on the final model's hyper-parameters and topology. During training the best model (the multi-output model with an attention loop depicted in figure) reached the F1-score of 91% in entity beginning detection and 94% in chemical entity part detection on the validation split. The undetected entity parts were mostly single-token entities for which the start-token d ng failed as well, which was consistent with the way the attention loop was supposed to work. We then evaluated this model on the CHEMDNER test dataset, yielding the F1-score of 90.5% in entity beginning detection and % in chemical entity part detection. Thus we conclude that the model demonstrates near human-level performance.

Accessibility and the user interface

While analysing the NER systems submitted for the CHEMDNER task, we've found that neither the source code, nor the trained models are available for some of the best-performing tools, limiting the ability to use and validate them. We thus made it our priority to publish all the source code needed to train, validate and use our models, by developing a Python package sciNER that can be easily integrated into third-party software. The package is openly available on GitHub. Our final model requires a machine with at least 16GB of RAM to run predictions. To train a custom model, we recommend ~~using~~ a machine with at least 64GB of RAM and a graphics processing unit (GPU) with at least 8GB of VRAM. Although a GPU is not strictly required for training and/or predicting, t ng the model on a 20-core CPU system would require at least 10 times as much time as training it on a single GPU. By default, only CUDA-enabled GPUs are supported unless a user will supply an OpenCL-compatible version of TensorFlow.

Conclusions

Here we've presented our end-to-end neural model for chemical named entities recognition in biomedical texts, trained on the CHEMDNER corpus. With its high F1 score the model reaches near human-level performance, while having no manually asserted rules, we clearly see several directions for further improvements. Fir t all, due to time constraints we haven't investigated many promising hyper-parameter and topology options. Secondly, while avoiding complicated preprocessing has been one of the top-priorities, we still believe that our fine-grained tokenisation strategy overcomplicates the problem (mostly due to over-segmentation) and can be improved by introducing a more intelligent tokeniser. Although such a tokeniser increases the risk of overlapping entities, the addition of a recurrent refiner loop into the model can effectively deal with it. Nevertheless, we find this end-to-end approach extremely promising, ~~b~~ ~~se of~~ its ability to optimise representations and feature extractors automatically with no manual interventions.

Competing interests

The authors declare that they have no competing interests.

Author's contributions

Ilia Korvigo and Mikhail Skoblov conceived and curated the study. Ilia Korvigo, Maxim Holmatov and Anatolii Zaikovskii contributed to data acquisition and supervision. Model development, training and validation was performed by Ilia Korvigo. Software development was performed by Ilia Korvigo and Anton Karazeev. The manuscript was written by Ilia Korvigo and Maxim Holmatov.

Acknowledgements

This work was mostly carried out in collaboration between the Laboratory of Functional Analysis of the Genome (Moscow State Institute of Physics and Technology, Moscow, Russia) and the Laboratory of Microbiological Monitoring and Bioremediation of Soils (All-Russia Institute of Agricultural Microbiology, St. Petersburg, Russia).

Author details

[1]Laboratory of functional analysis of the genome, Moscow Institute of Physics and Technology, Moscow, Russia. [2]All-Russia Institute for Agricultural Microbiology, St. Petersburg, Russia. [3]ITMO University, St. Petersburg, Russia. [4]St. Petersburg State Pediatric Medical University, St. Petersburg, Russia. [5]N.N. Petrov Institute of Oncology, Department of Tumor Biology, St. Petersburg, Russia. [6]St. Petersburg State University, St. Petersburg, Russia.

References

1. Krallinger, M., Rabal, O., Leitner, F., Vazquez, M., Salgado, D., Lu, Z., Leaman, R., Lu, Y., Ji, D., Lowe, D.M., Sayle, R.A., Batista-Navarro, R.T., Rak, R., Huber, T., Rocktäschel, T., Matos, S., Campos, D., Tang, B., Xu, H., Munkhdalai, T., Ryu, K.H., Ramanan, S.V., Nathan, S., Žitnik, S., Bajec, M., Weber, L., Irmer, M., Akhondi, S.A., Kors, J.A., Xu, S., An, X., Sikdar, U.K., Ekbal, A., Yoshioka, M., Dieb, T.M., Choi, M., Verspoor, K., Khabsa, M., Giles, C.L., Liu, H., Ravikumar, K.E., Lamurias, A., Couto, F.M., Dai, H.J., Tsai, R.T.H., Ata, C., Can, T., Usié, A., Alves, R., Segura-Bedmar, I., Martínez, P., Oyarzabal, J., Valencia, A.: The CHEMDNER corpus of chemicals and drugs and its annotation principles. Journal of Cheminformatics **7**(Suppl 1), 1–17 (2015). doi:10.1186/1758-2946-7-S1-S2
2. Leaman, R., Wei, C.-H., Lu, Z., Hunter, L., Neveol, A., Dogan, R.I., Lu, Z., Dogan, R.I., Murray, G., Neveol, A., Lu, Z., Rocktaschel, T., Weidlich, M., Leser, U., Smith, L., Tanabe, L., Ando, R., Kuo, C., Chung, I., Hsu, C., Lin, Y., Klinger, R., Friedrich, C., Ganchev, K., Torii, M., Liu, H., Haddow, B., Struble, C., Povinelli, R., Vlachos, A., Baumgartner, W., Hunter, L., Carpenter, B., Tsai, R., Dai, H., Liu, F., Chen, Y., Sun, C., Katrenko, S., Adriaans, P., Blaschke, C., Torres, R., Neves, M., Nakov, P., Wei, C., Kao, H., Lu, Z., Doğan, R.I., Lu, Z., Wei, C., Kao, H., Lu, Z., Leaman, R., Doğan, R., Lu, Z., Vazquez, M., Krallinger, M., Leitner, F., Valencia, A., Eltyeb, S., Salim, N., Hettne, K., Stierum, R., Schuemie, M., Hendriksen, P., Schijvenaars, B., Mulligen, E., Kleinjans, J., Kors, J., Klinger, R., Kolarik, C., Fluck, J., Hofmann-Apitius, M., Friedrich, C., Jessop, D., Adams, S., Willighagen, E., Hawizy, L., Murray-Rust, P., Kolarik, C., Klinger, R., Friedrich, C., Hoffmann-Apitius, M., Fluck, J., Rebholz-Schuhmann, D., Yepes, A.J., Li, C., Kafkas, S., Lewin, I., Kang, N., Corbett, P., Milward, D., Buyko, E., Beisswanger, E., Hornbostel, K., Kouznetsov, A., Witte, R., Laurila, J., Baker, C., Kuo, C., Clematide, S., Rinaldi, F., Farkas, R., Mora, G., Hara, K., Furlong, L., Rautschka, M., Neves, M., Pascual-Montano, A., Wei, Q., Collier, N., Chowdhury, M., Lavelli, A., Berlanga, R., Rebholz-Schuhmann, D., Yepes, A.J., Mulligen, E.V., Kang, N., Kors, J., Milward, D., Corbett, P., Buyko, E., Beisswanger, E., Hahn, U., Krallinger, M., Leitner, F., Rabal, O., Vazquez, M.,

Oyarzabal, J., Valencia, A., Leaman, R., Wei, C., Lu, Z., Krallinger, M., Rabal, O., Leitner, F., Vazquez, M., Salgado, D., Lu, Z., Leaman, R., Lu, Y., Ji, D., Lowe, D., Sayle, R., Batista-Navarro, R., Rak, R., Huber, T., Rocktaschel, T., Matos, S., Campos, D., Tang, B., Xu, H., Munkhdalai, T., Ryu, K., Ramanan, S., Nathan, S., Zitnik, S., Bajec, M., Weber, L., Irmer, M., Akhondi, S., Kors, J., Xu, S., An, X., Sikdar, U., Ekbal, A., Yoshioka, M., Dieb, T., Choi, M., Verspoor, K., Khabsa, M., Giles, C., Liu, H., Ravikumar, K., Lamurias, A., Couto, F., Dai, H., Tsai, R., Ata, C., Can, T., Usie, A., Alves, R., Segura-Bedmar, I., Martinez, P., Oryzabal, J., Valencia, A., Hastie, T., Tibshirani, R., Friedman, J., Leaman, R., Gonzalez, G., Wei, C.-H., Harris, B., Kao, H.-Y., Lu, Z., Timberlake, K., Porter, M., Lowe, D., Corbett, P., Murray-Rust, P., Glen, R., Sohn, S., Comeau, D., Kim, W., Wilbur, W., Hsu, C., Chang, Y., Kuo, C., Lin, Y., Huang, H., Chung, I., Pearl, J., Coletti, M., Bleich, H., de Matos, P., Dekker, A., Ennis, M., Hastings, J., Haug, K., Turner, S., Steinbeck, C., Chae, J., Jung, Y., Lee, T., Jung, S., Huh, C., Kim, G., Kim, H., Oh, H., Buyko, E., Tomanek, K., Hahn, U., Zhang, S., Elhadad, N., Leaman, J., Wei, C., Harris, B., Li, D., Berardini, T., Huala, E., Kao, H., Lu, Z., Lu, Z., Kao, H., Wei, C., Huang, M., Liu, J., Kuo, C., Hsu, C., Tsai, R., Dai, H., Okazaki, N., Cho, H., Gerner, M., Solt, I., Agarwal, S., Liu, F., Vishnyakova, D., Ruch, P., Romacker, M., Rinaldi, F., Bhattacharya, S., Srinivasan, P., Liu, H., Torii, M., Matos, S., Campos, D., Verspoor, K., Livingston, K., Wilbur, W.: tmChem: a high performance approach for chemical named entity recognition and normalization. Journal of Cheminformatics **7**(Suppl 1), 3 (2015). doi:10.1186/1758-2946-7-S1-S3

3. Akhondi, S.A., Hettne, K.M., Van Der Horst, E., Van Mulligen, E.M., Kors, J.A.: Recognition of chemical entities: Combining dictionary-based and grammar-based approaches. Journal of Cheminformatics **7**(Suppl 1), 1–11 (2015). doi:10.1186/1758-2946-7-S1-S10
4. Khabsa, M., Giles, C.L.: Chemical entity extraction using CRF and an ensemble of extractors. Journal of Cheminformatics **7**(Suppl 1), 1–9 (2015). doi:10.1186/1758-2946-7-S1-S12
5. Dai, H.J., Lai, P.T., Chang, Y.C., Tsai, R.T.H.: Enhancing of chemical compound and drug name recognition using representative tag scheme and fine-grained tokenization. Journal of Cheminformatics **7**(Suppl 1), 1–10 (2015). doi:10.1186/1758-2946-7-S1-S14
6. Xu, S., An, X., Zhu, L., Zhang, Y., Zhang, H.: A CRF-based system for recognizing chemical entity mentions (CEMs) in biomedical literature. Journal of Cheminformatics **7**(Suppl 1), 1–9 (2015). doi:10.1186/1758-2946-7-S1-S11
7. Tang, B., Feng, Y., Wang, X., Wu, Y., Zhang, Y., Jiang, M., Wang, J., Xu, H.: A comparison of conditional random fields and structured support vector machines for chemical entity recognition in biomedical literature. Journal of Cheminformatics **7**(Suppl 1), 4–9 (2015). doi:10.1186/1758-2946-7-S1-S8
8. Lowe, D.M., Sayle, R.A.: LeadMine: A grammar and dictionary driven approach to entity recognition. Journal of Cheminformatics **7**(Suppl 1), 1–9 (2015). doi:10.1186/1758-2946-7-S1-S5
9. Lu, Y., Ji, D., Yao, X., Wei, X., Liang, X.: CHEMDNER system with mixed conditional random fields and multi-scale word clustering. Journal of Cheminformatics **7** (2015). doi:10.1186/1758-2946-7-S1-S4
10. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed Representations of Words and Phrases and their Compositionality (2013). 1310.4546
11. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space, 1–12 (2013). doi:10.1162/153244303322533223. 1301.3781
12. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching Word Vectors with Subword Information (2016). doi:1511.09249v1. 1607.04606
13. Wieting, J., Bansal, M., Gimpel, K., Livescu, K.: Charagram: Embedding Words and Sentences via Character n-grams. Emnlp-2016, 1504–1515 (2016). 1607.02789
14. Lopez, M.M., Kalita, J.: Deep Learning applied to NLP (2017). 1703.03091
15. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. Advances in Neural Information Processing Systems (NIPS), 3104–3112 (2014). doi:10.1007/s10107-014-0839-0. 1409.3215
16. Sætre, R., Yoshida, K., Yakushiji, A., Miyao, Y., Matsubayashi, Y., Ohta, T.: AKANE System : Protein-Protein Interaction Pairs in the BioCreAtIvE2 Challenge , PPI-IPS subtask. Proceedings of the Second BioCreative Challenge Evaluation Workshop (January), 4–6 (2007)
17. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling, 1–9 (2014). 1412.3555
18. Jozefowicz, R., Zaremba, W., Sutskever, I.: An Empirical Exploration of Recurrent Network Architectures. Proceedings of The 32nd International Conference on Machine Learning **37**, 2342–2350 (2015). doi:10.1109/CVPR.2015.7298761. 1512.03385
19. Pennington, J., Socher, R., Manning, C.: Glove: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1532–1543 (2014). doi:10.3115/v1/D14-1162. 1504.06654
20. Ling, W., Luís, T., Marujo, L., Astudillo, R.F., Amir, S., Dyer, C., Black, A.W., Trancoso, I.: Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation. arXiv preprint (2015). doi:10.18653/v1/D15-1176. 1508.02096
21. Huang, Z., Xu, W., Yu, K.: Bidirectional LSTM-CRF Models for Sequence Tagging (2015). 1508.01991
22. Gal, Y., Ghahramani, Z.: A Theoretically Grounded Application of Dropout in Recurrent Neural Networks (2015). doi:10.1201/9781420049176. 1512.05287
23. Chollet, F., et al.: Keras. GitHub (2015)
24. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Software available from tensorflow.org (2015). https://www.tensorflow.org/
25. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization, 1–15 (2014). doi:10.1145/1830483.1830503. 1412.6980

Figures

Figure 1 The architecture of the final ChemPred model. The figure illustrates the topology and hyper-parameters used in the best-performing model, except for the input dropout rates (0.3 in convolutional and 0.1 in recurrent layers) and recurrent dropout rates (0.1 in all recurrent layers). The encircled multiplication sign represents time-step-wise multiplication node