Catalytic DS at BioCreative VII: DrugProt Track

A Syntactically-guided BiLSTM with BERT-derived Word Vectors

Dennis N. Mehay, PhD and Kuan-Fu Ding, PhD
Catalytic Data Science, Inc. (Bioinformatics and Data Science Division), Charleston, SC, USA

***Abstract*—Team "Catalytic" submitted two system runs to the DrugProt Main Track competition. First, as a strong baseline, we reimplemented BioBERT (1) in PyTorch. Second, noting poor performance of the BioBERT system when drugs and proteins were distant from one another in the sentence, we implemented a syntactically-guided bidirectional LSTM (Syn-BiLSTM) classification model (2) using the sequence of tokens along the shortest path between the entities in a dependency parse of the input sentence. The Syn-BiLSTM is trained as a classifier on this sequence using the final hidden states in the forward and backward directions. Evaluating as a classifier that predicts either a null interaction or one of the 13 specific interaction types, we show that the BiLSTM is more accurate than BioBERT at longer distances, while preserving accuracy at shorter distances, and, in a precision/recall evaluation setting, scores higher in precision and F1, although lower in recall. Additionally, we surpass the overall mean system precision and F1 performance as reported in the BioCreative VII official evaluation (3). Neither of our systems makes any explicit use of ontologies, gazetteers, or any other distillation of scientific knowledge about drug-protein interactions, but we anticipate that using such information would improve performance. The only heuristic constraints on entity interactions is that we assume interactions occur within (auto-detected) sentence boundaries. Finally, we discuss specific interaction type performance, its potential impact in a practical system, as well as future directions for research.**

Keywords—information extraction; drug-protein interaction; drug-gene interaction; classification; deep learning; LSTMs; transformers; BERT; syntactic dependency parsing

I. Introduction

The life sciences literature is vast and growing exponentially year over year, necessitating automated means of surveying the changing landscape of medical facts, such as interactions between drugs and genes and their protein products. Here we describe the system submissions of Team "Catalytic" for the BioCreative VII DrugProt Main Track (3), which, following the ChemProt task (4), seeks to test the state of the art in detecting fine-grained drug/chemical interactions with genes and their protein products (henceforth, "drug-protein interactions") from scientific publication texts.

We submitted two system runs:

- A PyTorch reimplementation of the BioBERT system (1). BioBERT is a standard BERT transformer-based sequence classifier (5), but where two whole named-entity phrases (here a chemical/drug entity and a gene/protein entity) are masked with special tokens, "@CHEM#" and "@GENE#". The system is then trained to classify the whole sequence using the special "[CLS]" token's activation vector which is passed through a linear classification layer. Error gradients are propagated back through the whole network, from the classification layer down through the underlying BERT network.
- Preliminary analysis of the BioBERT system results on a held-out portion of the development set revealed that classification accuracy drops considerably as character distance between drug-protein pairs increases (see Results). Because of this, we explicitly parse each input sentence and follow the parse path between the drug-protein entity pairs, emitting the token vectors along the way as features to a bidirectional LSTM classification model (2).

Evaluating as a classifier that predicts either a null interaction or one of the 13 specific interaction types, we show that the BiLSTM is more accurate than BioBERT at longer distances, while preserving accuracy at shorter distances, and, in a precision/recall evaluation setting, scores higher in precision and F1, although lower in recall. Additionally, we surpass the overall mean system precision and F1 performance as reported in the BioCreative VII overview (3).

Neither system uses named entity resolution, ontologies, gazetteers, tables of known drug-protein interactions, or any form of extra scientific knowledge such as drug or protein structural information. Incorporating such information—either directly as a rule-based decision process, or indirectly as features to the classifiers—would likely improve performance. The only heuristic constraints on entity interactions are that interactions are assumed to occur within (auto-detected) sentence boundaries. We discuss implementation details further in the Methods Section. We give results and further analyze performance in the Results and Analysis section, and discuss improvements and future research directions in Discussion.

II. Methods

Here we give more details about data preprocessing, as well as architectural and training details of our two submitted systems. We refer the reader to the official DrugProt report for more details about the dataset, annotation guidelines, etc. (3).

A. Data and Preprocessing

The DrugProt data (3) consists of 3,500 training abstracts and 750 development abstracts. In each of these datasets, drug and protein named entities have been manually marked up with character boundaries, and drug-protein interactions (if any) are manually annotated. The test set consists of 10,750 abstracts, of which only 750 are intended for evaluation. The identities of these 750 abstracts (in which entity and interaction type labels are annotated manually) were withheld from participants, as were the gold-standard relation labels. We further subdivided the development data into a development tuning (for measuring progress during training) and development test (for blind testing and analysis). We split out approximately 20% of the development abstracts into the dev-test subset in a grouped (by abstract ID), stratified (by relation and entity type) fashion. For this, we used a modified version of the code here: https://github.com/joaofig/strat-group-split

Each abstract was split into sentences using spaCy (9), except in cases where named entities would cross sentence split boundaries or (in the training and development sets only) where pairs of named entities that are labelled with an interaction would be separated into two auto-detected sentences. The training and development events are comprised of (1) each pair of entities that are labelled with an interaction type (AGONIST, INHIBITOR, etc.), as well as (2) all drug-protein pairs that are co-mentioned in the same sentence but that were not labelled as interacting. The latter cases are given the implicit null label, "NEGATIVE". At test time, all drug-protein pairs in each auto-detected sentence, are classified with either an interaction label or NEGATIVE. When submitting responses for the official evaluation, we simply did not produce a label for any drug-protein pair that was classified as having a NEGATIVE interaction type.

Table I gives the summary statistics for each dataset (training, dev-tuning, dev-test and test). In this table, only sentences that have at least one drug-protein entity pair are counted.

TABLE I. DRUGPROT DATA

Features of Dataset	Datasets			
	Training	***Dev-Tuning***	***Dev-Test***	***Test***[a]
Abstracts	3,500	598	152	10,750
Tot. Entities	89,529	15,176	3,682	310,805
Drug	46,274	7,930	1,923	143,767
Protein	43,255	7,246	1,759	167,038
Sentences[b]	13,014	2,193	508	48,976
Relations	66,600	11,397	2,428	N/A
Non-neg.	17,273	3,019	738	N/A
Negative	49,327	8,378	1,690	N/A

a. For all but 750 abstracts, the Test entities are automatically assigned.
b. Auto-detected sentences, only counting those with drug-protein pairs.

B. Baseline System: BioBERT ("Run 2")

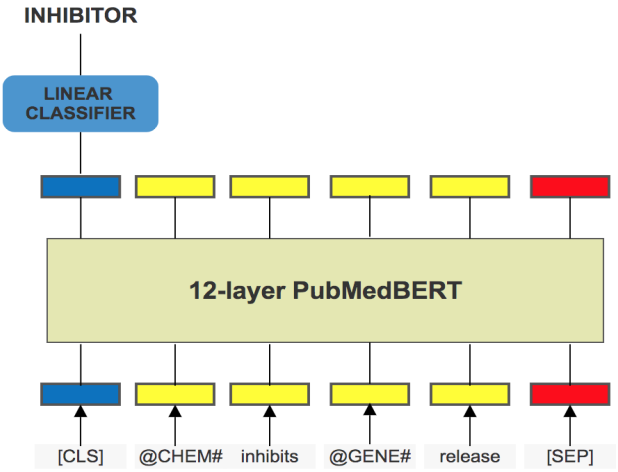
Following Lee and colleagues (1), we implement a sequence classification system that uses a BERT transformer to aggregate whole sentence information into the special "[CLS]" token at the start of every sentence encoding. BioBERT is identical in every way to sequence classification tasks from the original BERT paper (5, Section 4), except that the input is rewritten to mask each of a pair of named entity phrases whose relationship is to be determined with special tokens. In our case, we use "@CHEM#" and "@GENE#" to mask each drug/chemical and protein/gene named entity phrases, resp. The top-level hidden activation of the [CLS] token is then fed through a linear classifier to predict one of the 13 interaction types or NEGATIVE. Error gradients are fed back from the linear classification layer, through the entire BERT network via the [CLS] token's top hidden layer. In this way, the [CLS] top-layer activation is adjusted to express the contextual relationship between the special @CHEM# and @GENE# tokens, which the linear classifier can then exploit.

Lee and colleagues (1, Table 7) report excellent results on the 2017 ChemProt challenge (4), and, even though not directly comparable to the current DrugProt challenge, we chose this approach as a strong baseline. The system is implemented using the Huggingface transformer package, with PubMedBERT (8) trained on uncased full text and abstracts as the underlying transformer model (available here: https://huggingface.co/microsoft/BiomedNLP-PubMedBERT-base-uncased-abstract-fulltext). One departure from the description of the implementation in (1) is that we account for nested drug-protein entity pairs, e.g., "protein **tyrosine** phosphatase 1B" becomes "@GENE# **@CHEM#** #GENE@", and similarly for the reverse case. Lee and colleagues do not discuss such cases. Fig. 1 schematically illustrates our baseline architecture. We refer the reader to the original BERT paper (5) and the BioBERT paper (1) for more details.

C. Syntactic BiLSTM System: Syn-BiLSTM ("Run 1")

Preliminary analysis of the BioBERT reimplementation revealed dropping performance as the drug-protein entity pairs become more distant from one another. As others have shown (6,7), BERT transformers can represent linguistic structure at various layers within their multilayer structures, but such information is largely latent, and classifiers must be trained explicitly on hidden activations in the BERT network in order to accurately reproduce linguistic tasks such as grammatical parsing (7, Table 2). Instead, we explicitly parse each input sentence and derive a composite word vector for each syntactic token, feeding these word vectors and other parse-derived features into a bidirectional LSTM (BiLSTM) classification model (2) along the shortest path that the dependency parse connects the drug and protein entities using any of their constituent tokens. We call this system the "Syn-BiLSTM". The features emitted at each step in this path are:

Fig. 1. BioBERT Baseline System



- The composite (averaged) BERT WordPiece token activations corresponding to a full syntactic token.
- The dependency relation that the current dependency parse edge is labeled with.
- Whether or not the current token is the head or dependent of the current dependency relationship.

To obtain dependency parses, we use spaCy (9), with the ScispaCy (10) model `en_core_sci_lg`. We use the spaCy Transformers library (11) to wrap the Huggingface BERT model (also PubMedBERT, exactly as the baseline). To compute the shortest path, we encode each spacy parse as a networkx (12) `Graph` and run the built-in `shortest_path(.)` algorithm on all token pairs from each drug-protein entity pair. The shortest path of all token pairs is used to generate the features described above.

In this preliminary implementation, error gradients from BiLSTM training are not used to update the underlying BERT network, but we anticipate doing so in future implementations, most likely improving overall performance. Fig. 3 schematically illustrates our Syn-BiLSTM system.

D. Training

Both systems were trained using the Huggingface `transformers.Trainer` class, with a batch size of 16, dev-tuning evaluation runs every 1,000 batches, and a patience of 5 evaluation runs—i.e., if the performance does not improve after 5 evaluations, training is stopped and the best model checkpoint so far is retained. We use the `Adafactor` optimizer with `lr=None`, `scale_parameter=True`, `relative_step=True`, `warmup_init=True` and `weight_decay=0.0` (we did not find weight decay to be helpful). For both models we explored dropout values {0.1, 0.2, 0.3, 0.4, 0.5} and BiLSTMs of depths 1 and 2, and hidden layer sizes of {128, 256, 512}. The best-performing systems were chosen via 50/50 interpolation of F1 and classification accuracy on the held-out dev-test set. The best-performing BiLSTM has depth 1 and 256 hidden units and a dropout of 0.3 during training. The best BioBERT model had a dropout value of 0.5 during training.

III. Results and Analysis

We evaluate our systems in several ways. First, as simple classifiers, we evaluate the accuracy of predicting one of the 13 interaction types (AGONIST, INHIBITOR, etc.) or NEGATIVE (no interaction). We additionally report the binary accuracy of predicting the presence vs. the absence of a drug-protein interaction. We use the official DrugProt evaluation library (13) to compute precision, recall and F1 by interaction type, as well as by micro-averaged precision, recall and F1 overall. To illustrate the effect of syntactic parse information to relate distant phrases effectively, we give accuracy on the dev-test set binned by 10-character increments of drug-protein entity distances.

In all instances except recall on the blind test set, the Syn-BiLSTM outperforms the BioBERT model in aggregate metrics (Table II), and outperforms the DrugProt mean of precision and F1 on the test set. The Syn-BiLSTM also maintains its classification accuracy as the distances between

TABLE II. Acc and (Micro-averaged) P, R and F1

		Acc	Bin Acc	P	R	F1
BioBERT Baseline	Dev	82.8	84.3	66.4	60.7	63.4
	DevTest	78.0	79.1	69.8	43.0	53.2
	Test	N/A	N/A	64.3	60.2	62.2
Syn-BiLSTM	Dev	**84.3**	**86.0**	**69.5**	**62.0**	**65.5**
	DevTest	**82.2**	**83.8**	**73.5**	**57.4**	**64.5**
	Test	N/A	N/A	**67.5**	58.2	**62.5**
BC7 Mean	Test	N/A	N/A	64.3	**62.9**	62.0

Fig. 2. Classification Accuracy vs. Drug-Prot Distance

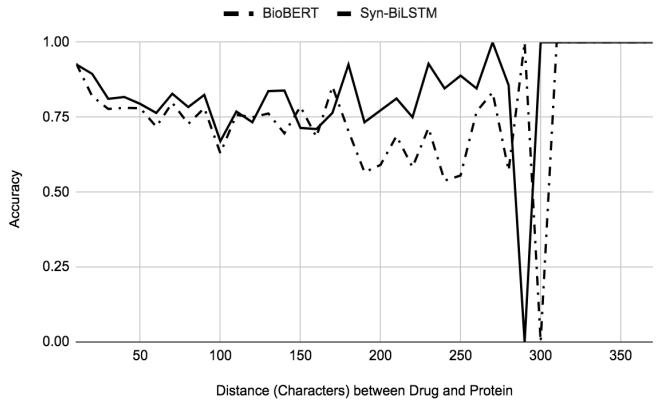
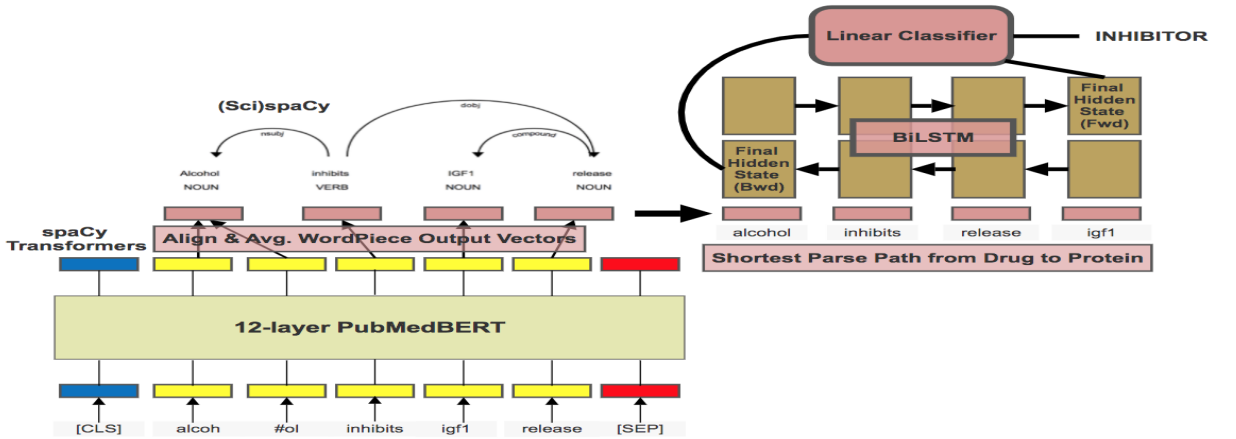


Fig. 3. Syn-BiLSTM System



the drug and protein grow (Fig. 2).

Table III presents the fine-grained results of our Syn-BiLSTM system on the test set. To better explain these results, in Fig. 4, we plot Syn-BiLSTM **DevTest** performance alongside training set size and **Dev F1**. As in the test results, low-data interaction types have zero predictions. As more training data is available, we see that the type-specific performance generally improves. AGONIST and ANTAGONIST labeling performance is very high with less than 6% training data each, likely due to the lexically signaled nature of these interaction types—viz., 'agoni*' and 'antagon*' are in 96% and 97%, resp., of the corresponding examples. Three interaction types—PRODUCT-OF, INDIRECT-DOWNREG. and INHIBITOR—have lower DevTest F1 than Dev F1, indicating possible overfitting. While a full error taxonomy is beyond the scope of this work, we note that, in particular, INDIRECT-DOWNREG. is often (understandably) conflated with INHIBITOR, and INHIBITOR often occurs in sentential contexts that do not disambiguate the interaction type—the interaction type can only be predicted in the context of the whole abstract.

IV. Discussion and Future Work

We have presented two main track DrugProt systems: our baseline, a PyTorch BioBERT reimplementation, and our Syn-BiLSTM system, which feeds BERT token encodings to a bidirectional LSTM in dependency parse order. The Syn-BiLSTM system, as we have shown, is much more resilient to long distances between drugs and proteins, and surpasses the BioBERT system in straight classification accuracy as well as precision and F1.

Finally, we note some obvious areas for improvement. First, the Syn-BiLSTM system would likely perform better if we update the underlying BERT model during training. Second, the use of ontologies, gazetteers and other knowledge sources—either as features or as rule-based components—could certainly help with model performance on known classes of drugs and proteins, especially on those types of interactions that do not occur frequently in training.

Finally, heuristically or otherwise combining whole-document information might help overcome false negative predictions where intra-sentential disambiguating information is lacking.

Fig. 4. Dev F1 and DevTest Performance by Type vs. Training Size

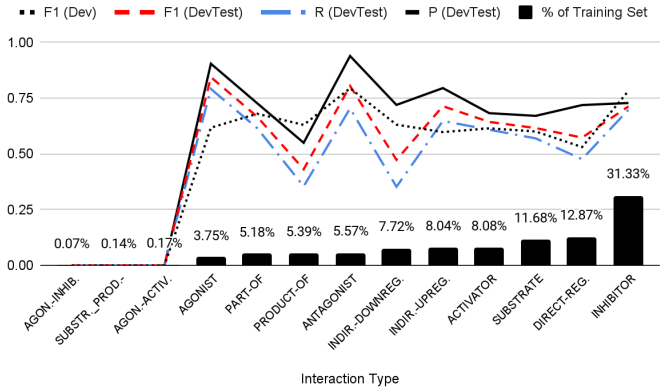


TABLE III. Syn-BiLSTM Interaction-specific P, R and F1 on Test Set

	P	R	F1
Activator	72.3	43.7	54.5
Agonist	75.9	65.3	70.2
Agonist-Inhib.	0	0	0
Antagonist	78.1	65.4	71.2
Direct-reg.	61.7	48.5	54.3
Indirect-downreg.	67.3	46.7	55.1
Indirect-upreg.	60.9	58.1	59.5
Inhibitor	73.5	73.5	73.4
Part-of	58.3	64.5	61.3
Product-of	60.0	60.2	60.1
Substrate	65.7	44.9	53.3
Substrate-product	0	0	0
Agonist-activator	0	0	0

References

1. Lee,J., Yoon,W., Kim,S., Kim,D., Kim,S., So.C.H. and Kang.,J. (2020) BioBERT: a pre-trained biomedical language representation model for biomedical text mining. In Wren,J. (ed.), *Bioinformatics*. Oxford University Press, Vol 36, Issue 4, pp.1234-1240.

2. Hochreiter,S. and Schmidhuber,J. (1997) Long Short-Term Memory. *Neural Comput*. Vol 9, Issue 8, pp. 1735–1780.

3. Miranda,A., Mehryary,F., Luoma,J., Pyysalo,S., Valencia,A. and Krallinger,M.. (2021) Overview of DrugProt BioCreative VII track: quality evaluation and large scale text mining of drug-gene/protein relations. In Proceedings of the seventh BioCreative challenge evaluation workshop.

4. Krallinger,M., et al. (2017) Overview of the BioCreative VI chemical-protein interaction track. In: Proceedings of the BioCreative VI Workshop, Bethesda, MD, USA, pp. 141–146.7.

5. Devlin,J., Ming-Wei,C., Lee,K. and Toutanova,K. (2019) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of NAACL: HLT, (Long and Short Papers). Volume 1, pp. 4171–4186.

6. Tenney,I., Xia,P., Chen,B., Wang,A., Poliak,A., McCoy,R.T., Kim,N., Van Durme,B., Bowman,S., Das,D. and Pavlick,E. (2019) What do you learn from context? Probing for sentence structure in contextualized word representations. In Proceedings of the International Conference on Learning Representations.

7. Jawahar,G., Sagot,B. and Seddah,D. (2019) What Does BERT Learn about the Structure of Language? In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, pp. 3651–3657.

8. Gu,Y., Tinn,R., Cheng,H., Lucas,M., Usuyama,N., Liu,X. Naumann,T., Gao,J. and Poon,H. (2020) Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing. arXiv:2007.15779.

9. Explosion (Accessed September 2021) spaCy 3. Retrieved from https://github.com/explosion/spacy

10. Neumann,M., King,D., Beltagy,I. and Ammar,W. (2019) ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing. In Proceedings of the 18th BioNLP Workshop and Shared Task. Florence, Italy, pp. 319–327.

11. Explosion (Accessed September 2021) spaCy Transformers. Retrieved from https://github.com/explosion/spacy-transformers.

12. NetworkX (Accessed September 2021). Retrieved from https://github.com/networkx/networkx.

13. BioCreative VII DrugProt Task Evaluation Software (Accessed September 2021). Retrieved from https://github.com/tonifuc3m/drugprot-evaluation-library