Normalizing Adverse Events using Recurrent Neural Networks with Attention

Kahyun Lee, M.S., Özlem Uzuner, Ph.D.
George Mason University, Fairfax, VA, USA

Abstract

Adverse events (AEs) are undesirable outcomes of medication administration and cause many hospitalizations as well as even deaths per year. Information about AEs can enable their prevention. Natural language processing (NLP) techniques can identify AEs from narratives and match them to a structured terminology. We propose a novel neural network for AE normalization utilizing bidirectional long short-term memory (biLSTM) with attention mechanism that generalizes to diverse datasets. We train this network to first learn a framework for general AE normalization and then to learn the specifics of the task on individual corpora. Our results on the datasets from the Text Analysis Conference (TAC) 2017-ADR track, FDA adverse drug event evaluation shared task, and the Social Media Mining for Health Applications Workshop & Shared Task 2019 show that our approach outperforms widely used rule-based normalizers on a diverse set of narratives. Additionally, it outperforms the best normalization system by 4.86 in macro-averaged F1-score in the TAC 2017-ADR track.

Introduction

Adverse events (AEs) are undesirable outcomes of medication administration and affect many people every year. Severe AEs are one of the leading causes of death in the United States. Furthermore, AEs are very costly; each occurrence of an AE increases healthcare costs by more than $3,200. However, AEs are preventable if information about them is available. The US Food and Drug Administration (FDA) Adverse Event Reporting System (FAERS) is the primary source of information about AE occurrences. This system aggregates information from the FDA post-marketing safety surveillance and risk assessment programs. More specifically, the FDA receives AE reports from consumers and manufacturers to minimize potential AE risks of prescription drugs. In order to convert the received information into a useful form, FDA clinical reviewers evaluate each report and manually extract AEs from unstructured text. They then manually code the extracted events using the Medical Dictionary for Regulatory Activities (MedDRA) terminology and link the events to their standardized terms. This provides a valuable resource for studying, understanding, and preventing AEs. However, in 2018 alone, the FDA received over 2 million reports. Given the volume and variety of received reports, AE extraction and MedDRA coding can benefit from automation. In addition to the FAERS reports, social media provides a complementary perspective on AEs. Many AEs are reported in social media by consumers, however, these AEs are buried in a plethora of irrelevant information and present challenges to extraction due to informal language, shorthand notations, ad hoc abbreviations, and the overall unstructured nature of consumer-generated social media text. Identification of AEs as experienced and described by patients via social media provides information on AEs that remain unreported to FAERS.

Natural language processing (NLP) technologies can help to handle the ever-growing volume of AE information automatically. This reduces the need for manual processing while making this information accessible for many applications. In the case of automatically processing AE mentions in narratives, NLP methods consist of two steps. First, AE extraction identifies spans of AE mentions in narratives. Second, AE normalization maps the text of these identified spans, called AEs descriptions, to a terminology, such as preferred terms (PTs) in MedDRA. AE normalization handles variability in the mentions of the same concept, e.g., myocardial infarction vs. heart attack. To obtain an as complete as possible picture of AEs, both of these steps need to be highly accurate and generalizable. Neural networks achieved great success on AE extraction, and various named entity recognition (NER) tasks which can be used for AE extraction. In this work, we focus on the AE normalization which is generally regarded as more difficult than NER. While rule-based normalizers are prevalent, recent work has shown that neural networks can achieve competitive performance and adapt more easily to new domains such as social media. However, these findings have only been validated on small datasets containing a limited number of normalized terms.

In this work, we propose a deep neural network for AE normalization. This neural network utilizes bidirectional long short-term memory (biLSTM) units with an attention mechanism. Given AE descriptions as inputs, it outputs normalized MedDRA PT ids. We train this network in two stages. First, to learn the general task, we train using large concept descriptions collected from associated terms of concepts from the Unified Medical Language System (UMLS). Second, to provide specialization of the framework to variations in observations in the data due to language and presentation, we further train using AE descriptions from task-specific datasets. We evaluate our system on three different datasets from three recent AE shared tasks: Text Analysis Conference (TAC) 2017-ADR track (henceforth,
TAC 2017), the FDA adverse drug event evaluation shared task17 (henceforth, the FDA shared task), and the Social Media Mining for Health Applications Workshop & Shared Task 201918 (henceforth, SMM4H 2019). Our results show that our system significantly outperforms widely-used normalizers such as Metamap19 and cTakes20 in AE normalization. It also outperforms the best performing system in TAC 2017.

This paper has three main contributions: (1) We propose a highly accurate normalizer based on recurrent neural networks (RNNs). To the best of our knowledge, this is the first work that applies bilSTM units with attention mechanism to an AE normalization task. (2) We train this system both to learn AE normalization in general and also to learn the manifestation of this task on individual corpora. The resulting system, evaluated on diverse datasets, provides a highly generalizable AE normalizer which can be applied to different corpora without further training. (3) We include the entirety of MedDRA terminology in this system. In other words, our system can encode an extracted AE description to any PT in MedDRA. This overcomes the demerits of existing machine learning-based normalizers that utilize small training sets for predicting a limited set of MedDRA terminology.

Related work

Both rule-based and machine learning-based methods have been used for medical concept normalization. The more robust and still dominant methods are rule-based, examples of which include: MetaMap19, cTakes20, Mgrep21, NegFinder22, Peregrine23, and Whatizit24. Among those, the most popular system is MetaMap, developed by the National Library of Medicine. MetaMap first parses arbitrary text inputs into noun phrases. Then it searches acronyms, abbreviations, variants, and synonyms of the noun phrases from dictionaries to retrieve candidate concepts from the UMLS Metathesaurus. Lastly, it scores each candidate using an evaluation function specifically devised for concept matching. Although the performances of other rule-based systems can vary, their pipelines are basically the same, but use different resources, target different terminologies, or add new NLP modules25,26.

In general, rule-based normalizers focus on morphological or semantic similarities between input text and the words in normalized terms. This means that in order for them to map input text to a normalized term, there must be overlapping keywords or a predefined relationship between the two. These approaches are, therefore unable to provide mappings when the input and the target normalized terms have no common keywords or have no known relationships in the existing dictionary. For example, the input ‘difficult to come off’ and its normalized term ‘withdrawal syndrome’ share no keywords and provide a challenge for rule-based normalizers. In addition, rule-based normalizers do not transfer well from one dataset to another, e.g., a system developed on EHR narratives would not perform well on social media text.

Machine learning approaches can address the challenges faced by rule-based methods on medical concept normalization. Limspatham and Collier4 proposed a neural networks-based normalizer which outperformed rule-based ones on social media datasets. Their model used no more than pre-trained word embeddings as features. Tutubalina et al.11 applied both RNNs and convolutional neural networks (CNNs) with various pre-trained word embeddings and semantic similarity features such as TF-IDF. For the recent shared task on normalization in TAC 2017, three out of the five participating teams used a rule-based approach27; however, the best system in the shared task used a combination of rules and machine learning1. This system employed the BM25 ranking score28, Jaccard similarity score, and translation-based ranking score27. It first retrieved 10 candidates for each input using the BM25 model in Lucene, calculated the BM25 ranking score, Jaccard similarity score, and translation-based ranking score for each candidate, and combined the scores using linear RankSVM28. The highest ranked MedDRA term was then chosen as the normalized term for each input.

Despite their contributions to this task, machine learning-based normalizers also have their challenges. For example, they need large training data sources, which may be scarce or unavailable. Per Kang et al.29, there have been not as many medical concept normalization shared tasks as named entity recognition ones, resulting in limited annotated data for the normalization task. Additionally, only fragments of the overall task are represented, since training data may cover only parts of terminologies. For example, the TAC 2017 dataset contains 1,941 PTs for 4,475 unique AE descriptions, but there are a total 22,774 PTs in MedDRA v20.1. Therefore, a machine learning-based normalizer
AE description - MedDRA PT collection from UMLS

trained on this data would be able to normalize at best 1,941 PTs out of the 22,774 in MedDRA.

Our system addresses the shortcomings of both rule-based and machine learning-based systems. Unlike rule-based systems, it can normalize AE expressions even in the absence of keyword matches with the normalized terms, and it can generalize to a diverse set of corpora, such as FAERS reports or social media text from Twitter. Unlike machine learning methods that learn only a fragment of the overall task, it provides a generalizable approach to the complete task. As a result, it can outperform well-known rule-based normalizers as well as the state of the art.

Methods

Focusing specifically on AE normalization, we present an RNNs-based normalizer. This normalizer utilizes biLSTM units and attention mechanism suggested by Bahdanau et al.29 We train this system first to learn pairs of AE descriptions and PT ids from a general resource, the UMLS. We then tune it to individual corpora by further training on samples from those corpora.

MedDRA and UMLS metathesaurus

The FDA uses MedDRA terminology for its AE regulatory activity. There are five levels in the MedDRA hierarchy; system organ class (SOC), high level group term (HLGT), high level term (HLT), PT, and lowest level term (LLT). PTs and LLTs, are relevant in this paper. PTs correspond to medical concepts such as symptoms, signs, or indications. LLTs represent variations in term usage for each PT. They are the most specific level in the MedDRA hierarchy, and are linked to only one PT. For example, LLT ‘feeling queasy’ is classified into PT ‘nausea’ < HLT ‘nausea and vomiting symptoms’ < HLGT ‘gastrointestinal signs and symptoms’ < SOC ‘gastrointestinal disorders’.

Although the FDA uses MedDRA PTs to uniquely identify AEs, the same AE may be represented in different ways by different terminologies. This presents challenges to effective retrieval of this information. The UMLS collects all medical concepts including AEs from over 200 terminologies such as MedDRA, SNOMED CT30 and ICD1031, and organizes them by concept unique identifiers (CUIs). For example, ‘acute abdomen’ in ICD10, ‘acute abdominal pain syndrome’ in SNOMED CT and ‘abdominal syndrome acute’ in MedDRA cluster into CUI C0000727. This large biomedical thesaurus enables us to gather AE expressions scattered in various terminologies in one place, and allows us to utilize CUIs to identify all AE expressions associated with each MedDRA PT. We refer to each of these MedDRA associated CUI descriptions as AE descriptions. We exclude duplicated AE descriptions and convert the unique ones into AE description-MedDRA PT pairs. Each MedDRA PT id has at least one and up to 185 AE descriptions. Figure 1 illustrates how we collect AE descriptions from UMLS.

Shared task datasets

We evaluate AE normalization using corpora from three shared tasks. TAC 2017 and the FDA shared tasks aim to retrieve AE descriptions from structured product labels (SPLs) and code the retrieved descriptions into MedDRA PTs. SMM4H tackles the same task but on tweets.

The TAC 2017 dataset contains 13,436 AE descriptions from 200 SPLs. 101 out of 200 SPLs in TAC 2017 are in the training set while the test set consists of the remaining 99. The total number of unique MedDRA PTs in this dataset is 1,941. 13,318 AE descriptions in this data set are mapped to single PTs but the remaining 118 descriptions are mapped into two PTs. For example, input AE description ‘angioedema of the mouth’ is mapped into both PT ‘10030110 oedema mouth’ and PT ‘10002424 angioedema’. No description matches 3 or more PTs.
Table 1. Overview of the shared task datasets and comparison to UMLS Metathesaurus

<table>
<thead>
<tr>
<th>Source of input data</th>
<th>UMLS</th>
<th>TAC 2017 ADR track</th>
<th>FDA shared task</th>
<th>SMM4H 2019</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of records</td>
<td>-</td>
<td>SPL</td>
<td>SPL</td>
<td>Tweet</td>
</tr>
<tr>
<td>Total number of AE descriptions</td>
<td>428,673</td>
<td>13,436</td>
<td>16,427</td>
<td>1,122</td>
</tr>
<tr>
<td>Total number of unique AE descriptions</td>
<td>241,096</td>
<td>4,475</td>
<td>5,156</td>
<td>746</td>
</tr>
<tr>
<td>Total number of unique MedDRA PTs</td>
<td>22,774</td>
<td>1,941</td>
<td>1,946</td>
<td>248</td>
</tr>
<tr>
<td>Average # of tokens in unique AE descriptions</td>
<td>4.07</td>
<td>3.79</td>
<td>3.64</td>
<td>3.39</td>
</tr>
</tbody>
</table>

The FDA shared task training data contains 16,427 AE descriptions from 100 SPLs. Unlike the ones in TAC 2017, each description in the FDA shared task matches with only one PT. The total number of unique PTs in the training set was 1,946. SMM4H 2019 dataset contains 1,122 AE descriptions from 684 tweets. No descriptions are mapped into multiple PTs. SMM4H 2019 also focuses on a relatively smaller number of targets, 347 LLTs. To make it consistent with other two shared tasks, we convert each of the 347 LLTs into its corresponding PT, resulting in 248 unique PTs. The gold standard for the official test sets for the SMM4H 2019 and the FDA shared tasks are not publicly available. Therefore, we perform a 5-fold cross validation with the training set for each shared task. Only for TAC 2017, we additionally evaluate our methods on the test set of the shared task. Table 1 summarizes the statistics of each dataset in comparison to the overall UMLS Metathesaurus.

Preprocessing

For AE descriptions mapped into two PTs, either PT would be an acceptable answer. Therefore, we keep only the first observed PT as a gold PT and drop the other PT. This prevents our model from observing two different answers for the same AE description. After collecting AE descriptions from training data, we remove all the special characters from AE description except for angle brackets (‘<’ and ‘>’), because these brackets have a very important meaning in some contexts. Especially, ‘>’ can represent ‘increased’, ‘elevated’, or ‘high’ (e.g., ‘>’ as in ‘AST >5.0’) while ‘<’ can stand for ‘decreased’ or ‘low’ (e.g., ‘<’ as in ‘hemoglobin (g/dl) <10’).

Systems

AE normalization requires modeling the language in AE descriptions and mapping them to the MedDRA PTs. We tackle this task with a novel AE normalizer that consists of biLSTMs with an attention mechanism and compare it to a vanilla-biRNN baseline.

vanilla-biRNN Baseline: RNNs are widely used to capture patterns in language because they can incorporate information from past observations when making predictions. Vanilla-biRNN has a simple structure and can only take advantage of short distance dependencies in the given input sequences.

biLSTM with Attention AE Normalizer: This novel AE normalizer consists of biLSTMs with attention mechanism, followed by dropout and fully-connected layer. Below, we introduce these components and describe the structure of the system.

- **biLSTM unit**: To capture longer distance dependencies than regular RNNs, two kinds of RNNs with memory are introduced. One is the LSTM and the other is the gated recurrent unit (GRU). Unlike the single-RNNs, these two RNNs can capture long distance dependencies using ‘gates’. The gates decide what are memorized and what are forgotten from their past observations. The LSTM, which consists of an input gate, a forget gate and an output gate, has been widely used for various tasks. The GRU is more recently introduced and consists of only two gates; a reset gate and an update gate. The GRU becomes popular because of its structural simplicity and competitive performance. Both LSTM and GRU can be run in a bidirectional setting, referred to as biLSTM and biGRU. This incorporates information from future observations as well as past observations into these systems, giving them an edge in performance. We tested both biLSTM and biGRU for AE normalization. Through cross-validation experiments on UMLS, we obtained better results from the biLSTM than biGRU, so we adopted biLSTM for our AE normalizer.

- **Attention**: The attention mechanism adjusts the weights of the observations in an input sequence so as to focus on the most significant clues that can provide a mapping from the input sequence to the output. Since the mechanism was initially introduced, several types of attention have been proposed.
use the attention mechanism proposed by Bahdanau et al.\textsuperscript{29}. In a typical network with this attention mechanism, a context vector is calculated using attention weights which is learned to estimate the importance of each element in the input sequence. This context vector is then concatenated with the original RNN’s output to make prediction. In our system, attention provides clues for the best mapping between AE descriptions and MedDRA PT ids by putting different weights for each input. For example, when a model receives ‘cardiovascular thrombotic events’ as an input, it should learn ‘cardiovascular’ as being more important than ‘thrombotic’ for proper normalization throughout the training process so that it matches the description into PT ‘10007649 cardiovascular disorder’ rather than ‘10043647 thrombotic stroke’.

- **Fully-connected layer:** The fully-connected layer connects each neuron in a layer with all the neurons of the previous layer, capturing non-linear relationships between inputs and outputs. In AE normalization, this layer generates a score for each PT id for each input AE description. The highest scored PT id is selected as the prediction for the corresponding AE description.

**Structure:** The structure of our AE normalizer is shown in Figure 2 and consists of a token-embedding layer, a biLSTM layer, an attention layer, a dropout layer, and a fully-connected layer. The input AE description is first converted into a sequence of pre-trained word vectors through the token-embedding layer. This sequence feeds into the biLSTM layer. Before the biLSTM output goes into the fully-connected layer, the vector is concatenated with the context vector which consists of a dot product of attention weight and hidden state for each time stamp. Some of the concatenated vector elements are dropped out (according to the dropout rate) and excluded from prediction to prevent overfitting. In the final fully-connected layer, the MedDRA PT id is predicted using the dropout applied concatenated vector.

**Training and hyperparameters:** We tuned the optimizer, learning rate, batch size, token-embedding dimension, and hidden layer dimensions in both biLSTM and attention layer during cross-validation on the training sets of each shared task data. For the token-embedding layer, we used word embeddings that are pre-trained on Wikipedia, all full-text documents from the PubMed Central, and all publication abstracts from PubMed\textsuperscript{39}. These embeddings have previously been shown to produce better results in the clinical domain than other publicly available pre-trained word embeddings such as GloVE\textsuperscript{40}. We experimented with expanding the token-embedding layer with deep affix features\textsuperscript{41} and character-embeddings. However, this expansion did not improve results and was therefore omitted. We set token-embedding dimension of 200, biLSTM dimension of 200, and attention dimension of 100. We applied dropout rate of 0.5. We trained the model using adaptive moment estimation (Adam) optimizer\textsuperscript{42} with the learning rate of 0.001. After tuning the parameters, we trained our system in two steps. For initial training, we used 241,096 unique AE description-PT id pairs extracted from UMLS. We iterated for 30 epochs on UMLS AE descriptions with the mini-batch size of 64. We then continued training on shared task-specific training sets (excluding randomly selected test set in case of FDA shared task and SMM4H 2019) with the mini-batch size of 16. This optimizes our system to the given dataset. We stopped iteration if there was no improvement in accuracy for 10 epochs.

**Evaluation methods and metrics**

We evaluate the performance of our systems on each dataset using 5-fold cross validated accuracy, a metric which has been commonly used in recent medical concept normalization literature\textsuperscript{11,12,13}. We compare our performances to two widely-adopted normalization systems, MetaMap\textsuperscript{19} and cTakes\textsuperscript{20}, which have shown robust performance on multiple datasets\textsuperscript{10,16,25}.

![Figure 2. Overview of the structure of our normalizer](image-url)
We also compare our result on the TAC 2017 shared task test set with the best performing system in the same shared task using the official evaluation script of the shared task. The script measures Precision/Recall/F1-score computed as follows: Precision (P) = true positives / (true positives + false positives), Recall (R) = true positives / (true positives + false negatives), and F1-score (F1) = 2*P*R / (P+R). We use the primary evaluation metric, macro-averaged by SPL F1-score, specified by TAC 2017 organizers. This metric increases emphasis on minority SPLs by weighting each SPL equally regardless of the number of samples it contains.

Statistical significances are calculated using approximate randomization\(^\text{4}\). Approximate randomization uses a null hypothesis that two systems will produce identical scores. This hypothesis is tested by randomly shuffling the predictions of the two systems, re-assigning the shuffled predictions, and testing for statistical significance between the scores generated from the shuffles. Random shuffling is repeated, and the overall statistical significance is determined based on the count of shuffles that produced statistically significant scores. We tested with 9,999 shuffles, and a significance level of 0.01.

**Results**

We trained our system on both the UMLS and each shared task’s training dataset. The accuracies of different normalizers on the three test datasets are summarized in Table 2. We run the two rule-based systems, Metamap\(^\text{19}\) and cTakes\(^\text{20}\), with their default settings. We use vanilla-biRNN as our baseline to examine the usefulness of biLSTM units and attention mechanism on this task. We evaluate our system performance in two settings: (1) after initial training on UMLS AE descriptions and (2) after additional training with 4 out of 5-folds training dataset for each shared task. Separate from those two-step settings, we also perform 5-folds cross validations only with individual shared-task datasets to investigate the effect of training on UMLS before tuning to specific datasets. To measure the contribution of attention to our system, we also evaluate our model without attention.

In all combinations, RNNs systems show better performance than rule-based systems except only one case of vanilla-biRNN (trained on UMLS) in TAC 2017. The biLSTMs with and without attention outperform the rule-based normalizers in all three datasets without using shared task-specific data. Our systems further outperform both rule-based systems when additionally trained on shared task-specific data.

We compare the performance of biLSTM with attention with the highest performing system in TAC 2017 shared task using the official evaluation script. As presented in Table 3, biLSTM with attention outperforms this system in both micro- and macro-averaged F1-scores. We cannot test the statistical significance of the performance difference because the best system in the shared task is not publicly available.

**Table 2.** The accuracies of different normalizers on the test set of each shared task

<table>
<thead>
<tr>
<th>System</th>
<th>Training data</th>
<th>TAC 2017 ADR Track</th>
<th>FDA Shared Task</th>
<th>SMM4H 2019</th>
</tr>
</thead>
<tbody>
<tr>
<td>cTakes</td>
<td>UMLS</td>
<td>66.92(^*)</td>
<td>67.06(^*)</td>
<td>24.52(^*)</td>
</tr>
<tr>
<td></td>
<td>Shared task data</td>
<td>68.04(^*)</td>
<td>69.89(^*)</td>
<td>27.72(^*)</td>
</tr>
<tr>
<td></td>
<td>UMLS &amp; shared task</td>
<td>81.47(^*)</td>
<td>88.98(^*)</td>
<td>50.00(^*)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>88.30(^*)</td>
<td>91.60(^*)</td>
<td>58.48(^*)</td>
</tr>
<tr>
<td>vanilla-biRNN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>UMLS</td>
<td>65.75(^*)</td>
<td>71.96(^*)</td>
<td>37.95(^*)</td>
</tr>
<tr>
<td></td>
<td>Shared task data</td>
<td>81.47(^*)</td>
<td>88.98(^*)</td>
<td>50.00(^*)</td>
</tr>
<tr>
<td></td>
<td>UMLS &amp; shared task</td>
<td>88.30(^*)</td>
<td>91.60(^*)</td>
<td>58.48(^*)</td>
</tr>
<tr>
<td>biLSTM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>UMLS</td>
<td>74.77(^*)</td>
<td>77.08(^*)</td>
<td>40.18(^*)</td>
</tr>
<tr>
<td></td>
<td>Shared task data</td>
<td>82.11(^*)</td>
<td>89.98(^*)</td>
<td>56.70(^*)</td>
</tr>
<tr>
<td></td>
<td>UMLS &amp; shared task</td>
<td>89.17(^*)</td>
<td>93.64(^*)</td>
<td>68.30(^*)</td>
</tr>
<tr>
<td>biLSTM with attention</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>UMLS</td>
<td>75.00(^*)</td>
<td>77.35(^*)</td>
<td>41.07(^*)</td>
</tr>
<tr>
<td></td>
<td>Shared task data</td>
<td>82.78(^*)</td>
<td>90.23(^*)</td>
<td>56.70(^*)</td>
</tr>
<tr>
<td></td>
<td>UMLS &amp; shared task</td>
<td>89.23(^*)</td>
<td>93.79(^*)</td>
<td>68.30(^*)</td>
</tr>
</tbody>
</table>

\(^*\) Statistically significant differences from the vanilla-biRNN run on the same dataset at the level of 0.01 are asterisked.
Table 3. The performance comparison with the best system in TAC 2017 shared task using official evaluation script

<table>
<thead>
<tr>
<th>Best system in the shared task</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Micro</td>
<td>Recall</td>
<td>F1-score</td>
</tr>
<tr>
<td></td>
<td>84.17</td>
<td>89.84</td>
<td>86.91</td>
</tr>
<tr>
<td></td>
<td>Macro</td>
<td>83.02</td>
<td>89.06</td>
</tr>
<tr>
<td>biLSTM with attention trained on UMLS</td>
<td>Micro</td>
<td>76.79</td>
<td>82.58</td>
</tr>
<tr>
<td></td>
<td>Macro</td>
<td>76.05</td>
<td>81.17</td>
</tr>
<tr>
<td>biLSTM with attention trained on UMLS &amp; shared task data</td>
<td>Micro</td>
<td>90.45</td>
<td>91.48</td>
</tr>
<tr>
<td></td>
<td>Macro</td>
<td>89.76</td>
<td>90.70</td>
</tr>
</tbody>
</table>

(*) The primary metric measure of TAC 2017 shared task for each system is asterisked.

Discussion

Results in Table 2 show that the biLSTMs with and without attention outperform the rule-based normalizers in all three datasets trained using only UMLS data. Since no task-specific data is used in training, this shows that our approach is highly generalizable. There is however, some variability in the results. For example, unlike AE descriptions from SPLs, the ones from SMM4H 2019 are more informal and thus harder to normalize. As a result, all systems struggle on social media text; however, the performance drop of rule-based systems is larger. MetaMap can match only 27.72% of AE descriptions correctly whereas biLSTMs with attention map 41.07% and 68.30% correctly without and with shared task-specific data respectively. biLSTM systems also significantly outperform the vanilla-biRNN, and for shared tasks using clinical data the performance difference is more pronounced before training with task-specific data. When the systems were trained on UMLS only, the accuracy increased from vanilla-biRNN to biLSTM by 9.02 and 5.12 percent points for TAC 2017 and FDA AE Shared Task, respectively. After adding shared task-specific data, the accuracy increased only by 0.87 and 2.04 percent points for TAC 2017 and FDA AE Shared Task. This implies the difference in architecture plays a more important role when training only on UMLS data but the gap between the vanilla-biRNN and the biLSTM solutions close with the addition of shared task-specific training data for clinical tasks. The addition of task-specific training data for the social media task, SMM4H, increased the difference in performance from vanilla-biRNN to biLSTM from 2.23 to 9.82 percent point without and with task-specific data respectively. This implies long term dependencies hidden in the informal text are captured better by the biLSTM than vanilla-biRNN. The self-attention mechanism further improves performance but its effect is less than that of switching from a vanilla-biRNN to a biLSTM architecture; 0.23 and 0.27 percent points without shared task-specific data and 0.06 and 0.15 percent points with shared task-specific data for TAC 2017 and FDA AE Shared Task respectively. No explicit accuracy increases with attention mechanism are observed in SMM4H 2019. This is likely because of small test dataset, which contains only 224 AE descriptions.

Error Analysis

The strengths and weaknesses of each system are revealed by manually reviewing the errors. Both of the evaluated rule-based systems work well when there are exact matches or pre-defined related terms between input AE description and PT description. Both correctly normalize the inputs ‘night terrors’ into ‘10041010 sleep terrors’, ‘tiredness’ into ‘10016256 fatigue’, and ‘allergic reaction’ into ‘10020751 hypersensitivity’. However, even when there are exact matches or pre-defined related terms between the input and gold standard descriptions, rule-based systems can miss some cases. This happens when multiple undefined terms coexist in the input description. Put simply, the rule-based systems are less generalizable than machine learning systems, and fail when AEs are described in unspecified ways. For example, the rule-based systems correctly map ‘feeling ill’ to the normalized term ‘10025482 malaise’, but fail to map ‘feeling like shit’ to the same normalized term. This limitation is particularly pronounced when normalizing social media text. In contrast, our biLSTM systems can expand their ability to informal AE descriptions without losing useful information from general resources like the UMLS. These systems correctly normalized the fore-mentioned input AE descriptions. Some additional interesting examples of correct AE description to MedDRA PT mappings which both rule-based systems missed include ‘coming off effexor is NOT fun’ to ‘10048010 withdrawal syndrome’, ‘zombie’ to ‘10016322 feeling abnormal’, and ‘don’t want to get up’ to ‘10041349 somnolence’. Those AE descriptions are all informal and do not share any token with MedDRA PT, and are therefore difficult for rule-based normalizers.

The errors from our best system from each dataset can be divided into several categories. The main category contains erroneous predictions which are closely related to the ground truth. For example, for the input AE description ‘chloride low’, our system mapped the description into ‘10021021 hypochloraemia’ where the ground truth was ‘blood chloride low’.
decreased’. In the case of ‘permanent vision loss’, our system mapped the AE description to ‘10047571 visual impairment’ but the ground truth was ‘10005169 blindness’. Another example is ‘10027374 mental impairment’. For the input AE description, our system predicts ‘impair mental abilities’ as its normalized term, where ‘10027353 mental disability’ was the ground truth. These incorrectly normalized terms are very closely related with ground truth. In some cases, our system predictions are nearly synonymous or more specific than ground truth annotations. The ground truth of AE description ‘worsening in mood’ is ‘10027940 mood altered’ whereas our system prediction is ‘10027951 mood swings’. Neither ‘alteration’ or ‘swing’ imply negative change, as described by the input description ‘worsening’. For another example, AE description ‘facial nerve paresis’ is annotated with ‘10033985 paresis’ in gold. Our system prediction for the description is ‘10051267 facial paresis’ which is more specifically related to the description than the ground truth. This strength mainly comes from the power of pre-trained word embeddings and the high integrity of UMLS Metathesaurus. Conversely, our system misses some self-evident cases where input AE descriptions exactly match with PT descriptions. Such cases can be easily normalized by rule-based normalizers, but since our system does not directly compare input AE descriptions with MedDRA PT descriptions these cases may be missed.

Another type of error occurs when our system focuses on the less important parts of an AE description. For example, our prediction for the AE description of ‘bacterial opportunistic infections’ was ‘10030901 opportunistic infection’ whereas the ground truth was ‘10060945 bacterial infection’. Our system should have been trained to put more weight on ‘bacterial’ than ‘opportunistic’ throughout training.

Our system made another category of errors caused when an AE description contained unobserved tokens; especially, a substantial amount of medical acronyms. For example, ‘ETT’ (endotracheal tube), ‘IRR’ (infusion related reaction) and ‘MI’ (myocardial infarction) are not observed during training but are included in the test set. Although those acronyms exist in the UMLS, they are not found in any of the 241,096 unique AE descriptions (see Table 1) we extracted from UMLS and used in training. Incorporating external dictionaries could help mitigate these problems.

Limitations

We acknowledge some limitations of this study. The first problem is n to n mapping between CUIs and MedDRA PTs in the UMLS. In 323 cases, one CUI matches with multiple MedDRA PTs. For example, CUI C0020541 matches with both MedDRA PT ‘10025600 malignant hypertension’ and ‘10000358 accelerated hypertension’. Our model cannot differentiate the two AEs because we extract ‘AE description-MedDRA PT’ pairs by CUI, those two PTs share the same set of AE descriptions. Another limitation is that our model is sensitive to the inconsistencies of the dataset. In TAC 2017 shared task training set, AE description ‘oxygen desaturation’ is annotated with MedDRA PT ‘10021143 hypoxia’ in one SPL, but with another PT ‘10050081 neonatal hypoxia’ in another SPL. This gives two different labels for the same input which inevitably deteriorates our system performance. Finally, unlike some normalizers which can process raw clinical text and extract medical concepts, our system does not contain a medical concept extractor. It needs a separate NER system, such as the one presented in Tao et al. or Dernoncourt et al.

Conclusion

In this paper, we propose a biLSTM with attention mechanism for AE normalization. This system can handle the general AE normalization task by first learning mappings of AE descriptions from UMLS to MedDRA PT ids. It can then be further improved by adding training instances from a specific dataset. The resulting model differs from other machine learning-based models that only learn the PTs found in task-specific datasets; unlike those models, it can encode an AE description to any MedDRA PT id. In addition to being a more general solution to AE normalization, this system outperforms existing widely-adopted rule-based normalizers such as cTakes and MetaMap based only on mappings learned from UMLS. It also outperforms the best system in the TAC 2017 shared task. Error analysis of our system shows that this system can capture semantic meaning contained in MedDRA PTs but can miss AE descriptions that exactly match MedDRA PT descriptions which can be encoded easily by rule-based normalizers. Unobserved tokens such as rare acronyms are another challenge. This implies our system, which is already very generalizable and highly performing, can be further improved by adding more data for training, incorporating rules, or adopting external resources such as an acronyms dictionary into this system.

Acknowledgments

We warmly thank Samuel Henry and Paul F. Barry for their helpful suggestions and assistance.
References

25. Lu CJ, Tormey D, McCreedy L, Browne AC. Enhanced LexSynonym Acquisition for Effective UMLS Concept
44. Tao C, Michele F, Uzuner Ö. Extracting ADRs from Drug Labels using Bi-LSTM and CRFs. AMIA 2018 Annu Symp. 2018;