Dependencies for Drug-Prot Relation Extraction CLaC at BioCreative VII Track 1

Parsa Bagherzadeh and Sabine Bergler
CLaC Labs, Concordia University, Montreal, Canada
{p_bagher, bergler} @ cse.concordia.ca

***Abstract*—For the task of relation extraction between chemicals and genes we report on the potential of dependency relations. Entities of interest are explicitly marked in our input. All our runs outperform the competition mean** $(.61)$ **and our best run yields a F1 score of** $.67$**.**

***Keywords*—Multi-input RIM, modular model, knowledge sources**

I. Introduction

This paper describes the CLaC submission to BioCreative VII Track 1: Drug-Prot relation extraction (4). The Drug-Prot track involves finding possible relations between a set of pre-annotated chemicals and genes or gene products. In the case of a relationship between a chemical and a gene, the relation has to be classified into one of the 14 categories listed in Figure 1. Figure 2 illustrates gold standard annotations for an instance of the DIRECT-REGULATOR relation between the chemical *NCFP* and the gene *mGlu5*. Note that no relation should be reported for the pair *CPPHA* and *mGlu5*.

To address the task, we explicitly feed the two entities of interest along with the sentence as input to ClinicalBERT and use the CLS token in a modular model that leverages dependency relations.

None | PART-OF | ACTIVATOR | INHIBITOR
INDIRECT-DOWNREGULATOR | DIRECT-REGULATOR
INDIRECT-UPREGULATOR | ANTAGONIST
AGONIST-INHIBITOR | AGONIST-ACTIVATOR | AGONIST
SUBSTRATE | PRODUCT-OF | SUBSTRATE_PRODUCT-OF

Fig. 1. 14 relation types for Drug-Prot task

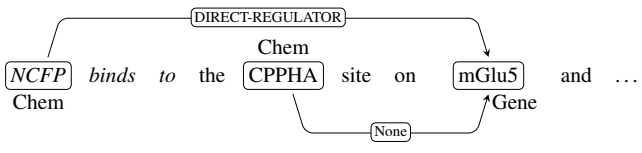

Fig. 2. An example of two chemical mentions, one gene mention, and an instance of a DIRECT-REGULATOR relation

II. System Description

A. Preprocessing

We process the abstracts using a GATE pipeline (9). The text is tokenized using ANNIE English tokenizer (the alternate version) and for sentence splitting the ANNIE sentence splitter is used. It should be noted that we merge all tokens in the span of a chemical into a single token.

B. Multi-input RIMs

Multi-input Recurrent Independent Mechanisms (mi-RIMs) is a modular architecture that comprises M recurrent modules (2). The modules can enter into competition mode by forcing only k of them to be active at each time step. In addition to competition, the modules are able to interact with one another. At each time step, the module attend to each others hidden states and update their hidden state accordingly.

C. Input configuration

a) Marking entities in input: Considering an input sentence S_i and a chemical-gene entity pair $(Chem_k, Gene_j)$, we construct an input for ClinicalBERT (1) that explicitly identifies the entities of interest as

[CLS] $Chem_k$ [SEP] $Gene_j$ [SEP] S_i

as exemplified in Figure 3.

D. Task specific representation learning

We use a mi-RIM with two recurrent modules R_1 and R_2. The module R_1 is a simple LSTM unit operating on ClinicalBERT embeddings and R_2 is a Graph-LSTM (5) that encodes dependency information using a bi-directional recurrent architecture. The forward pass encodes all of the dependencies from a dependency parse where the dependent follows the governor, and the backward pass encodes those dependencies, where the dependent precedes the governor in the input sequence (see Figure 4). At time step t the input to the recurrent module is the token at position t as well as the hidden states at all time steps corresponding to its governors.

We use the Stanford Parser (3) for extracting dependency relations.

E. Classification

We use the [CLS] token of ClinicalBERT as well as all hidden states of module R_1 to classify the input for its relation type. During inference time, we do not report the pairs for which no target relation is predicted.

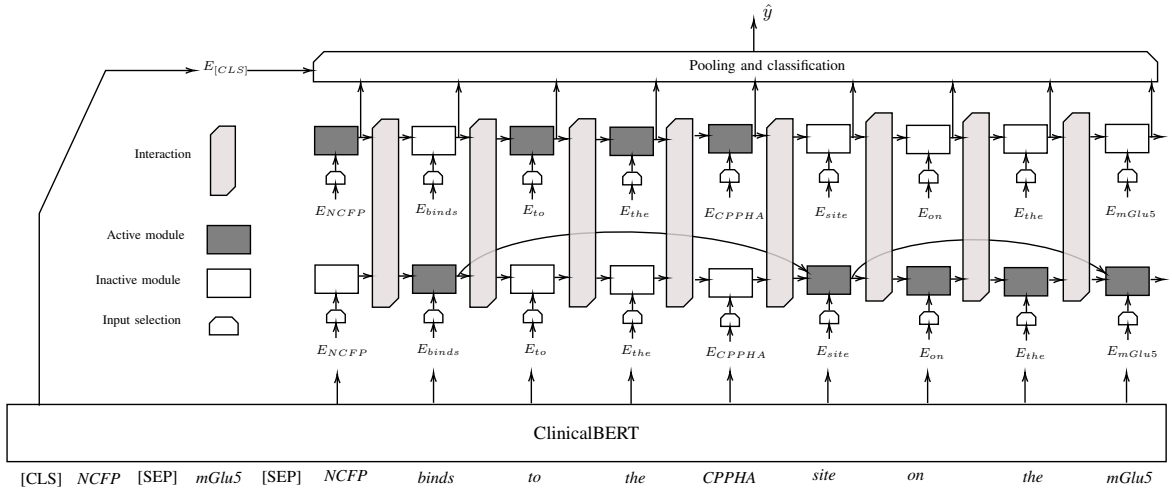

Fig. 3. A multi-input RIM with one LSTM on ClinicalBert embeddings (top) and a Graph-LSTM encoding dependency relations over the same embedding input (bottom) for a forward pass

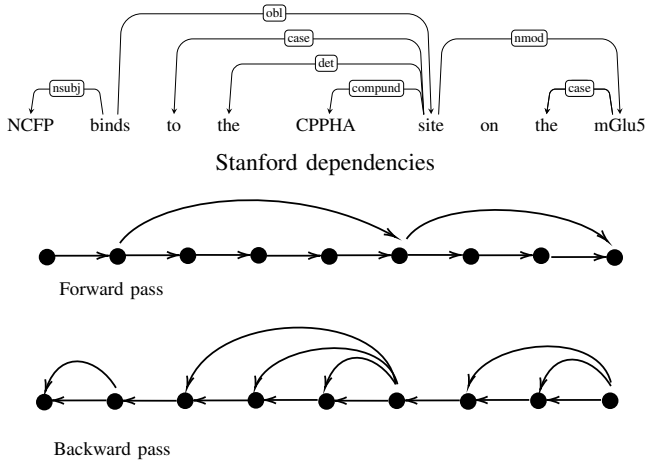

Fig. 4. Graph-LSTM model for encoding dependency relations

F. Training paradigm

Only a small fraction of chemical-gene pairs in a sentence actually realize one of the target relationships and the system should not predict any of the 13 target relation types for pairs that have no relations. We introduce the None label to specify all unrelated pairs.

Suppose $\mathcal{C}_i$ and $\mathcal{G}_i$ are the set of chemical and gene entities in sentence S_i, respectively. We calculate the cross product $\mathcal{C}_i \times \mathcal{G}_i$ for all chemical-gene pairs. Pairs for which the training data does not report a label are considered unrelated pairs (i.e. assigned our None label). Consider, for example, the set of chemicals $\mathcal{C} = \{NCFP, CPPHA\}$ and the set of genes $\mathcal{G} = \{mGlu5\}$ in Figure 2. $\mathcal{C} \times \mathcal{G} = \{(NCFP, mGlu5), (CPPHA, mGlu5)\}$ specifies all possible pairings where $(NCFP, mGlu5) = DIRECT-REGULATOR$ and $(CPPHA, mGlu5) = None$.

Since the None labels far outnumber other labels we have to address the imbalance issue. We randomly drop out (with a probability $p = 0.5$) those pairs whose label is None, i.e.we do not use them for calculating loss. All pairs with labels other than NONE are used for training.

We train the proposed system for 10 epochs using Adam optimizer (6) with $lr = .1e - 5$. The PyTorch library is used for implementations (7).

III. Results

A. Development phase

Figure I reports the performance of the proposed system on the development set provided by the organizers. We note that the size of the development set (750 abstracts) is significantly smaller than the training (3499 abstracts) and test sets (10750 abstracts). We report the results for two variants of the proposed model, ClinBERT (single LSTM on ClinicalBERT without dependencies) and +Dep (one LSTM on ClinicalBERT plus a Graph-LSTM encoding dependencies).

Adding the dependency module leads to consistent improvements across all classes for precision and recall.

TABLE I
Results on development data

	ClinB ($k=1$)			+Dep ($k=1$)			+Dep ($k=2$)		
Relation type	P	R	F1	P	R	F1	P	R	F1
ACTIV	.75	.54	.64	.80	.58	.67	.81	.62	.70
AGONIST	.68	.62	.65	.70	.65	.67	.71	.67	.68
AGONIST-INHIB	.00	.00	.00	1	.50	.66	1	.50	.66
ANTAGONIST	.72	.77	.74	.76	.78	.77	.77	.78	.78
DIRECT-REGU	.60	.60	.60	.65	.61	.63	.65	.63	.64
IND-DOWNREG	.67	.66	.66	.72	.68	.70	.73	.69	.70
IND-UPREG	.73	.62	.67	.76	.63	.69	.76	.65	.70
INHIBITOR	.74	.81	.77	.76	.82	.78	.77	.85	.80
PART-OF	.59	.56	.57	.65	.59	.61	.65	.62	.63
PROD-OF	.50	.54	.51	.54	.55	.55	.55	.57	.56
SUBS	.52	.68	.58	.58	.69	.63	.70	.65	.67
SUBS_PROD-OF	.00	.00	.00	1	.66	.80	1	.66	.80
AGONIST-ACTIV	.00	.00	.00	.80	.40	.53	.80	.40	.53
Mirco-average	.66	.68	.66	.70	.69	.69	.72	.71	.71

For the classes AGONIST-INHIBITOR, AGONIST-ACTIVATOR, and SUBSTRATE_PRODUCT-OF, adding the dependency module leads to significant changes in scores. This is due to the fact that these classes are rare. For instance, of 3761 annotated relations only 10 are of type AGONIST-ACTIVATOR and any correct prediction leads to a sizeable improvement in the score.

Overall, limiting activation to one active module does not lead to performance gain in this system configuration, neither on the development set nor the test set.

a) Error analysis:

Example 1 shows two instances of the chemical *rifampicin* ($\{T_3, T_4\}$) and one instance of the gene *nuclear pregnane X receptor* ($\{T_{11}\}$). Consequently, the system considers two candidate pairs (T_3, T_{11}) and (T_4, T_{11}). All three variants of the system classify both pairs as ACTIVATOR. Note that the trigger word *activates* suggest this classification. However, only the prediction for (T_4, T_{11}) is correct and the gold label for (T_3, T_{11}) is reported as AGONIST-ACTIVATOR. We consider the basis of this classification to lie outside the sample sentence that requires additional lexicalization or expertise.

Example 1:

In general, rifampicin^{T_3} can act on a pattern: rifampicin^{T_4} activates the nuclear pregnane X receptor$^{T_{11}}$

Example 2 on the other hand shows a correct classification due to the dependency module. The pair (T_7, T_{22}) belongs to the AGONIST-ACTIVATOR relation. Note that the two entities are far apart in the sentence. We observed that in such cases, the ClinicalBERT model fails to provide correct predictions and often predicts no relations (None). This confirms the observations made by studies such as (8) that BERT-like models often fail to capture long distance dependencies.

Example 2:

In conclusion, fenoterol-induced constitutive beta(2)-adrenoceptor$^{T_{22}}$ activity reduces muscarinic receptor agonist- and histamine-induced contractions of bovine tracheal smooth muscle, which can be reversed by the inverse agonist timolol^{T_7}.

Our experiment shows that explicit dependency information can overcome that tendency in ClinicalBERT. On the development data, adding the dependency module did not result in performance loss (exception: recall for SUBS).

A majority of sentences in the training data include several entities. Example 3 shows a sentence that includes two mentions of chemicals $\{T_1, T_{11}\}$ and two mentions of genes $\{T_{12}, T_{13}\}$. For this example, the set of all possible pairs is $\{(T_1, T_{12}), (T_1, T_{13}), (T_{11}, T_{12}), (T_{11}, T_{13})\}$, among which only (T_{11}, T_{12}) and (T_1, T_{13}) have ANTAGONIST relation and no relation (None) should be reported for the other two pairs. The two variants ClinB $(k = 1)$ and +Dep $(k = 1)$ classify all pairs as ANTAGONIST, however, +Dep $(k = 2)$ successfully classifies those pairs that have no relations. While not fully realized on the test set, this potential of dependencies to filter out unrelated pairs is interesting.

Example 3:

In the hot-plate test in mice, the antinociceptive action of the alpha 2-adrenoceptor$^{T_{12}}$ agonist, UK 14,304$^{T_{11}}$, was abolished by the alpha 2-adrenoceptor$^{T_{13}}$ antagonist, idazoxan^{T_1} ...

B. Evaluation phase

The official competition results are provided in Figure II. All of our submissions significantly outperform the competition mean micro-average. The competition results, however, are not commensurate with development results. As expected, the dependency relations improve precision, however, a significant drop in recall offsets the gains.

TABLE II
OFFICIAL COMPETITION RESULTS

	ClinB ($k = 1$)			+Dep ($k = 1$)			+Dep ($k = 2$)		
Relation type	P	R	F1	P	R	F1	P	R	F1
ACTIV	.75	.68	.71	.72	.71	.72	.72	.71	.72
AGONIST	.69	.68	.69	.77	.69	.73	.77	.69	.73
AGONIST-INHIB	.00	.00	.00	.50	.33	.40	.00	.00	.00
ANTAGONIST	.73	.84	.79	.72	.84	.77	.72	.84	.77
DIRECT-REGU	.58	.58	.58	.61	.55	.58	.61	.55	.58
IND-DOWNREG	.63	.75	.68	.73	.60	.66	.73	.60	.66
IND-UPREG	.68	.62	.65	.64	.54	.59	.64	.54	.59
INHIBITOR	.74	.81	.78	.78	.74	.76	.78	.74	.76
PART-OF	.50	.58	.54	.61	.43	.51	.61	.43	.51
PROD-OF	.44	.68	.53	.60	.57	.59	.60	.57	.59
SUBS	.51	.59	.54	.52	.49	.50	.52	.49	.50
SUBS_PROD-OF	.00	.00	.00	.00	.00	.00	.00	.00	.00
AGONIST-ACTIV	.00	.00	.00	.00	.00	.00	.00	.00	.00
Micro-average	.64	.70	.67	.68	.63	.66	.68	.63	.66
Competition Mean	.64	.62	.61						
Competition Std	.19	.24	.22						

IV. CONCLUSION

Injecting dependency information through one of two LSTMs in the multi-input RIM architecture for the task of relation extraction between chemicals and genes showed promise on the development data but failed to produce our best run in competition on a test dataset much bigger than the training dataset. Interestingly, injecting dependencies led to improvements in precision, but to a larger loss in recall. The baseline ClinicalBERT system performed, in contrast, almost identically on development and test sets.

REFERENCES

1. Alsentzer, E., et. al., (2019) Publicly available clinical BERT embeddings. In Proceedings of the 2nd Clinical Natural Language Processing Workshop.
2. Bagherzadeh, P., Bergler, S., (2021) Multi-input Recurrent Independent Mechanisms for leveraging knowledge sources: Case studies on sentiment analysis and health text mining. In Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures.
3. de Marneffe, M., MacCartney, B., Manning, C. D., (2006) Generating typed dependency parses from phrase structure parses, in Proccesings of LREC.
4. 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. Proceedings of the seventh BioCreative challenge evaluation workshop.
5. Peng, N., et. al, (2017) Cross-sentence n-ary relation extraction with Graph LSTMs, in Transactions of the Association for Computational Linguistics, Volume 5, Pages 101–115.
6. Kingma, D. P., Ba, J. L., (2015) Adam: A method for stochastic optimization, In Proceedings of the 3rd International Conference on Learning Representations, ICLR'15
7. Paszke, A., et al., (2017) Automatic differentiation in PyTorch, NIPS
8. Jawahar, G., et al. (2019) What Does BERT Learn about the Structure of Language?, In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics
9. Cunningham, H., et al. (2002) GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications, In Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)