Neural Multi-Task Learning for Adverse Drug Reaction Extraction

Feifan Liu
University of Massachusetts Medical School

Let us know how access to this document benefits you.

Follow this and additional works at: https://escholarship.umassmed.edu/qhs_pp

Part of the Artificial Intelligence and Robotics Commons, Chemicals and Drugs Commons, Health Information Technology Commons, Health Services Research Commons, and the Patient Safety Commons

Repository Citation

This material is brought to you by eScholarship@UMassChan. It has been accepted for inclusion in Population and Quantitative Health Sciences Publications by an authorized administrator of eScholarship@UMassChan. For more information, please contact Lisa.Palmer@umassmed.edu.
Neural Multi-Task Learning for Adverse Drug Reaction Extraction

Feifan Liu, PhD1, Xiaoyu Zheng, MS1, Hong Yu, PhD2, Jennifer Tjia, MD1
1University of Massachusetts Medical School, Worcester, MA, USA; 2University of Massachusetts Lowell, Lowell, MA, USA

Abstract
A reliable and searchable knowledge database of adverse drug reactions (ADRs) is highly important and valuable for improving patient safety at the point of care. In this paper, we proposed a neural multi-task learning system, NeuroADR, to extract ADRs as well as relevant modifiers from free-text drug labels. Specifically, the NeuroADR system exploited a hierarchical multi-task learning (HMTL) framework to perform named entity recognition (NER) and relation extraction (RE) jointly, where interactions among the learned deep encoder representations from different subtasks are explored. Different from the conventional HMTL approach, NeuroADR adopted a novel task decomposition strategy to generate auxiliary subtasks for more inter-task interactions and integrated a new label encoding schema for better handling discontinuous entities. Experimental results demonstrate the effectiveness of the proposed system.

Introduction
Drug labels are intended to provide health care professionals with clear and concise prescribing information that will enhance patient safety at the point of care 1. That important information is in an unstructured format that greatly limits its potential for real-life clinical practices, thus automatic extraction of ADRs associated with relevant properties from narrative drug labels has drawn increasing attention in pharmacovigilance community 2, natural language processing community 3 and the government 4. In 2017, the U.S. Food and Drug Administration (FDA) and the U.S. National Library of Medicine (NLM) jointly organized a shared task entitled “Adverse Drug Reaction Extraction from Drug Labels” at the Text Analysis Conference (TAC-ADR), which further advanced text mining techniques for ADRs extraction from drug labels 5. Our study focuses on the named entity recognition (NER) task, i.e. extracting ADRs and related concept modifiers (Severity, Factors, DrugClass, Negation, Animal), and relation extraction (RE) task, i.e. identifying the relations (Negated, Hypothetical, Effect) between ADRs and related concept modifiers. For example, in the text “Grade 3 cutaneous reactions”, “Grade 3” is a Severity, “cutaneous reactions” is an ADR, and there is an “Effect” relation between them.

Most of the existing systems for ADRs extraction exploited deep learning approaches which have shown promising results in many natural language processing (NLP) tasks 6. For instance, Saldana explored convolutional neural networks (CNN) for detecting ADR relevant sentences 7 and Alimova et al. utilized interactive attention neural network (IAN) to detect ADRs from biomedical texts 8. To effectively train deep neural networks, however, it usually needs millions of labeled samples which are often prohibitively expensive to get in many real-life applications 9. To address this challenge, semi-supervised methods based on co-training 10 and neural network pre-training 11 have been proposed respectively for extracting adverse drug reaction mentions from tweets. A popular alternative solution is Multi-Task Learning (MTL) 12, which has been widely applied and led to successes across all applications of machine learning, including speech recognition 13, NLP 14, computer vision 15. MTL has also been applied to ADR extraction from social media texts 16,17.

Several prior works have developed hierarchical MTL 18-20, which integrates supervised feedback from each task at different levels of a specific hierarchy, achieving better system performance than traditional MTL approaches. Hierarchical MTL can be seen as a seamless way to combine multi-task and cascaded learning which is especially helpful for NLP tasks with low-level tasks feeding into high-level ones 18. More recently, Sanh et al. further demonstrated the effectiveness of a hierarchically supervised multi-task learning on four related semantic tasks 21 without complex regularization schemes. Although conventional MTL approaches through shared components have been successfully applied in biomedical domain 22,23, exploring the emerging hierarchical MTL in biomedical applications is still an untapped but promising area. There lacks an understanding of how effective the hierarchical MTL works and what adaptations are needed to increase its potential without compromising the system generalizability in the biomedical domain. In this work, we proposed a new hierarchical MTL system, NeuroADR, for efficient ADR extraction from narrative drug labels. Unlike the top performed system at TAC-ADR 2017, NeuroADR is end-to-end trainable and was able to achieve comparable performance without relying on any handcrafted heuristic rules.
Our contributions are the following: (1) We propose a new hierarchical MTL architecture combining different subtasks related to ADR extraction, a first study to evaluate the state-of-the-art hierarchical MTL in the biomedical domain. (2) The new NeuroADR model exploits task decomposition and coupling strategies allowing more inductive transfer connections among different tasks at different hierarchy levels for optimal learning capacity. (3) A new label encoding schema was proposed to better handle discontinuous entities. (4) We evaluate and give insights on the impact of each adaptation strategy.

**Methods**

In this section, we describe our NeuroADR architecture regarding different tasks and components involved in the hierarchical multitask learning process. The main idea is that low-level tasks (e.g. NER tasks) are supervised at the lower level of the hierarchy while keeping more complex tasks (e.g. RE task) at deeper layers, and the supervisions from low-level tasks can provide feedback to deeper-level supervisions. Note that NER subtasks of different levels of complexity could be put at different levels of the hierarchy so that simple subtasks (e.g. NER-ADR-C) can provide feedback to more complex subtasks (e.g. NER-ADR-D) as described below. The architecture of our model is shown in Figure 1, where red lines indicate the decoding process for each task and blue lines for hidden representation flowing interaction among tasks.

**Words Embeddings**

As shown in Figure 1, we explore three word embeddings: (1) the state-of-the-art BERT (Bidirectional Encoder Representations from Transformers) encoder, which has been shown to outperform ELMo on various NLP tasks. (2) Domain Embedding: a customized word embedding trained through skip-gram setting using all PubMed open access articles, 99,700 EHR notes, and English Wikipedia articles in 2015. This embedding contains 3 billion tokens and the dimension is 200. (3) CharCNN: the convolutional neural network (CNN) based Character-level word embeddings. Therefore, for any word $w_i$ in a given input sentence ($s = w_1, w_2, \ldots, w_n$), the embedding layer concatenates three embeddings as its final embeddings ($e_{w_i}$).

**Task Decomposition**

As mentioned earlier, we focus on NER and RE tasks in the context of ADR extraction. Motivated by Liebel et al. who shows that adding auxiliary tasks as additional regularization can boost the performance of multi-task learning, we decompose original NER tasks into three tasks in the NeuroADR architecture (Figure 1): (1) **NER-ADR-C** for recognizing ADR continuous named entities; (2) **NER-ADR-D** for handling both continuous and discontinuous ADR named entities with a new encoding described later; (3) **NER-Modifier** for recognizing 5 types of modifier entities (Severity, Factors, DrugClass, Negation, Animal). The rationale behind this is that Modifier entities may prefer different hidden representations from ADR entities, and representations taking into account discontinuous entities may differ from the ones not. Through decomposition, those less complex but related subtasks fit well in the multi-task setting. The encoder for each task takes the word embeddings concatenated with the encoder outputs from the preceding tasks in the hierarchy (denoted by blue arrows in Figure 1) through multi-layer BiLSTM and derive an encoded sequence ($e_{NEX}$) into the final conditional random fields (CRF) layer for inferring the NER output.
Task Coupling

In the TAC-ADR task setting, Modifier concepts in drug labels are annotated as ground truth only if they are related to at least one ADR. Therefore, performing tasks to blindly recognize all the ADRs and Modifiers is prone to more false positives. The idea of task coupling is to do the modifier entity recognition and relation extraction at one shot. It is implemented as a typical sequence labeling task, where labels combining modifier type with relation type are predicted. What is special for this step is that each training example is specific to one target ADR, so only modifiers related to that ADR according to the ground truth get the entity labels and others will be labeled as "O" (outside). Similar to Xu et al. 29, we couple modifier NER task with the RE task (“RE-Coupling” in Figure 1) where the RE-Coupling module is trained to simultaneously extract mentions of modifier concepts that are associated with a given ADR entity and classify the relation type between the recognized modifier and the given ADR. To this end, if there is more than one ADR in a sentence, we generate a separate training instance for each ADR entity in the sentence. Similar to other tasks, the encoder of the RE-Coupling task takes the word embeddings and other encoder outputs provided by other tasks in the hierarchy and derives an encoded sequence (ERE-Coupling) into the CRF layer. We used traditional BIOUL (Begin, Inside, Outside, Unit, Last) tags together with modifier type and relation type as labels for modifier entities so that modifier entity and associated relations can be inferred simultaneously. For example, in “no vomiting”, the label for “no” would be “U-Negation_Negated”, indicating there is a “Negation_Negated” relation between “no” and the target ADR “vomiting”.

New Label Encoding for Discontinuous Entities

Traditional BIOUL labels for NER can only handle entities with continuous sequences of words. However, there are over 7% discontinuous entity mentions in the TAC-ADR data. To solve disjoint concepts, {B, I, O, HB, HI, DB, DI} was proposed 30 where HB/Hi refers to words that are shared by multiple concepts (head components) and DB/DI refers to words that belong to discontinuous concepts but not shared (discontinuous components). However, it is challenging and confusing in the decoding process to determine whether to combine each pair of identified HB/Hi and DB/DI or not. In this study, we propose a new label encoding to handle discontinuous entities by adding Head (H) to indicate shared headwords, Discontinuous label (D) for discontinuous components and Tail label (T) for shared tail words, leading to the encoding schema of (B, I, O, U, L, H, T, D) combined with the entity types. Figure 2 shows an example of the new encoding BIOUL-HTD. With the new encoding, it is clear that we shouldn’t connect the shared “changes” with discontinuous “a-” due to Head and Tail labels while it is ambiguous for the previous BIOUHD encoding.

![Figure 2. Example of a New Label Encoding. Entities in this example are: (a) electroretinographic changes; (b) ERG changes; (c) a-wave amplitude decrease; (d) b-wave amplitude decrease.](image)

Experiments

Dataset and Evaluation Metrics

The training data from TAC-ADR 2017 consists of manually annotated 101 drug labels. It contains 15,722 mentions and 3228 relations. The official test set contains 99 drug labels with 13735 mentions and 2039 relations annotated. The official evaluation script is provided to calculate the primary metrics of the micro-averaged F1 score with an exact match. Due to space limits, a more detailed data description is referred to the TAC-ADR task overview 5.
We implemented the HMTL model proposed in \(^{21}\) on our tasks using their standard NER and RE setup. For a fair comparison, we use our new label encoding so that discontinuous entities can also be handled. We also separately trained NER modules (NER-ADR-C, NER-ADR-D, and NER-Modifier) and the RE module (RE-Coupling) in the NeuroADR architecture to show the pipeline (sequentially perform NER and RE, with the output from the former as the input of the latter) results as shown in Table 1. We can see that NeuroADR outperforms both the \(^{21}\) system and pipeline system across different metrics, leading to the best F1 of 81.53% for NER and 38.58% for RE.

**Table 1.** Overall performance of NeuroADR.

<table>
<thead>
<tr>
<th>Model</th>
<th>NER Precision</th>
<th>NER Recall</th>
<th>NER F1</th>
<th>RE Precision</th>
<th>RE Recall</th>
<th>RE F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Sanh,2018)</td>
<td>78.84</td>
<td>78.28</td>
<td>78.56</td>
<td>44.54</td>
<td>28.87</td>
<td>35.03</td>
</tr>
<tr>
<td>Pipeline Approach</td>
<td>81.48</td>
<td>80.06</td>
<td>80.77</td>
<td>35.41</td>
<td>36.75</td>
<td>36.07</td>
</tr>
<tr>
<td>NeuroADR</td>
<td>82.45</td>
<td>80.63</td>
<td>81.53</td>
<td>42.05</td>
<td>35.64</td>
<td>38.58</td>
</tr>
</tbody>
</table>

Ablation Study of NeuroADR Performance

To better understand the effects of different adaptations proposed in NeuroADR, we conducted an ablation study as shown in Table 2. We ran the experiments in the following setting, i.e. starting with the NeuroADR architecture, we made only one change at a time. (1) BERT vs. ELMo (row A): replacing BERT with ELMo \(^{31}\); (2) Domain Embedding vs. Glove \(^{32}\) (row B): replacing domain embedding with Glove; (3) BIOUL-HTD vs. BIO-HD (row C): replace the proposed new encoding with the previous one in \(^{30}\); (4) Removing auxiliary task NER-Modifier (row D); (5) Merge NER-ADR-C with NER-Modifier into a single NER task (row E); (6) Replacing RE-Coupling with the RE module \(^{21}\).

We can see all the strategies we’ve exploited in NeuroADR contribute to the system’s performance positively, especially for adopting BERT (row A), task decomposition (row D and E), and the new encoding (row C). Domain embedding (row B) and task coupling (row F) also provide help in boosting the system’s performance.

**Table 2.** Ablation study of NeuroADR performance.

<table>
<thead>
<tr>
<th>Model</th>
<th>NER Precision</th>
<th>NER Recall</th>
<th>NER F1</th>
<th>RE Precision</th>
<th>RE Recall</th>
<th>RE F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>NeuroADR</td>
<td>82.45</td>
<td>80.63</td>
<td>81.53</td>
<td>42.05</td>
<td>35.64</td>
<td>38.58</td>
</tr>
<tr>
<td>Row A</td>
<td>82.43</td>
<td>77.00</td>
<td>79.62</td>
<td>38.15</td>
<td>27.99</td>
<td>32.29</td>
</tr>
<tr>
<td>Row B</td>
<td>80.87</td>
<td>81.40</td>
<td>81.14</td>
<td>38.74</td>
<td>37.86</td>
<td>38.30</td>
</tr>
<tr>
<td>Row C</td>
<td>82.82</td>
<td>79.18</td>
<td>80.96</td>
<td>41.14</td>
<td>34.69</td>
<td>37.64</td>
</tr>
<tr>
<td>Row D</td>
<td>81.83</td>
<td>79.69</td>
<td>80.75</td>
<td>40.60</td>
<td>32.43</td>
<td>36.06</td>
</tr>
<tr>
<td>Row E</td>
<td>83.28</td>
<td>78.33</td>
<td>80.73</td>
<td>40.90</td>
<td>30.13</td>
<td>34.70</td>
</tr>
<tr>
<td>Row F</td>
<td>82.36</td>
<td>80.14</td>
<td>81.24</td>
<td><strong>46.81</strong></td>
<td>32.01</td>
<td>38.02</td>
</tr>
</tbody>
</table>

Performance of NeuroADR per NER and RE Type

The statistics and performance of each NER and RE type are shown in Table 3. In the training set, the ADR mentions accounts for over 85%, which led to much better performance (F1 of 84.17%) compared with the modifier mentions. The performance for the Drug Class is pretty low (F1 of 40.7%), which is partially due to the complexities in their mention expressions (e.g. capitalized abbreviations tend to be recognized as Drug Class). It is worth mentioning that the model achieved the highest precision of 86.76% for the Animal mention, with F1 of 76.26%. As for RE, the performance on three types of relations is relatively low, yielding the F1 of 30%-44%. The Effect relation shows the worst performance, although there are a good amount of training examples available. In addition to the challenges posed by the language variations and ambiguities, there are some errors in the ground truth. For example, in the drug label of ADRMPAS, the sentence “Adempas may cause fetal harm” appears three times, but in the ground truth, the term “may” and “fetal harm” are only labeled once, which will adversely affect the performance for both NER and RE tasks.
Table 3. Performance on the testing data per type with counts in both training and testing.

<table>
<thead>
<tr>
<th>NER</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
<th># in Testing</th>
<th># in Training</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADR</td>
<td>84.52</td>
<td>83.83</td>
<td>84.17</td>
<td>12,693</td>
<td>13,795</td>
</tr>
<tr>
<td>Negation</td>
<td>72.93</td>
<td>56.64</td>
<td>63.76</td>
<td>173</td>
<td>98</td>
</tr>
<tr>
<td>Severity</td>
<td>65.14</td>
<td>57.45</td>
<td>61.05</td>
<td>947</td>
<td>934</td>
</tr>
<tr>
<td>Animal</td>
<td>86.76</td>
<td>68.02</td>
<td>76.26</td>
<td>86</td>
<td>44</td>
</tr>
<tr>
<td>Factor</td>
<td>69.73</td>
<td>70.26</td>
<td>69.99</td>
<td>562</td>
<td>602</td>
</tr>
<tr>
<td>Drug Class</td>
<td>50.93</td>
<td>33.89</td>
<td>40.70</td>
<td>164</td>
<td>249</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Relation</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
<th># in Testing</th>
<th># in Training</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negated</td>
<td>45.41</td>
<td>33.03</td>
<td>38.24</td>
<td>288</td>
<td>163</td>
</tr>
<tr>
<td>Hypothetical</td>
<td>45.81</td>
<td>42.48</td>
<td>44.08</td>
<td>1,486</td>
<td>1,611</td>
</tr>
<tr>
<td>Effect</td>
<td>35.64</td>
<td>27.67</td>
<td>30.89</td>
<td>1,181</td>
<td>1,454</td>
</tr>
</tbody>
</table>

Discussion and Conclusion

We proposed a hierarchical multi-task learning system, NeuroADR, for extracting ADR information from drug labels. We found that task decomposition and the new encoding can improve the performance in the multi-task setting. Our model’s performance on NER is slightly lower than the best result (81.53% vs. 82.48) in TAC-ADR 2017, however, our HMTL based model is end-to-end trainable, and not relying on any external linguistic tools, and handcrafted heuristic rules as the best system \(^2\) does. The proposed model has great potential if the heuristic-based pre-processing can be integrated. As shown in Table 3, our model achieved the best state-of-the-art performance on recognizing Factor (F1 of 69.99% vs. 68%) and Negation (F1 of 63.76 vs. 62.3%) entities \(^5\). The RE performance is low across all the teams where our model’s performance (38.58%) ranked the third.

Error analysis shows that the Drug Class entity performed worst with the F1 score of 40.70%, followed by Severity entity with the F1 score of 61.05%. Abbreviations introduced some confusion for the system when detecting Drug Class, e.g. "LABA" in "LABA, such vilanterol, ..." is a drug class but is missed in our model predictions, while "AED" in "... was added to the current AED therapy" was incorrectly picked up as Drug Class by the model. We do find some annotation errors, e.g. the ground truth annotated "another 5-HT3 receptor antagonist alone" as Drug Class, and our model only recognized "5-HT3 receptor antagonist" which makes more sense. Among relations, the "Effect" relation type obtained the lowest F1 score of 30.89% and most cases are false negatives, resulting in a low recall of 27.67%.

There are limitations to this study. First, we only evaluated our model on one dataset and further validation is needed. However, the dataset we used is relatively complex including a good amount of discontinuous entities and NeuroADR demonstrates competitive performance comparing the state-of-the-art methods. Second, we made an empirical ad-hoc design for the hierarchy structure to engaging different subtasks in an HMTL setting, which may have limited the potential of mutual benefits of multiple learning tasks. We will explore a more systematic approach, e.g. incorporating a similar strategy from fully adaptive feature sharing \(^3\), so that the decomposed subtasks can be dynamically learned.

References


