Entity and Action extraction in Wet Lab Protocols

Bohan Zhang, HAN Han
Department of Computer Science and Engineering
The Ohio State University
{zhang.8856, han.1242}@osu.edu

Abstract

In this paper, we reproduce two models, maximum entropy and BiLSTM-CRF proposed by Kulkarni et al. (2018) for the entity and action extraction task in wet lab protocols. We use SciBert model with 3 3 additional linear layers to achieve the state-of-the-art results. We also use an adjusted Actor-Critic reinforcement algorithm mixed with traditional maximum-likelihood cross-entropy loss to replace the CRF layer for possible model extensions.

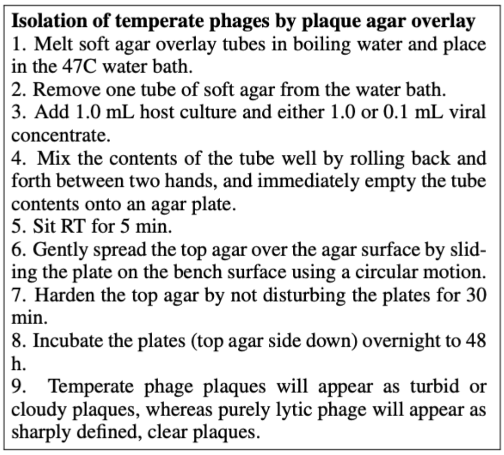
1 Introduction

Due to an increase in biological experiments in recent years, automation in wet laboratory procedures can be beneficial to eliminate human misunderstanding of the wet lab protocol. However, the majority of protocols are written in natural language. An annotated corpus of instructions for machine-reading in Wet Lab Protocols was published in 2018 (Kulkarni et al., 2018). This annotated corpus of wet lab protocols can enable further research on interpreting natural language instructions and enabling robotic automation, with practical applications in biology and life sciences.

One important task and procedure involved in wet lab protocols is the entity and action extraction. In our study, We reproduce two models, maximum entropy and BiLSTM-CRF, proposed in Kulkarni et al. (2018) but with several changes in feature utilization to have better results. Besides that, we use SciBert (Beltagy et al., 2019), a pre-trained multilayer bidirectional Transformer language model based on BERT but trained on a large scientific corpus to improve performance on downstream scientific NLP task. Adding three linear layers on the top of SciBert outperforms the past state-of-the-art achieved by a general framework for information extractions using Dynamic Span Graphs (Luan et al., 2019). We also use an adjusted Actor-Critic reinforcement algorithm mixed with traditional maximum-likelihood cross-entropy loss to replace the CRF layer for future model extensions. This learning algorithm outperforms BiLSTM-CRF model.

2 Wet Lab Protocols

Wet laboratories mainly work on biology and chemistry experiments. It was created by research groups around the world, adapting from the canonical source and published in the Materials and Method section at the end of a scientific article in biology and chemistry fields. An example of one set of wet lab protocols is shown in figure 1.



Isolation of temperate phages by plaque agar overlay
1. Melt soft agar overlay tubes in boiling water and place in the 47C water bath.
2. Remove one tube of soft agar from the water bath.
3. Add 1.0 mL host culture and either 1.0 or 0.1 mL viral concentrate.
4. Mix the contents of the tube well by rolling back and forth between two hands, and immediately empty the tube contents onto an agar plate.
5. Sit RT for 5 min.
6. Gently spread the top agar over the agar surface by sliding the plate on the bench surface using a circular motion.
7. Harden the top agar by not disturbing the plates for 30 min.
8. Incubate the plates (top agar side down) overnight to 48 h.
9. Temperate phage plaques will appear as turbid or cloudy plaques, whereas purely lytic phage will appear as sharply defined, clear plaques.

Figure 1: An example of wet lab protocols

Here We briefly introduce two general types of taggers that we are trying to extract. The first type is **Action**: In wet lab protocols, action is a short description of a task tying various entities meaningfully. For example, verbs like add, incubate, and remove are action words. The second is **Entity**: Kulkarni et al. (2018)

classified entities in the protocols into 17 tags. For example, `concentration` tags appear like $60\%10x, 10M, 1g/ml$, and `method` tags, which usually have longer average span than others, appear like *rolling back and forth between two hands*. The data is spilt into training, test, dev sets in the ratio of 6:2:2. Examples of action and entities are shown in figure 2.

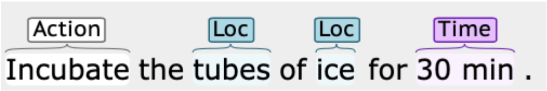


Figure 2: Examples of action and entity types.

3 Methodology

In this section, we will briefly introduce several models we apply in this task.

3.1 Maximum Entropy

In the maximum entropy model for action and entity extraction (Borthwick and Grishman, 1999), we used three types of features based on the current word and context words within a window of size 2, a similar setting with Kulkarni et al. (2018) with several changes:

- **Part of speech taggers** which are specically fine tuned for biomedical texts and generated by the GENIA POS Tagger (Tsuruoka and Tsujii, 2005)

- **Lexical features** which include unigrams, bigrams as well as their lemmas and synonyms from WordNet (Miller, 1995). Instead of arbitrarily plugging in all synonyms returned by WordNet, we only select three closest synonyms and pad if less than three.

- **Dependency parse features** which include dependent and governor words and the corresponding dependency type to capture syntactic information. We used the latest Stanford dependency parser released in 2019 originated from Chen and Manning (2014)

3.2 BiLSTM+CRF

The general state-of-the-art model for lots of sequence labeling tasks is Bidirectional LSTM with a Conditional Random Fields (CRF) layer. (Ma and Hovy, 2016). In our work, we initialized input embeddings using 300-dimension Glove word vectors (Pennington et al., 2014) instead of 200-dimension pre-trained vectors of PubMed and PMC biomedical texts. We found that 300-d Glove vectors made the model converge faster than 200-d PubMed. It's plausible since 300-d provides rich information and all these vectors will be tuned over iterations no matter how they are initialized. For words unseen in the pre-trained vocabulary, we randomly initialized using a normal distribution of mean 0 and variance 1.

An architecture graph of the BiLSTM-CRF model applied to the general named entity recognition task is shown in figure 3.

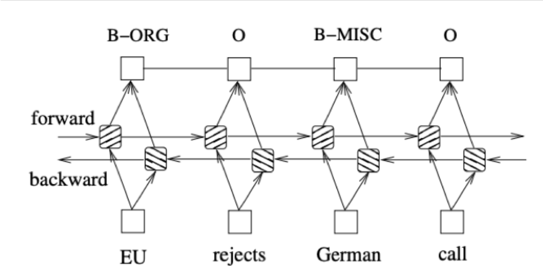


Figure 3: BiLSTM-CRF Model

3.3 SciBert+Three FFNN Layers

Pre-trained models have been sweeping across the entire NLP community in the past few years and Bert is one of the most popular models among all of them. As proposed in (Yu et al., 2019), Bert is still not good at cross-domain tasks as it is trained on news articles and Wikipedia data. Also, even past work provided several feasible strategies, like preceding multi-task fine-tuning (Sun et al., 2019), to fine-tune Bert for cross-domain usage, they still required considerable data sets that were consistent with those trained for Bert and expensive computing power.

In order to leverage the performance boosted by pre-trained models in the scientific domain, which is the domain of wet-lab protocols, we use SciBert (Beltagy et al., 2019), a pre-trained model on a large multi-domain corpus of scientific publications. They claimed to significantly outperform Bert in several tasks like sequence tagging, sentence classification and dependency parsing in the scientific domain, so we believe it can help on the entity extraction task in a similar domain.

We add 3 fully connected feedforward neural networks (FFNNs) on the top of SciBert architecture. The dimension of each layer is 768 (default output dimension of Bert family), 400 and 100. The non-linear function applied between each layer is the

Tanh function.

For mismatched tokens generated because of the different tokenizations between human annotation and SciBert, we map the first token of a word tokenized by SciBert to the original word and use the SciBert output embeddings of that token as the representation of the original word. This method is advised by huggingface, who released the current version of SciBert.

3.4 Adjusted Actor Critic Learning Mixed with Supervised Learning

Considering CRF is highly computationally complex at the training time and hard to extend with new data and new architectures, we try to change the CRF with another neural layer. Past work has proven that reinforcement learning (RL) algorithms can outperform BiLSTM-CRF and moderately offset the RNNs exposure bias (Najafi et al., 2018). We decided to apply an Adjusted Actor-Critic (AAC) training algorithm (Najafi et al., 2018) to replace the CRF layer and the algorithm they provided is shown in Algorithm 1.

There are mainly three parts in the AAC training:

- **Sequence-level Credits** which show how good a generated sequence is at each time step. It is defined as follow: At time step t, $G_t = \sum_{i=0}^{n-1} [r_{t+i}] + V_{\theta'}(t+n)$. n allows us to control bias-variance trade-off, with a large n resulting in less bias but higher variance. Reward r_t is +1 if the generated token $\tilde{y}_t$ is the same as the gold token y_t, and as 0 otherwise. Critic network V_θ is explained next.

- **Critic Network** We use three linear layers connected by leaky-ReLU activation functions (Nair and Hinton, 2010) between each hidden layer as our critic.

- **Adjusted Training** Clipping δ_t to zero by defining the adjusted advantage $a\delta_t$ as $adjust(y_t, \tilde{y}_t, \delta_t) \times \delta_t$ where:

$$adjust(y_t, \tilde{y}_t, \delta_t) = \begin{cases} 0 \text{ if } \tilde{y}_t = y_t \text{ \& } \delta_t < 0 \\ 0 \text{ if } \tilde{y}_t \neq y_t \text{ \& } \delta_t > 0 \\ 1 \quad\quad otherwise \end{cases}$$

Algorithm 1 Adjusted Actor-Critic Training

```
Input: Source X, target Y, and n as hyperparameter
Greedy decode X using  to get:
    • the output sequence Y = $(\tilde{y}_1, ..., \tilde{y}_l)$
    • the output sequence D = $(d_1, ..., d_l)$
    • context vectors C = $(c_1, ..., c_l)$
For each output target position t:
    • $r_t$=1 if $\tilde{y}_t = y_t$, 0 otherwise
    • $V_{\theta'}(t) = CriticNetwork(d_t, c_t, \theta')$
$\text{loss}_\theta = 0; \text{loss}_{\theta'} = 0$
For each output target position t:
    • $G_t = \sum_{i=0}^{n-1} [r_{t+i}] + V_{\theta'}(t+n)$
    • $\delta_t = G_t - V_{\theta'}(t)$
    • $a\delta_t = \text{adjust}(y_t, \hat{y}_t, \delta_t) \times \delta_t$
    • $\text{loss}_\theta = \text{loss}_\theta - a\delta_t \ln p_\theta (\hat{y}_t | X, \hat{y}_{t'<t})$
    • $\text{loss}_{\theta'} = \text{loss}_{\theta'} + \delta_t \times \delta_t$
Backpropagate through $loss_\theta$ to update $\theta$
Perform a gradient step along loss $\theta'$ to update $\theta'$
```

For a long time, RL algorithms are known to be hard to train for NLP tasks. Lots of tricks have been applied to address this problem. A mixed loss of RL loss and maximum-likelihood cross-entropy loss in supervised learning has shown effective in tasks like summarization and machine translation (Paulus et al., 2017, Wu et al., 2016). The mixed loss can provide a reasonable direction for RL to start in the first several iterations and lead RL to converge quicker. The loss is defined as shown in the following equation:

$$L_{mixed} = \gamma L_{rl} + (1-\gamma) L_{ml} \quad (1)$$

where γ is a scaling factor accounting for the difference in magnitude between L_{rl} and L_{ml}.

For all neural models, we used Adadelta (Zeiler, 2012) optimization without batch data and selected models with the best micro-avg F1 on the dev set in 40 epochs and report final numbers on the test set.

4 Results

The F1 scores of entities and action compared across the various models are shown in Table 1. The past state-of-the-art of micro-F1 was 79.5,

Entity/ Action	MaxEnt	BiLSTM+ CRF	ACC+Mixed Loss($\gamma = 0.9$)	SciBert+ 3FFNN
Action	83.97	87.19	85.03	**88.94**
Amount	78.89	78.93	80.88	**90.22**
Conc.	71.06	67.91	**82.11**	58.28
Device	52.21	53.59	51.65	**73.40**
Gen.-Measure	**30.05**	28.72	07.14	22.22
Location	69.45	70.88	69.36	**76.77**
Meas.-Type	**44.89**	30.77	21.05	16.67
Mention	36.13	10.26	**52.31**	08.11
Method	40.22	41.92	38.10	**56.25**
Modifier	52.13	57.46	50.25	**62.07**
Numerical	**66.32**	49.06	54.55	58.54
Reagent	75.24	79.97	81.89	**85.62**
Seal	**61.24**	44.44	16.67	14.29
Size	39.40	21.49	**61.54**	46.15
Speed	86.94	88.15	94.44	**98.70**
Temperature	83.86	77.60	84.85	**85.54**
Time	88.12	85.82	**93.71**	89.54
pH	57.39	**66.67**	61.90	61.90
micro-F1	73.56	76.04	76.26	**80.45**

Table 1: F1 scores of entities and action compared across the various models.

achieved by Dynamic Span Graphs (Luan et al., 2019) which is designed for general information extraction tasks. They didn't provide specific scores for each entity/action, so we couldn't compare in detail and we decided not to put them in an additional column to save space. The SciBert+3FFNN model reports the new state-of-the-art micro-F1 score of 80.45. The ACC training with mixed loss gets F1 score of 76.26, which is better than F1 score of 76.04 achieved by the BiLSTM-CRF model. The Maximum Entropy model achieves F1 score of 73.56, which is better than the numbers reported in Kulkarni et al. (2018) due to several changes we made to features and using more up-to-date dependency parser.

We can see that SciBert+3FFNN achieve the best F1 score for 9 entities/action of 18 in total. For entities with a large number of training cases like Action, Reagent, Modifier, and Location, SciBert significant outperforms other three models (average 5.99% better on F1 score). The ACC training gets the best numbers on 4 types of entities when λ is set to 0.9. Both two models outperform the micro-F1 score mentioned in (Kulkarni et al.,2018).

5 Conclusions

In conclusion, we reproduce two models, maximum entropy and BiLSTM-CRF, applied in Kulkarni et al., (2018) with several changes in feature utilization and report consistent results. Following that, we adapt the AAC training to outperform BiLSTM-CRF and SciBert models to achieve the state-of-art. For future work, we'd like to focus on the robustness of models. All these models fluctuate on some entities with few cases. This can be done by adding more data, but annotating more data can be extensive so we may want to try some distant learning or more cross-domain learning.

References

Andrew Borthwick and Ralph Grishman. 1999. A maximum entropy approach to named entity recognition. Ph. D. Thesis, Dept. of Computer Science, New York University.

Chaitanya Kulkarni, Wei Xu, Alan Ritter, Raghu Machiraju. 2018. An Annotated Corpus for Machine Reading of Instructions in Wet Lab Protocols (EMNLP).

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP).

Chi Sun, Xipeng Qiu, Yige Xu, Xuanjing Huang. 2019. How to Fine-Tune BERT for Text Classification? arXiv:1905.05583.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SCIBERT: A Pretrained Language Model for Scientific Text(IJCNLP).

Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (NAACL).

Jeffrey Pennington, Richard Socher, Christopher D. Manning. 2019. GloVe: Global Vectors for Word Representation (EMNLP).

Matthew D. Zeiler. 2012. ADADELTA: An Adaptive Learning Rate Method. arXiv preprints:1212.5701.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. a deep reinforced model for abstractive summarization (ICLR).

Saeed Najafi, Colin Cherry, and Grzegorz Kondrak. 2019. efficitnet sequence labeling with actor-critic training (ICLR).

Shanshan Yu, Jindian Su, Da Luo. 2019.

Improving BERT-based Text Classification with Auxiliary Sentence and Domain Knowledge (IEEE Access).

Vinod Nair, Geoffrey E. Hinton. 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. In ICML'10 Proceedings of the 27th International Conference on International Conference on Machine Learning.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL).

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, Hannaneh Hajishirzi. 2019. A General Framework for Information Extraction using Dynamic Span Graphs (NAACL).

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, ukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, Jeffrey Dean. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. arXiv:1609.08144.