Team EP at TAC 2018: Automating data extraction in systematic reviews of environmental agents

Artur Nowak and **Paweł Kunstman**
Evidence Prime
{artur.nowak,pawel.kunstman}@evidenceprime.com

Abstract

We describe our entry for the Systematic Review Information Extraction track of the 2018 Text Analysis Conference. Our solution is an end-to-end, deep learning, sequence tagging model based on the BI-LSTM-CRF architecture. However, we use interleaved, alternating LSTM layers with highway connections instead of the more traditional approach, where last hidden states of both directions are concatenated to create an input to the next layer. We also make extensive use of pre-trained word embeddings, namely GloVe and ELMo. Thanks to a number of regularization techniques, we were able to achieve relatively large capacity of the model (31.3M+ of trainable parameters) for the size of training set (100 documents, less than 200K tokens). The system's official score was 60.9% (micro-F1) and it ranked first for the Task 1. Additionally, after rectifying an obvious mistake in the submission format, the system scored 67.35%.

1 Introduction

Systematic reviews play a fundamental role in health decision-making. They offer a comprehensive and unbiased synthesis of human knowledge on a given subject, produced through a standarized, transparent and scrupulous process. The result is crucial for creation of trustworthy health hazard assessments and clinical practice guidelines.

Systematic review process strives to achieve perfect recall. To accomplish that, the net is first cast wide – thousands of papers are retrieved and are manually sifted. The included citations are then subject to data extraction, in which the information relevant to the given research question is selected. As advocated by organizations setting standards for systematic reviews, such as Cochrane, both steps should be performed independently by several researchers to rule out human errors or biases. This further increases the labor intensiveness of the process.

As a result, many reviews take up to two years to finish and may be already outdated at the moment of publication. The increase in the scientific output (more than 800,000 papers are indexed by MEDLINE every year), as well as questions requiring rapid responses (e.g. in cases of chemical spills), demand solutions for expediting the process without sacrificing quality. Recent advances in Natural Language Processing technologies may be the answer.

In 2018, the National Toxicology Program (NTP) and the Environmental Protection Agency (EPA) co-organized Systematic Review Information Extraction (SRIE) track as a part of Text Analysis Conference (TAC). The objective of the track was to evaluate automatic information extraction approaches that could aid in performing systematic reviews of environmental agents.

The track consisted of two tasks:

1. Entity recognition of experimental design factors for the categories of exposure, animal group, dose group and endpoint.
2. Relation extraction between experimental design factors from Task 1.

This paper describes our approach for Task 1, the only one we participated in.

2 Datasets and pre-processing

2.1 Characteristics of the datasets

The organizers prepared two datasets: a training set consisting of 100 annotated "Material and methods" sections, extracted from PubMed Central articles, along with their identifiers; a test set of 100 such texts, for which annotation were not released to the participants until the submission deadline. To discourage manual annotation, the published test set contained further 344 texts that were not used in evaluation.

The training set contained nearly 7K sentences and 152K words (space-separated, our custom tokenization rules produced nearly 200K tokens). There were 15,253 mentions in the training set. Total of 24 entity classes were used, with the most frequent being: ENDPOINT (29%), TESTARTICLE (13%) and SPECIES (11%). For six classes, there were less than 50 examples in the dataset: TESTARTICLEVERIFICATION (6), TIMEATLASTDOSE (23), TESTARTICLEPURITY (28), CELLLINE (39), SAMPLESIZE (45), TIMEATFIRSTDOSE (47).

Apart of the sheer number of classes, they also had complex semantics. For instance, the least frequent class (TESTARTICLEVERIFICATION) was characterized in the annotation guidelines as: "Annotate the statement which indicates that the chemical was confirmed. This may refer to a third party assessment where another company confirmed the chemical.". The decision boundary for even the most common classes is far from trivial. For example, TESTARTICLE is defined as "the exposure (chemical or stressor) for which the experimental design is intended to evaluate [...] Reagents used for endpoint analysis [...] are not annotated as test articles.".

Furthermore, the mention endpoints often didn't agree with common tokenization rules. For instance, in cases like "15GD" (which is a shortcut for "15th gestational day"), "15" and "GD" were annotated as TIMEATDOSE and TIMEUNITS, respectively. In GROUPNAME "Control-Sal", "Sal" was additionally annotated as VEHICLE (saline).

This highlights another trait of the dataset: a large number of overlapping and discontiguous mentions. Among ENDPOINTs, 42% of examples were discontiguous, i.e. they consisted of multiple spans that were not adjacent. Moreover, the discontiguous mentions often spanned multiple sentences. Furthermore, nearly 60% of ENDPOINTs had at least one character in common with another ENDPOINT mention. Likewise, 16% of GROUPNAME mentions were discontiguous, similar number overlapped with another GROUPNAME and more than twice the number overlapped with a TESTARTICLE.

2.2 Data pre-processing and augmentation

Texts were first broken into sentences using Punkt tokenizer from NLTK (Kiss and Strunk, 2006). They were then split into tokens with spaCy library (Honnibal and Montani, 2017), using a number of custom rules to ensure that token boundaries line up with the mention spans from the training set.

The spans were also stripped of trailing and leading white-space and punctuation (we observed that both were inconsistently used in the training annotations). In the cases, in which the offsets didn't match the tokens (e.g. due to inconsistent handling of new line characters in the training files or annotations that began mid-word), they were corrected manually.

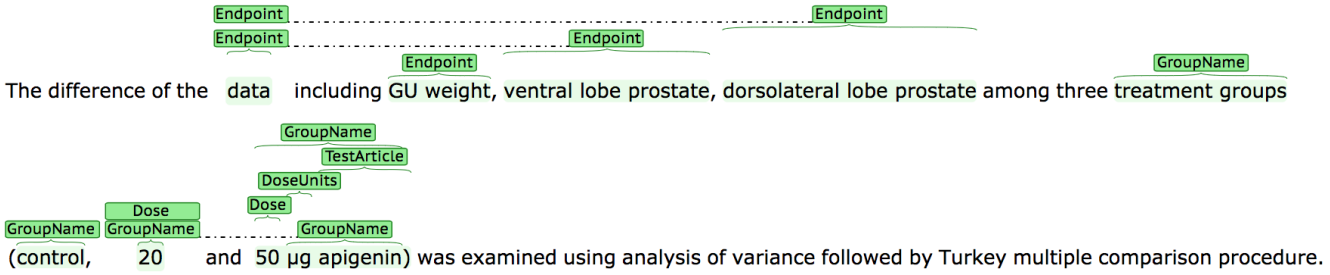
Although, as previously mentioned, Task 1 involved a large number of overlapping and discontiguous entities, for this year's evaluation we focused on predicting 'linear' mentions. While many techniques exist for handling both overlapping and discontiguous mentions – some of them were explored as part of TAC 2017 Adverse Drug Reaction Extraction from Drug Labels track (Roberts et al., 2017) – only some address mentions spanning multiple sentences. This problem is structurally similar to co-reference resolution, so perhaps methods used for this task can be explored in the future. Nevertheless, we feel that creating a robust model for the simplified problem is still a prerequisite for tackling the full challenge.

Furthermore, the official evaluation metric (micro-averaged F1) was calculated using partial matches with the gold annotations. The threshold was initially set to 40%, with 50% used in the final scoring. This effectively meant that it sufficed to detect just the longest segment for the most common case: multiple discontiguous ENDPOINTs that shared some 'prefix', but all had unique 'root' word, e.g. "genes [...] functions", "genes [...] processes".

Thus, we decided to transform the task into a sequence tagging problem using IOB2 (i.e. B- tag is used in the beginning of every mention) annotation scheme. Discontiguous mentions were joined if the distance between consecutive spans was ≤ 5 characters. The remaining ones were treated as separate mentions (with the same class).

If a sentence contained overlapping mentions, it was emitted for every 'level' of mentions with different classes. In other words, if there were n classes associated with a token, the sentence was outputted n times. First, mentions with the lowest number of spans and the greatest total length were chosen, then the ones that appeared second in such order etc. Figure 1 illustrates the encoding.

Finally, we did a round-trip (or: back and forth) translations of the training text (Ostyakov, 2018): through French (using Microsoft Translator API) and Russian (using Yandex API). The major difficulty in



(a) An example sentence, featuring both discontiguous and overlapping annotations

B-Endpoint B-Endpoint I-Endpoint B-Endpoint I-Endpoint I-Endpoint B-Endpoint I-Endpoint I-Endpoint
The difference between data , including GU weight , ventral lobe prostate , dorsolateral lobe prostate between 3
B-GroupName I-GroupName B-GroupName B-GroupName B-GroupName I-GroupName I-GroupName
treatment groups (control , 20 and 50 µg apigenin) was examined using the analysis of the variance
followed by Turkey multiple comparison procedure.

B-Endpoint B-Endpoint I-Endpoint B-Endpoint I-Endpoint I-Endpoint B-Endpoint I-Endpoint I-Endpoint
The difference between data , including GU weight , ventral lobe prostate , dorsolateral lobe prostate between 3
B-GroupName I-GroupName B-GroupName B-Dose B-Dose B-DoseUnits B-TestArticle
treatment groups (control , 20 and 50 µg apigenin) was examined using the analysis of the variance followed
by Turkey multiple comparison procedure.

(b) Two sentences that jointly represent the input

Figure 1: An illustration of the data representation

performing this task was preserving word alignments, so the mentions' character offsets could be restored. Hence, we replaced the mentions with encoded, untranslatable strings during the translation phase. This way, their position in the result could be easily determined. For total of 3 documents it wasn't possible to recover the offsets, so the original texts were used. This way, we were able to triple the number of training documents. Figure 2 shows this technique in action. Neither extra sentences for overlapping mentions, nor round-trip translations were used to evaluate out-of-fold predictions during cross-validation.

Each token was also supplemented with its relative position in document, rounded to two decimal places. For one of the runs, we downloaded paper titles and abstracts using PubMed API. We then identified all the abbreviations in the training text and the abstract using a rule-based algorithm (S Schwartz and Hearst, 2003). For all the tokens (that weren't stop words) we added a Boolean feature for whether they (or their expansions) appear in the paper title. We refer to this feature as `in_title` in the description below.

3 Model architecture and training

Our solution is an end-to-end, deep learning, sequence tagging model based on the BI-LSTM-CRF (Huang et al., 2015) architecture. However, we use stacked, alternating LSTM layers (Zhou and Xu, 2015) with highway connections (He et al., 2017; Srivastava et al., 2015) instead of the more traditional approach, where last hidden states of both directions are concatenated to create an input to the next layer. The implementation was based on the AllenNLP library (Gardner et al., 2017). The model architecture is shown in Figure 3.

Embeddings layer: Each token is represented by 1452 dimensional vector, consisting of:

- 300-dimensional GloVe (Pennington et al., 2014) embedding (cased, trained on 840B tokens from Common Crawl).
- 1024-dimensional ELMo (Peters et al., 2018) embedding that was originally trained on 5.5B tokens.
- 128-dimensional output from a character-level, one-layer CNN.

We fine-tuned ELMo vectors for two epochs on 48,141 PubMed abstracts (full texts, where available, total of nearly 365M tokens; training and test documents were explicitly excluded). This resulted in drop in perplexity (measured on the Task 1 training set) from more than 300 to 27. This step is especially important, because of the custom tokenization rules that we mentioned in Section 2.2 – ELMo contains a character-level encoder.

Generally, ELMo exposes hidden state from all three layers of the model. The final representation is a weighted sum of these three vectors, where the weights are learned during training for the downstream task.

Briefly, MWCNT trace metal contamination was 0.78%, with sodium (0.41%) and iron (0.32%) being the major metal contaminants. Average MWCNT surface area measured by nitrogen absorption-desorption technique (Brunauer-Emmett-Teller method, BET) was 26 m2/g. MWCNT median length was 3.86 µm and count mean diameter was 49 ± 13.4 (mean ± S.D.) nm, as determined by scanning electron microscopy of MWCNTs suspended in dispersion medium as described below [12].

(a) The original text

In short, MWCNT contamination of trace metals was 0.78%, sodium (0.41%) and iron (0.32%) being the main metal contaminants. The mean surface area MWCNT measured by the nitrogen absorption-desorption technique (Brunauer-Emmett-Teller, BET) was 26 m2/g. MWCNT The median length was 3.86 µm and the average counting diameter was 49 ± 13.4 (Average ± S.D.) NM, as determined by scanning electron microscopy of MWCNTs in suspension in a dispersion medium as described below [12].

(b) Translated to French and back to English

Briefly, MWCNT trace metal contamination was 0.78%, sodium (0.41%) and iron (0.32%) are the major metal contaminants. Average MWCNT surface area measured by adsorption-desorption of nitrogen method (brunauer-Emmett-teller method, Bet) was 26 m2/g. MWCNT average length was 3.86 µm and calculate the average diameter was 49 ± 13.4 (mean ± S. D.) nm, as determined using scanning electron microscopy MWCNTs suspended in the dispersion medium as described below [12].

(c) Translated to Russian and back to English

Figure 2: Example of the round-trip translation data augmentation technique at work. Overall, some degree of paraphrasing is achieved at the cost of incorrect replacements (absorption ≠ adsorption) in the most troublesome parts.

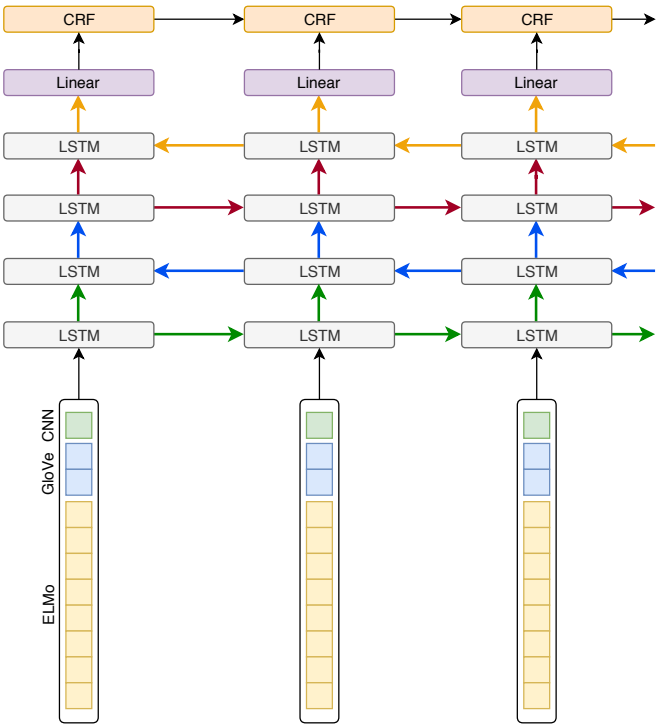


Figure 3: The model architecture. Coloured arrows share dropout masks.

However, in this work we set the L2 regularization parameter $\lambda = 1$ for the weights, which effectively leads to a simple average over the layers. Both GloVe and ELMo embeddings were frozen during training of the model.

The input to the CNN is 16-dimensional (learned) embedding of token characters. It is then passed to one convolution layer with kernel size of 3 (i.e. trigrams) and 128 filters. ReLU was used as the activation function. The concatenated embeddings were followed by a dropout layer (with $d_1 = 0.75$). Then, the remaining features were added – in case of our best run, it was only the relative offset information.

LSTM layers: The token representations are then fed into four, alternating layers of LSTM with highway connections and hidden state of size 800. Highway connections (Srivastava et al., 2015) are essentially an extension to the LSTM (Hochreiter and Schmidhuber, 1997) that adds gated combination of the 'traditional' LSTM cell output and a linear transformation of its input:

$$h_t = w_t \odot \tilde{h}_t + (1 - w_t) \odot W_h [x_t] \quad (1)$$

$$\tilde{h}_t = o_t \odot \tanh(c_t) \quad (2)$$

$$\begin{aligned} c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \\ \tilde{c}_t &= \tanh(W_c [1; h_{t-1}; x_t]) \\ w_t &= \sigma(W_w [1; h_{t-1}; x_t]) \\ o_t &= \sigma(W_o [1; h_{t-1}; x_t]) \\ f_t &= \sigma(W_f [1; h_{t-1}; x_t]) \\ i_t &= \sigma(W_i [1; h_{t-1}; x_t]) \end{aligned} \quad (3)$$

where $\odot$ is element-wise multiplication, $\sigma(\cdot)$ is element-wise sigmoid function and $[1; a; b]$ is a vector created by horizontally stacking vectors $[1]$ (for a bias term), a and b. Here, $\tilde{h}_t$ (2) is the hidden state (time-step output) of the LSTM cell, as originally defined. Equation 1 is the new definition of the hidden state that uses the new gate defined in 3.

Moreover, $d_2 = 0.5$ dropout with a constant (i.e. the same for all time steps) mask is applied to h_t. This realizes variational inference based dropout introduced in (Gal and Ghahramani, 2016). The only difference is that the same mask is used both for the 'output' (going

to the next layer) and recurrent connections:

$$h'_t = z \odot h_t$$

CRF: Traditional (Srivastava et al., 2014) $d_3 = 0.75$ dropout is applied to the last hidden state of the LSTM and the result is passed to a linear layer that projects the vector into a 49-dimensional (number of tags) space.

This vector is used as input to a linear-chain Conditional Random Field (Lafferty et al., 2001). The Viterbi algorithm is used for decoding, with constraints in place to penalize disallowed tag transitions (e.g. "B-ENDPOINT → I-TESTARTICLE").

The model was trained to minimize negative log likelihood produced by the CRF using Adam optimizer (Kingma and Ba, 2014), with starting learning rate of 0.001 and batch size of 32. 5-fold cross validation was used, with validation metric being micro-averaged F1 using exact span matching. The learning rate was halved if the validation metric didn't improve by at least 0.001 in 5 epochs. Early stopping was used with 10-epoch patience.

4 Results

The official evaluation script scored the submissions by creating a maximum match between gold annotations and predictions that intersect with them at least in 50% (character-wise) and calculating micro-averaged F1 measure for such mapping. We submitted 3 runs:

1. Result of the model described in Section 3 (without the `in_title` feature), trained on the whole training set for 23 epochs.

2. A majority vote of models (without the `in_title` feature) trained during 5-fold CV. In case of ties, the first tag in lexicographic order was taken. The F1 score (using the official evaluation script) of the out-of-fold predictions was 69.90% for 50% similarity threshold.

3. A majority vote of 10 models: five from the run 2 and another five, using `in_title` feature, trained during 5-fold CV. The local CV score for the 5 `in_title` models was 70.11% for 50% threshold.

The submission files contained lists of mentions, along with the detected character offsets. There was some confusion how to count the line endings for the purpose of offset calculation. Our first submission treated all line endings (including Windows-style CRLF) as a single character. After contacting the organizers, this turned out to be incorrect. Our second and final submission counted all line endings as two characters, overlooking the fact that 10 out of 100 test files used single-character, Unix-style line endings.

We feel that the score obtained after rectifying this obvious mistake is more representative of the overall system performance. Therefore, we report both the official score (from our second submission) and the result of re-scoring our second submission after replacing these 10 files with the ones from our first submission. The results are presented in Tables 1 and 2.

Run ID	Official score	Score with correction
`ep_1`	60.29	66.76
`ep_2`	**60.90**	**67.35**
`ep_3`	60.61	67.07

Table 1: The scores of our three submitted runs for similarity threshold 50%.

Mention class	No. examples	F1 (5-CV)	F1 (Test)
Total	15 265	69.90	67.35
Endpoint	4411	66.89	61.47
TestArticle	1922	63.29	64.19
Species	1624	95.33	95.95
GroupName	963	67.08	62.40
EndpointUnitOfMeasure	706	42.27	40.41
TimeEndpointAssessed	672	57.27	55.51
Dose	659	78.47	75.85
Sex	612	96.27	98.36
TimeUnits	608	68.03	61.26
DoseRoute	572	69.24	69.80
DoseUnits	493	77.50	72.33
Vehicle	440	63.03	67.15
GroupSize	387	77.79	75.74
Strain	375	78.56	76.00
DoseDuration	216	59.78	56.80
DoseDurationUnits	204	57.83	56.60
TimeAtDose	117	34.29	35.68
DoseFrequency	96	41.56	59.78
TimeAtFirstDose	47	3.92	0.00
SampleSize	45	43.84	50.00
CellLine	39	50.00	50.77
TestArticlePurity	28	34.04	60.00
TimeAtLastDose	23	0.00	0.00
TestArticleVerification	6	0.00	0.00

Table 2: Detailed results of our best run (after correcting the submission format), along with numbers of mentions in the training set.

ID	5-fold CV	Δ	Single model	Δ	Ensemble	Δ
LSTM-800	70.56	0.66	67.54	0.78	67.65	0.30
LSTM-400	70.50	0.60	**67.59**	0.83	**68.00**	0.65
IN-TITLE	70.11	0.21	N/A	N/A	67.52	0.17
SUBMISSION	69.90	–	66.76	–	67.35	–
NO-HIGHWAY	69.72	−0.18	66.42	−0.34	66.64	−0.71
NO-OVERLAPS	69.46	−0.44	65.07	−1.69	66.47	−0.88
LSTM-400-DROPOUT	69.45	−0.45	65.53	−1.23	67.28	−0.07
NO-TRANSLATIONS	69.42	−0.48	65.92	−0.84	67.23	−0.12
NO-ELMO-FINETUNING	67.71	−2.19	65.16	−1.60	65.42	−1.93

Table 3: The estimation of impact of various design choices on the final result. The entries are sorted by the out-of-fold scores from CV. The **SUBMISSION** here uses score from `ep_1` run for the single model and `ep_2` for the ensemble performance.

5 Discussion

To better estimate the impact of the described techniques on our final result, we performed a series of ablation studies. We re-trained our best model (`ep_2`), removing one of its feature at time:

1. **LSTM-800**: Stacked BI-LSTM with two layers and hidden size of 800 (instead of four alternating LSTM layers).
2. **LSTM-400**: Stacked BI-LSTM with two layers of size 400.
3. **IN-TITLE**: Majority vote of 5 `in_title` models – in other words, `ep_3` submission without ensembling with `ep_2` models.
4. **NO-HIGHWAY**: Traditional LSTM cell definition, without the highway connection.
5. **NO-OVERLAPS**: Without extra sentences generated for overlaps (as in Figure 1).
6. **LSTM-400-DROPOUT**: Stacked BI-LSTM with two layers of size 400 and dropout only between LSTM layers, as proposed in (Zaremba et al., 2014).
7. **NO-TRANSLATIONS**: Without the round-trip translation augmentation (see Figure 2).
8. **NO-ELMO-FINETUNING**: ELMo vectors as published by (Peters et al., 2018), without fine-tuning on the PubMed data.

To enable fair comparison of the ablated models with the submitted ones, we trained them both for 23 epochs (as was the case with the `ep_1` submission) and during 5-fold cross-validation, using the same learning rate schedule and early stopping strategy, ensembling the results the same way as was done for the `ep_2`. The results are presented in Table 3.

Perhaps the most striking thing about the ablation results is that the 'traditional' LSTM layout outsperformed the 'alternating' one we chose for our submission. Our decision at that time was based on a score calculated on 20% validation set, due to time constraints. Cross-validation results clearly show the winner here, although their translation into performance on the test set is inconsistent. Apart of the flipped results of the LSTM-800 and the LSTM-400, small differences in CV score are sometimes associated with large discrepancies in test set performance. This is mostly due to small size of the data set (low precision of the estimate), stochastic nature of the training process and hyperparameters (such as number of epochs for single models) aligning better with some of the models.

The results of ensembling are also varied. More testing is required to pinpoint the actual impact on the final score, by re-training the models several times with different random seeds and averaging the results. Some ablated models that perform poorly in the single-model scenario (e.g. NO-OVERLAPS, LSTM-400-DROPOUT) are able to regain a lot of accuracy when ensembled. Also, our data augmentation technique (NO-TRANSLATIONS) seem to have far smaller impact on the final score then we expected. Finally, the fact that our third run (`ep_3`) fared worse than the second one is explained by the fact that we needlessly included the models from `ep_2` in the ensemble (IN-TITLE). This makes us optimistic about future work on

including title information in the model.

We presented our solution for automating data extraction in systematic reviews of environmental agents. Although we made considerable simplifications to the original problem setting, our error analysis shows that the system already delivers substantial value for potential users.

We plan to further explore the problem of structured predictions, taking into the consideration discontiguous and overlapping mentions. We also would like to apply the experience we gained from working on this problem to the Task 2 of the Track.

Our current best model completely ignores the document context. This information is intuitively the single most important feature for many classes, as the whole articles will generally describe the same SPECIES, TESTARTICLES etc.

However, our extensive exploration of adding the document context by means of attention layers, carrying over hidden state, memory cells or even handcrafted features led to unsatisfactory results. We think that the major cause of this situation is overfitting, due to a small number of documents. We also plan to work on alleviating this issue in our further work.

References

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *NIPS*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. AllenNLP: A deep semantic natural language processing platform.

Luheng He, Kenton Lee, Mike Lewis, and Luke S. Zettlemoyer. 2017. Deep semantic role labeling: What works and what's next. In *ACL*.

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

Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, arxiv:abs/1508.01991.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Tibor Kiss and Jan Strunk. 2006. Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4):485–525.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Pavel Ostyakov. 2018. A simple technique for extending dataset. https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/discussion/48038 [Accessed: 2018-10-30].

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Kirk Roberts, Dina Demner-Fushman, and Joseph M. Tonning. 2017. Overview of the TAC 2017 adverse reaction extraction from drug labels track. In *Proceedings of the 2017 Text Analysis Conference, TAC 2017, Gaithersburg, Maryland, USA, November 13-14, 2017*. NIST.

Ariel S Schwartz and Marti Hearst. 2003. A simple algorithm for identifying abbreviation definitions in biomedical text. 4:451–62.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks. *CoRR*, abs/1505.00387.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *CoRR*, abs/1409.2329.

Jie Zhou and Wei Xu. 2015. End-to-end learning of semantic role labeling using recurrent neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1127–1137. Association for Computational Linguistics.