Combining rule-based and neural network systems for extracting adverse reactions from drug labels

Anne Cocos and Aaron J. Masino
The Children's Hospital of Philadelphia
acocos@seas.upenn.edu masinoa@email.chop.edu

Abstract

This report summarizes the system submitted by The Children's Hospital of Philadelphia (CHOP) to the TAC 2017 Adverse Drug Reaction Extraction from Drug Labels track. Our system combines a rule-based table extraction module and a recurrent neural network in a pipelined process to extract adverse drug reactions from drug label text. Identified reactions are then normalized against the MedDRA® preferred terms list based on word embedding similarity. Our system identified reactions in the test set with 47.99 macro-F1 (Task 3), and correctly normalized terms with 57.27 macro-F1 (Task 4).

1 Introduction

Adverse Drug Reactions (ADRs) are undesired drug effects that can have significant clinical and economic costs. Pharmacovigilance, or post-market drug safety surveillance, identifies adverse drug reactions that occur after a drug's release. Most pharmacovigilance currently relies on passive spontaneous reporting system (SRS) databases such as the Food and Drug Administration (FDA) Adverse Event Reporting System (FAERS). Such passive reporting can be limited by delayed reports and under-reporting (Hakkarainen et al., 2012; Sultana et al., 2013; Ahmad, 2003; Li et al., 2014). For this reason, the FDA has considerable interest in automatically extracting mentions of adverse reactions from drug labels. The National Institute of Standards and Technology (NIST) established Adverse Drug Reaction (ADR) Extraction from Drug Labels as a track in its 2017 Text Analytics Conference (TAC) to address the issue. The Children's Hospital of Philadelphia submitted a system to address Task 3 (ADR extraction) and Task 4 (Normalization).

NIST provides TAC participants with labeled and unlabeled drug labels in XML format for training and evaluation. The labels consist of both free text and tabular data. A quick glance at three randomly-selected drug labels from the training data indicates that all three contain tables, and 4 of the 4 total tables contain adverse drug reactions in one of the columns. For this reason, the CHOP team decided to treat tabular and non-tabular data separately in its system.

The CHOP system combines rule-based table extraction with a bi-directional long short-term memory (BLSTM) recurrent neural network (RNN) (Hochreiter and Schmidhuber, 1997; Schuster and Paliwal, 1997; Mesnil et al., 2013) to identify positive ADR mentions in drug labels. The system then normalizes the extracted reaction strings against the MedDRA®[1] (Brown et al., 1999) dictionary using a simple word embedding-based approach.

In this workshop notebook paper, we describe the CHOP systems for ADR extraction and normalization, and present initial results.

2 Data Description

The dataset used for the tasks of ADR extraction and normalization consists of 101 annotated drug labels for training, and 2208 unlabeled labels of which 99 were used for testing. The 101 training labels contain a total of 7038 true ADR mentions, and the 99 test labels contain a total of 6343 true ADR mentions. The number of true ADR mentions is the sum of unique ADR mentions per label, summed over labels. The test set contains a total of 5185 ADR normalizations, again counted in terms of unique normalizations per label, summed

[1] MedDRA® the Medical Dictionary for Regulatory Activities terminology is the international medical terminology developed under the auspices of the International Conference on Harmonisation of Technical Requirements for Registration of Pharmaceuticals for Human Use (ICH). MedDRA® trademark is owned by IFPMA on behalf of ICH.

over labels.

All labels are in XML format, and contain a `<Text>` element with one or more `<Section>` elements consisting of free drug label text.

3 Reaction Extraction Method

The Adverse Reaction extraction task (Task 3) of the TAC shared task requires systems to identify all positive adverse reaction mentions in the drug label text. The CHOP system can be summarized as a two-stage pipeline:

- First, a **rule-based table extraction module** identifies adverse reactions that are listed in table format, and removes the lines containing them from further consideration.
- Second, a **bi-directional recurrent neural network** labels tokens from the remaining lines as *part-of* or *not-part-of* a span of text containing a reaction mention.

In this section we describe the development of both pipeline stages.

3.1 Rule-based Extraction from Tables

Due to the prevalence of reactions embedded within tables in the drug labels, we employ a simple rule-based system to identify lines of drug label text that are likely to comprise tables, and to extract reactions from a single column of each identified table. An example table from the AMPYRA drug label is in Figure 1.

In our pipelined system, all lines identified as containing a reaction by the table extraction module are removed from further consideration, and the remainder passed to the second stage in the pipeline for consideration by the RNN. Our rule-based method assumes that all adverse reactions contained in tables are positive.

The table extraction module takes in a sequence of lines containing text from a drug label, and extracts reactions from likely tables based on the following rules:

1. Collect all subsets of contiguous lines that (a) begin with one or more whitespace characters, (b) contain a span of at least three whitespace characters (indicating column divisions), and (c) do not contain the words *Adverse* or *Reaction*. Call each subset a likely table.
2. Within each likely table, identify column offsets in terms of number of spaces from the start of each line.
3. For each column, count the number of strings within the column that match a lower-level or preferred term from MedDRA (Brown et al., 1999). The column with the most matches is predicted to contain adverse reactions.
4. Extract all strings within the chosen column as Adverse Reactions, unless (a) the string has more punctuation characters than letters, (b) starts with a punctuation character, or (c) the string contains two or more contiguous whitespace characters.

The table extraction module returns the set of identified reactions and their line numbers as output. We store the identified reactions to be reported by the system, and strip the lines containing them from the drug label text. The lines which were not predicted to contain reactions are passed to the next stage in the pipeline.

We evaluated the table extraction module's performance when used in isolation over the provided set of annotated drug labels (*train*), and the test set (*test*). The module correctly identified 1416 of the 7038 true ADRs in the *train* set with only 33 false positives. It also correctly identified 1471 of the 6343 true ADRs in the *test* set with 27 false positives. As expected for this type of rule-based model, the table extraction module achieves very high precision, but low recall due to ignoring all text outside of predicted tables. Its full scores are given in Table 1.

Metric	Train	Test
Micro-Precision	97.72	98.20
Micro-Recall	20.12	23.19
Micro-F1	33.37	37.52
Macro-Precision	85.78	92.37
Macro-Recall	21.91	22.39
Macro-F1	32.64	34.75

Table 1: Performance of the table extraction module when used in isolation over the training set of 101 annotated drug labels, and test set of 99 drug labels. The module identified 1416 of 7038 true ADR mentions in the training set, and 1471 of 6343 true ADR mentions in the test set.

Table 1: Adverse reactions with an incidence >=2% of AMPYRA treated MS patients, and more frequent with AMPYRA compared to placebo in controlled clinical trials

Adverse Reaction	Placebo(N=238)	AMPYRA10 mg twice daily(N=400)
Urinary tract infection	8%	12%
Insomnia	4%	9%
Dizziness	4%	7%
Headache	4%	7%
Nausea	3%	7%
Asthenia	4%	7%
Back pain	2%	5%
Balance disorder	1%	5%
Multiple sclerosis relapse	3%	4%
Paresthesia	3%	4%
Nasopharyngitis	2%	4%
Constipation	2%	3%
Dyspepsia	1%	2%
Pharyngolaryngeal pain	1%	2%

Figure 1: An example of a table contained within free text in the AMPYRA drug label. Lines containing tables are typically indented, and contain columns delineated by multiple whitespace characters. Our table extraction module extracts all strings in the first column (except the "Adverse Reaction" header) as identified ADRs.

3.2 Bidirectional LSTM for Extraction from Text

The second part of our pipelined system for ADR extraction is a bi-directional long short-term memory recurrent neural network (BLSTM) (Hochreiter and Schmidhuber, 1997; Schuster and Paliwal, 1997; Mesnil et al., 2013). The BLSTM takes a line of tokenized text as input, and makes a binary prediction for each token as to whether it is likely to comprise (part of) an ADR mention. Our BLSTM uses the same architecture as one previously developed for the task of extracting ADR mentions from social media text, Cocos et al. (2016). We make only one slight modification to adapt it for the drug label dataset, using word embedding features trained over medical text rather than social media text.

3.2.1 Model Architecture

A BLSTM is a recurrent neural network that operates over a sequence of tokens in both directions (left-to-right and right-to-left), ultimately predicting a label for each token in the sequence. Specifically, our BLSTM combines two RNNs: a *forward* RNN proceses the sequence from left to right, and a *reverse* RNN processes the sequence from right to left. The outputs of both RNN are averaged for each token to compute the model's final label prediction. The predicted ADRs for each sequence are precisely each contiguous span of positively-predicted tokens.

The RNNs each consist of a single layer of 256 LSTM hidden units. We implement the model using the Keras Python library (Chollet et al., 2015) over a Theano backend (Bergstra et al., 2010; Bastien et al., 2012), optimizing for cross-entropy loss. The code for our model is publicly available.[2]

The BLSTM represents each token as a fixed-length real-valued word embedding. The word embeddings, which we hold fixed (do not allow the model to update) through training, are 100-dimensional FastText (Bojanowski et al., 2016) embeddings trained over the MEDLINE® abstracts corpus (roughly 3B tokens).

3.2.2 Data normalization and labeling

Lines passed to the BLSTM from the table extraction module must be pre-processed prior to input. We do minimal pre-processing of the text from the XML drug labels, simply tokenizing (using the NLTK `word_tokenize` module (Bird et al., 2009)) and converting all letters to lowercase.

For training, we assign each token a binary label indicating whether it falls within (*I*) or outside (*O*) an annotated ADR mention. When labeling the training data, we take care to label as *I* only tokens belonging to *positive* ADR mentions (i.e. not negated, and not related by a Hypothetical relation to a DrugClass or Animal entity).

3.2.3 Model Training

To train the BLSTM, we used all lines from all annotated drug labels that were not flagged by the ta-

[2] https://github.com/chop-dbhi/twitter-adr-blstm

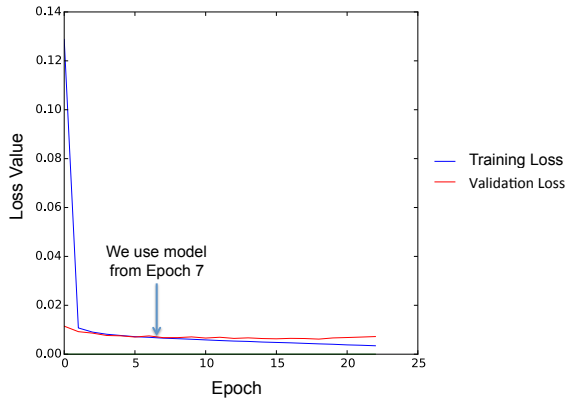

Figure 2: Training curve for our BLSTM model. For prediction we use the model saved after epoch 7.

Metric	Approx. Match	Exact Match
Micro-Precision	80.02	63.84
Micro-Recall	81.64	65.14
Micro-F1	80.82	64.48

Table 2: Performance of the BLSTM in terms of *approximate* and *exact* matches of ADR spans in the validation set.

ble extraction module as comprising a table (8438 lines in total). We tokenized and normalized the lines, and randomly split them into 90% training and 10% validation subsets. We then trained the BLSTM online (one line at a time) for 22 epochs. We ultimately used the model that was saved when the training and validation losses began to diverge, after 7 epochs. Figure 2 shows the model training curve.

In order to gauge the performance of the BLSTM part of the pipeline during development, we evaluated the precision, recall, and F-score of predicted ADR tokens in the validation set. We examined two metrics: *approximate matching*, which considers a predicted ADR span to be correct if it overlaps with any ground-truth span, and *exact matching* (used in the TAC evaluation) which considers a predicted ADR span to be correct only if it exactly matches a ground-truth span. Our model's scores over the validation set are given in Table 2.

3.3 Evaluation and Results

The macro- and micro- precision, recall, and F-score achieved by our entire pipelined model over the test set are given in Table 3.

Metric	Test
Micro-Precision	64.29
Micro-Recall	39.57
Micro-F1	48.99
Macro-Precision	62.97
Macro-Recall	39.95
Macro-F1	47.99

Table 3: Full Task 3 ADR extraction results over the test set. Our system correctly identified 2510 of the 6343 true ADRs, with 1394 false positives and 3833 false negatives.

Metric	Test
Micro-Precision	71.78
Micro-Recall	50.14
Micro-F1	59.04
Macro-Precision	70.12
Macro-Recall	49.84
Macro-F1	57.27

Table 4: Full Task 4 normalization results over the test set. Our system correctly identified 2600 of the 5185 normalizations in the test set, with 1022 false positives.

4 Normalization Method

After identifying predicted ADRs within the drug label text, the CHOP team implemented a simple normalization system to map each ADR to its most applicable MedDRA® preferred term (PT) and lower-level term (LLT). We performed the mapping using a very simple method based on word embedding similarity.

In order to map a predicted ADR to its most-applicable LLT, we simply select the LLT whose word embedding is closest to that of the predicted ADR based on cosine similarity. We then mapped the LLT to its associated PT based on the MedDRA taxonomy. For this task we use the same MEDLINE-trained FastText word embeddings that were used for the BLSTM.

The macro- and micro- precision, recall, and F-Score achieved by our normalization system are given in Table 4.

5 Conclusion

In this paper we have summarized the systems submitted by the CHOP team for the ADR extraction (Task 3) and normalization (Task 4) tasks for the 2017 NIST Text Analytics Conference, ADR

track.

Our system for ADR extraction consists of a two-stage pipeline, with a rule-based table extraction module followed by a binary recurrent neural network used to identify likely ADR mentions.

Our system for normalization relied on simple word embedding cosine similarity, using word embeddings trained over the MEDLINE Abstracts corpus.

References

Syed Rizwanuddin Ahmad. 2003. Adverse drug event monitoring at the food and drug administration. *Journal of general internal medicine* 18(1):57–60.

Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian Goodfellow, Arnaud Bergeron, Nicolas Bouchard, David Warde-Farley, and Yoshua Bengio. 2012. Theano: new features and speed improvements. *arXiv preprint arXiv:1211.5590* .

James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. 2010. Theano: A cpu and gpu math compiler in python. In *Proc. 9th Python in Science Conf*. pages 1–7.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606* .

Elliot G. Brown, Louise Wood, and Sue Wood. 1999. The medical dictionary for regulatory activities (meddra). *Drug Safety* 20(2):109–117.

François Chollet et al. 2015. Keras.

Anne Cocos, Alexander G Fiks, and Aaron J Masino. 2016. Deep learning for pharmacovigilance: recurrent neural network architectures for labeling adverse drug reactions in twitter posts. *Journal of the American Medical Informatics Association* 24(4):813. https://doi.org/10.1093/jamia/ocw180.

Katja M Hakkarainen, Khadidja Hedna, Max Petzold, and Staffan Hägg. 2012. Percentage of patients with preventable adverse drug reactions and preventability of adverse drug reactions–a meta-analysis. *PloS one* 7(3):e33236.

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

Hui Li, Xiao-Jing Guo, Xiao-Fei Ye, Hong Jiang, Wen-Min Du, Jin-Fang Xu, Xin-Ji Zhang, and Jia He. 2014. Adverse drug reactions of spontaneous reports in shanghai pediatric population. *PLoS One* 9(2):e89829.

Grégoire Mesnil, Xiaodong He, Li Deng, and Yoshua Bengio. 2013. Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In *Interspeech*. pages 3771–3775.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45(11):2673–2681.

Janet Sultana, Paola Cutroneo, and Gianluca Trifirò. 2013. Clinical and economic burden of adverse drug reactions. *Journal of pharmacology & pharmacotherapeutics* 4(Suppl1):S73.