Computer Methods and Programs in Biomedicine 176 (2019) 33–41

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.elsevier.com/locate/cmpb

An adverse drug effect mentions extraction method based on weighted online recurrent extreme learning machine

Ed-drissiya El-allaly*, Mourad Sarrouti, Noureddine En-Nahnahi, Said Ouatik El Alaoui

Laboratory of Informatics and Modeling, FSDM, Sidi Mohammed Ben Abdellah University, Fez, Morocco

ARTICLE INFO

Article history:
Received 6 January 2019
Revised 5 April 2019
Accepted 26 April 2019

Keywords:
Adverse drug effect
Weighted online recurrent extreme learning machine
Biomedical named entity recognition
Natural language processing
Biomedical informatics
Pharmacovigilance

ABSTRACT

Background and objective: Automatic extraction of adverse drug effect (ADE) mentions from biomedical texts is a challenging research problem that has attracted significant attention from the pharmacovigilance and biomedical text mining communities. Indeed, deep learning based methods have recently been employed to solve this issue with great success. However, they fail to effectively identify the boundary of mentions. In this paper, we propose a weighted online recurrent extreme learning machine (WOR-ELM) based method to overcome this drawback.

Methods: The proposed method for ADE mentions extraction from biomedical texts is divided into two stages: span detection and ADE mentions classification. At the first stage, we identify the boundary of the mentions irrespective of their types with a WOR-ELM in a given sentence. At the second stage, another WOR-ELM is used to classify the identified mentions to the appropriate type. Both stages use the concatenation of character-level and word-level embeddings as features. The character-level embedding is obtained using a modified online recurrent extreme learning machine, whereas the word-level embedding is obtained from a pre-trained model.

Results: Several experiments were carried out on a well-known ADE corpus to evaluate the effectiveness and demonstrate the usefulness of the proposed method. The obtained results show that our method achieves an F-score of 87.5%, which outperforms the current state-of-the-art methods.

Conclusions: Our research results indicate that the proposed method for adverse drug effect mentions extraction from text can significantly improve performance over existing methods. Our experiments show the effectiveness of incorporating word-level and character level embeddings as features for WOR-ELM. They also illustrate the benefits of using IOU segment to represent ADE mentions.

1. Introduction

Adverse drug effect (ADE), which is defined as an injury caused by the drug at the normal dosage [1], is a major public health concern since drugs often lead to unexpected side effects or a variety of adverse drug reactions (ADR). Therefore, the early detection of ADE is of vital importance to minimize a drug-associated morbidity and mortality. Typically, most information about ADE are obtained through the clinical trials [2]. However, they cannot detect all ADEs, mainly because the low number of participants with relatively short duration. As a result, unobserved ADE from clinical trials which are identified during post-marketing surveillance are reported to the spontaneous adverse event reporting systems (AERS), like food and drug administration (FDA[1]). However, studies have shown that AERS receive a low number of reports [3]. On the contrast, the most up-to-date unstructured information relevant to ADE is hidden in biomedical publications from the MEDLINE database [4]. This represents an expanding data collection currently comprised of about more than 25 million citations[2] which contains also more than 340,000 ADE case reports [5]. The ADE extraction from biomedical texts is generally composed of two tasks including: (1) mentions recognition such as drugs and diseases, and (2) identification of possible ADE relations between them. The first task can be considered as a biomedical named entity recognition (BNER) problem, while the second can be considered as a relation classification problem. In this paper, we focus on the first task to identify mentions of ADE from textual data. Indeed, the recognition of mentions of ADE from biomedical

* Corresponding author.
E-mail addresses: eddrissiya.elallaly@usmba.ac.ma (E.-d. El-allaly), mourad.sarrouti@usmba.ac.ma (M. Sarrouti), noureddine.en-nahnahi@usmba.ac.ma (N. En-Nahnahi), said.ouatikelalaoui@usmba.ac.ma (S. Ouatik El Alaoui).

[1] https://www.fda.gov/.
[2] https://www.nlm.nih.gov/pubs/factsheets/medline.html.

https://doi.org/10.1016/j.cmpb.2019.04.029
0169-2607/

text remains one of the most important tasks of ADE systems as the overall performance of such systems heavily depends on the effectiveness of the integrated named entity recognition system: if a named entity recognition system fails to identify named entities, further processing steps to extract relationships will inevitably fail too. In the rest of the paper, we use mentions to indicate named entities concerning ADE.

Due to the importance of the BNER task in many biomedical text mining applications such as question answering [6–12], information retrieval [13] and ADE [14,15], recently, this task has witnessed a growing interest among natural language processing (NLP) researchers. In this context, several BNER methods have been developed [16] which can generally be categorized as dictionary-based method [17,18], rule-based method [19,20] and machine learning based method [21–23]. Recently, deep neural network have shown great success for many text mining and natural language processing (NLP) tasks compared with other methods which require much manual feature engineering effort. Recurrent neural network (RNN) [24] and its variants long-short term memory (LSTM) [25] have shown great success in various sequence labelling problems such as BNER [26,27] without handcrafted features. These models are able to learn relevant representations for words. Relatively, few studies have been carried out on extracting ADE mentions from biomedical text by using deep neural network models [28,29]. Despite their successful results, these models cannot learn the semantic dependency between output labels since the overall prediction fails to effectively identify the boundary of mentions which require to integrate additional features to the input layer.

Extreme learning machine (ELM) [30,31], on the other hand, is an efficient learning algorithm for neural networks which has attracted a great attention because of its extremely fast learning speed. More recently, online recurrent extreme learning machine (OR-ELM) [32], a variant of ELM, has shown promising results in time-series prediction problem which outperforms other sequential learning algorithms such as LSTM. OR-ELM is an extension of online sequential extreme learning machine (OS-ELM) [33], which can be applied to learn recurrent neural network by applying a normalization method called layer normalization (LN) and ELM auto-encoder (ELM-AE) [34].

In this paper, we propose an adverse drug effect mentions extraction method based on weighted online recurrent extreme learning machine (WOR-ELM) to identify drugs and diseases mentions from biomedical texts. Our method is composed of two main stages: (1) span detection stage and (2) ADE mentions classification stage, where each stage consist also of two main phases: features representation and classification phases. In the first stage, span detection, which aims to detect the ADE mentions without identifying their types, we first generate the character-level embedding using a modified online recurrent extreme learning machine. We then concatenate it with word-level embedding to get the feature representation for each word in the vocabulary. Next, the extracted features are fed into the WOR-ELM for identifying the mentions from a given input sentence. In the second stage, ADE mentions classification, another WOR-ELM is used to classify the identified mentions obtained by the first stage to the appropriate type, i.e., Diseases or Drugs. We train WOR-ELM on the same features that were used in the span detection stage. Consequently, the proposed ADE mentions extraction method based on WOR-ELM offers significant advantages such as feature learning capability at each stage, fast learning speed and better generalization performance. Experimental results on a standard ADE corpus [35] demonstrate the effectiveness of our proposed method. The obtained results show that our method is more effective as compared with the current state-of-the-art methods [28,29].

The remainder of the paper is organized as follows. In Section 2, we describe a related work of adverse drug effect mentions extraction methods. Section 3 details the proposed method. Section 4 reports the experimental results of the proposed method where experimental settings, evaluation metrics and datasets are also provided. Section 5 discusses the obtained results. Finally, Section 6 concludes this paper and proposes the future works.

2. Related work

The scientific literature has become a very significant source for publishing new drug-associated adverse effect discovered by pharmacovigilance researchers [1]. Therefore, an automatic extraction of ADE from biomedical texts is challenging.

Gurulingappa et al. [36] have adapted a hybrid-based method to extract relations between drugs and diseases from biomedical litteratures. The authors have used ProMiner system, a dictionary derived from ontologies, to recognize ADE mentions from texts. Then, they have employed java simple relation extraction (JSRE) system [37], a machine learning system based on support vector machines (SVMs) with different kernels for relation extraction.

Kang et al. [38] have developed a knowledge-based system for the identification of relations between ADE mentions. The system is composed of two module: concept recognition and knowledge-based. At the first module, the authors have employed a combination of dictionary-based and rule-based methods to effectively identify drugs and diseases mentions from biomedical texts. the Peregrine[3] system was used as a dictionary-based concept recognition. At the second module, they have integrated the informations contained in the unified medical language system (UMLS) to extract potential adverse drug effect relations.

Li et al. [28] have explored a deep neural system to jointly identify drug and disease mentions from biomedical texts and eventually relation between them. They have used a feed-forward neural network classifier which consists of three layers: input layer, hidden layer and output layer. First, the input layer includes a concatenation of variable-length and fixed-length features where the variable-length features such as multi-word entity mentions are transformed to fixed-length embeddings by using convolutional neural network (CNN). Second, the hidden layer makes a leveraged rectified linear units (RELU) non-linear activation function. Finally, the output layer predicts the best label sequence for a given input sentence.

Li et al. [29] have proposed a neural joint model for ADE extraction. They have used a bidirectional LSTM (Bi-LSTM) to recognize drugs and diseases mentions from biomedical texts. The character-level representations with CNN, word and POS embeddings are fed into the networks to learn the representations of mentions for a given input sentence. the authors have used another Bi-LSTM, which was stacked with the first one, for relation extraction.

Though the previous studies have proven to be quite successful of ADE extraction from biomedical texts, they still require further efforts in order to improve their performance. In this work, we propose an efficient method based on weighted online recurrent extreme learning machine ADE mentions extraction from biomedical texts. The contributions that we make in this paper can be summarized in the following points:

- We design a weighted online recurrent extreme learning machine based method to identify ADE mentions from biomedical texts. The proposed method outperforms the state-of-the-art ones on the ADE corpus.

[3] https://trac.nbic.nl/data-mining/.

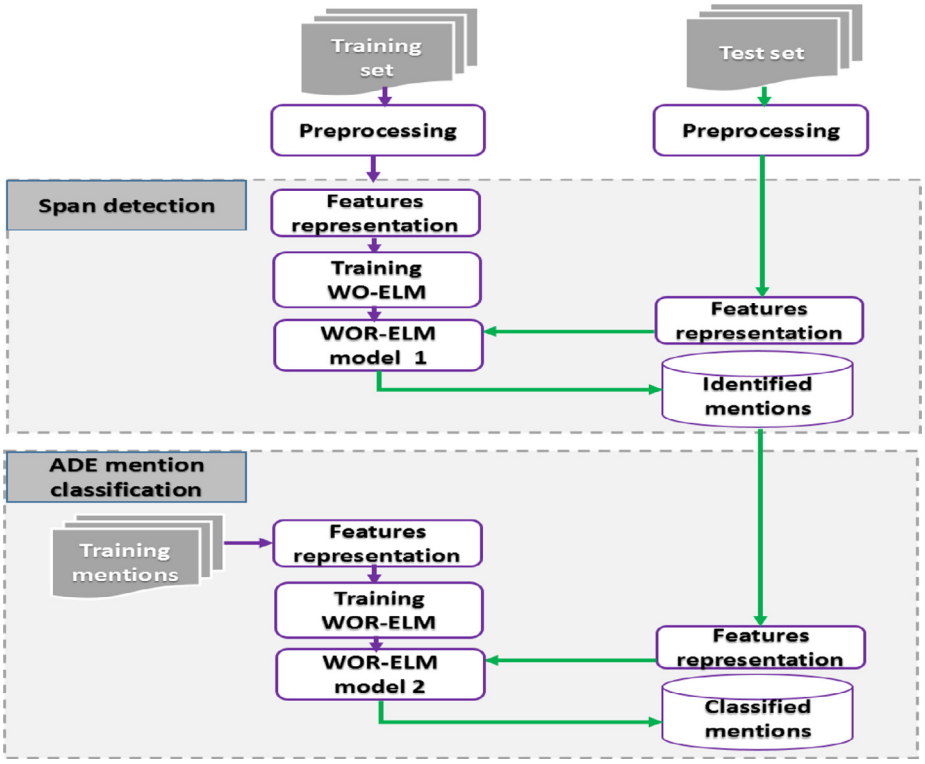

Fig. 1. The flowchart of the proposed adverse drug effect mentions extraction method.

- We propose a character-level embedding with a modified online recurrent extreme learning machine to capture morphological features.
- We evaluate the impact of five well-known segments representation to investigate whether the method can effectively detect the ADE mentions boundary.

3. Methods

The proposed method is comprised of two main stages: (1) span detection to identify the boundary of the mentions in a given sentence through using IOU segment representation, and (2) ADE mentions classification to classify the identified mentions into the appropriate type (Diseases or Drugs). The two stages consist also of two main phases: first, the input words are represented as a dense vector to get the features representation by concatenating word-level embedding and character-level embedding with a modified online recurrent extreme learning machine, then the extracted features are fed into the WOR-ELM for classification. In fact, most of the state-of-the-art ADE mentions extraction systems integrate the span detection stage with the ADE mentions classification one, which the output sequence is represented by combining a tag information from the used segment representation with a semantic mention type. However, the number of features increases according to the increased number of labels. Our proposed method has the feature leaning capability which can extract relevant information from both word-level and character-level embeddings at each stage through using the WOR-ELM. It tends to achieve accurate and better generalization performance with fast learning speed. On the other hand, the proposed character-level embedding has good potential to represent the morphological information of word meaningfully in the hidden layer of the modified OR-ELM by incorporating the ELM-AE and the LN procedure. The flowchart of the proposed method is shown in Fig. 1. The two stages are described sequentially in the following subsections.

3.1. Span detection

The span detection stage is formalized as a sequence labeling problem, where each feature vector of word is labeled with a tag that denotes whether a word is part of a mention for a given input sentence. We use an IOU segment representation which includes three tags, i.e., I (inside), O (outside) and U (Unit). If a mention, irrespective of its type, consists of a single word then the U-Mention tag is used. If a mention contains two or more words, the I-Mention tag is assigned. the O tag is used for the remaining non mention words. For example, given a sentence "*The clinical course suggests that the interstitial pneumonitis was induced by hydroxyurea*", the corresponding label sequence is "*O O O O O O I-Mention I-Mention O O O U-Mention*".

3.1.1. Preprocessing

Text preprocessing is first conducted before features representation which can affect the final performance significantly. In this work, two preprocessing operations are conducted: tokenization and replacing the numbers to zeroes. Each of these operations is detailed below.

Tokenization. We use the implementation provided by the NLTK toolkit.[4] We employ the tokenization using not only whitespaces but also punctuations by defining some heuristic rules to handle the complicated words (e.g., azithromycin-induced).

Replacing the numbers to zeroes. This leads us to reduce the size of the vocabulary and make the embedding more compact.

3.1.2. Features representation

The first layer in the span detection stage converts each word in the sentence into a real vectors that capture semantic and

[4] http://www.nltk.org/.

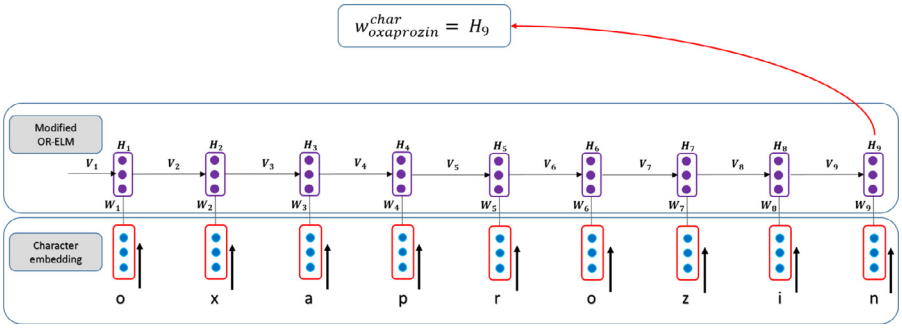

Fig. 2. Example of computing character-level embedding with the modified OR-ELM.

morphological information. Given a sentence $S = \{w_1, w_2, \ldots, w_n\}$ consisting of n words. Each word w_i is converted into a vector $v_i = [w_i^{word}, w_i^{char}]$ which is obtained by concatenation of two sub-vectors: the word-level embedding $w_i^{word} \in \mathbb{R}^{d^{word}}$, and character-level embedding $w_i^{char} \in \mathbb{R}^{d^{char}}$.

Word-level embedding. Also known as distributed word representation, word-level embedding captures distributional semantic information of the word [39]. The pre-trained word-level embedding derived from large amounts of unlabeled dataset, on the other hand, have been widely applied to the biomedical domain and have proven the applicability and effectiveness on biomedical named entity recognition. We adopt a published set of 200-dimensional biomedical word embeddings [40] which was trained using the skip-gram algorithm [39] on 26 million abstracts and citations in PubMed.[5] This pre-trained word-level embedding maps each words to a dense real vector in an embedding matrix $W^{word} \in \mathbb{R}^{d^{word} \times |V^{word}|}$ where $|V^{word}|$ is the size of the vocabulary and d^{word} is the dimension of the word embedding. Each column $w_i^{word} \in \mathbb{R}^{d^{word}}$ corresponds to the word-level embedding of the ith word in the vocabulary.

Character-level embedding. This embedding has been used on many NLP tasks such as BNER [27,29,41] It provides additional information to the word-level embedding which captures typical morphological information from word. However, most of existing methods do not contribute to a significant increase in terms of the overall performance when they are concatenated with other feature embeddings. Thus, our method to compute character-level embedding takes into consideration all characters of the word since it involves and extracts the appropriate representation of word. To do this, we use a modified online recurrent extreme learning machine (OR-ELM). We first project each character in the corpus into a character embedding which is randomly initialized. We denote the character sequence of the word w_i with $\{c_1, c_2, \ldots, c_l\}$, where c_j corresponds to the jth character of w_i and l is the length of the word. Let $e_1^i, \ldots, e_l^i$ be the sequence of character embedding of w_i. Then, the character embeddings are fed into the OR-ELM through reading the word character by character from left to right to compute the vector embedding w_i^{char}. As shown in Fig. 2, upon the arrival the jth character embedding e_j^i, its corresponding hidden layer output matrix H_j is computed using Eq. (1).

$$H_j = g(norm(W_j e_j^i + V_j H_{j-1})) \tag{1}$$

where W_j is the OR-ELM's input weight, V_j is the OR-ELM's hidden weight, H_{j-1} is the OR-ELM's hidden layer output of the previous character embedding e_{j-1}^i, g(.) is a sigmoid activation function and norm(.) function indicates a LN procedure. Finally, the character-based vector embedding of the ith word is the last WOR-ELM' s hidden-layer output matrix H_l given by the Eq. (2).

$$w_i^{char} = H_l \tag{2}$$

3.1.3. Classification

After representing each word as a real vector by concatenating the corresponding word-level and character-level embeddings, these features are fed into WOR-ELM from left to right to learn and predict the representations of words in a sentence. The details of the WOR-ELM architecture for span detection stage are shown in the part (a) of Fig. 3. WOR-ELM takes advantage of OR-ELM [32] to train recurrent neural network by applying a LN procedure and ELM auto-encoder [34]. The LN procedure is used to improve the overall performance and overcome the internal covariate shift problem [42]. Similar to OR-ELM, WOR-ELM consists of two phases: initialization and sequential learning. The first one uses the fully online initialization method [43] while the second employs three networks: an RNN, which is the main network for the prediction, and two ELM-auto-encoder networks called ELM-AE-IW and ELM-AE-HW for learning RNN's input weights and hidden weights, respectively. To tackle the unbalanced tag distribution which a large number of words not belonging to any mention, WOR-ELM also exploits weighted online sequential extreme learning machine (WOS-ELM) [44] to update the WOR-ELM' s output weights. To do this, it integrates the weight value given by Eq. (3).

$$w_{k+1} = \frac{1}{s_{k+1}^j} \quad if \quad v_{k+1} \in S_{k+1}^j \tag{3}$$

where v_{k+1} is the feature vector, S_{k+1}^j represents a tag j at time step $k+1$ and s_{k+1}^j is the total number of words in tag j up till time step $k+1$. Then, the WOR-ELM' s output weight β_{k+1} is updated using Eq. (4).

$$\beta_{k+1} = \beta_k + P_{k+1} H_{k+1}{}^T w_{k+1} (M_{k+1} - H_{k+1}\beta_k) \tag{4}$$

where $Mk+1$ is the one-hot vector of the target tag and P_{k+1} is the auxiliary matrix which is used to compute the output weights β_{k+1} given by Eq. (5).

$$P_{k+1} = P_k - P_k H_{k+1}{}^T (w_{k+1}{}^{-1} + H_{k+1} P_k H_{k+1}{}^T)^{-1} H_{k+1} P_k \tag{5}$$

We summarize the sequential learning in Fig. 4 for the purposes of brevity. More details about the two phases can be found in [32]. Finally, The output layer calculates the distribution over all labels via the softmax function to select the one with the highest probability value.

3.2. ADE mentions classification

In ADE mentions classification stage, the mentions identified by the previous stage are classified into one of the two types,

[5] https://www.ncbi.nlm.nih.gov/pubmed/.

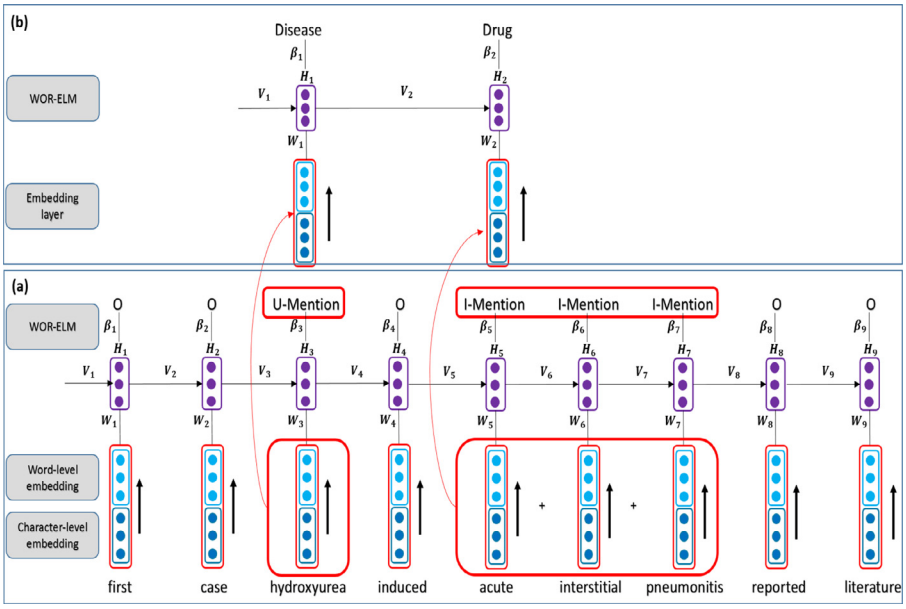

Fig. 3. The overall architecture of the proposed method where H_i, W_i, V_i and β_i are WOR-ELM's hidden layer, input weight, hidden weight and output weight, respectively.

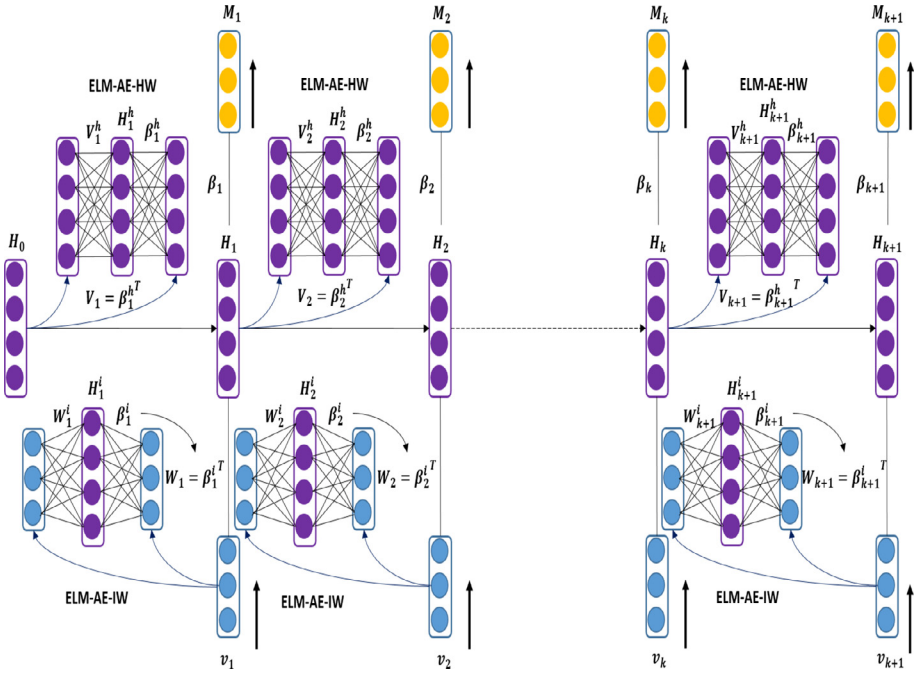

Fig. 4. The Sequential learning phase of WOR-ELM.

e.g., Diseases or Drugs. At this time, non-mention words (i.e., "O" tag) are ignored. For instance, in the sentence provided in the Section 3.1, the "*interstitial pneumonitis*" and "*hydroxyurea*" are identified as mentions by the first stage after post-processing, and the following Disease and Drug mention types can be assigned to them, respectively. The second stage consists also of two phases: features representation and classification with WOR-ELM.

3.2.1. Features representation

In the first phase of the ADE mentions classification stage, the word-level and character-level embeddings obtained by the previous stage are also used as input features for the identified mentions. Unlike span detection stage, the identified mentions are represented by summing all the contained word vectors whether they are a multi-word and by keeping the same features otherwise. In other words, let $b_i = \{w_1, w_2, \ldots, w_n\}$ be the ith identified mention by the first stage which consists of n words. Then, the corresponding feature vector is given by Eq. (6).

$$b_i = \begin{cases} \sum_{j=1}^{n} v_j, & \text{if } n > 1 \\ v_j, & \text{if } n = 1 \end{cases} \tag{6}$$

Recall that $v_j = [w_j^{word}, w_j^{char}]$ is the concatenation of word-level and character-level embeddings for the jth word in the vocabulary.

3.2.2. Classification

The selected features are fed into another WOR-ELM to classify the identified mentions to the appropriate ADE types, e.g., Diseases or Drugs. The details of the WOR-ELM architecture for the second

stage are shown in the part (b) of Fig. 3. The WOR-ELM's overall architecture is similar for both span detection and ADE mention classification stages. The only difference appears in the input and output layers where the first stage takes the vectors of words in the sentence as input and the IOU segment representation as output, while the second takes the vectors of the identified mentions as input and one of the two ADE mention types as output.

4. Experimental results

In this section, we first report our evaluation results of the proposed adverse drug effect mentions extraction method on the ADE corpus. We then compare our method with the current state-of-the-art methods presented and evaluated in [28,29].

4.1. Datasets

We evaluate our proposed method on the ADE corpus [35] which is distributed as separated sentences. The corpus is divided into two kinds of sentences that come from 2972 MEDLINE case reports: 6821 positive sentences which contain at least one ADE mention, and 16,695 negative ones, otherwise. Similarly to the experimental setting presented in [28,29], only the positive sentences are used in our evaluation and the sentences with overlapping mention are ignored.

4.2. Evaluation metrics

We evaluate the performance of the proposed method using precision (P), recall (R) and F-score (F1) defined in the standard Eq. (7), Eq. (8), and Eq. (9), respectively.

$$P = \frac{TP}{TP + FP} \tag{7}$$

$$R = \frac{TP}{TP + FN} \tag{8}$$

$$F1 = \frac{2 \times P \times R}{P + R} \tag{9}$$

where TP is the number of correct mentions that the system returns, FP is the number of incorrect mentions that the system returns, and FN is the number of missing ones.

4.3. Hyper-parameters setting

We adopt the implementation of OR-ELM[6] publicly available to develop our proposed method. We tune the hyper-parameters by conducting 10-fold cross-validation, where 10% of the data are used for the development, 10% for the test and 80% for training set. The WOR-ELM's hyper-parameters of the first stage are the same as that in the second stage of our proposed ADE mentions extraction method to simplify our research. The hyper-parameters include the hidden layer dimension of WOR-ELM, the word-level embedding dimension, the character-level embedding dimension and the regularization parameter. The hyper-parameters and their values used in our experiments are shown in Table 1. The size of the pre-trained word-level embedding is set to 200, while each character is randomly initialized with a 20-dimensional vector and fine-tuned during training. To improve the stability of WOR-ELM and prevent the overfitting issue, we apply the regularization parameter with 0.001. We set the character-level embedding dimension and the hidden layer dimension of WOR-ELM as 20 and 100, respectively.

[6] https://github.com/chickenbestlover/Online-Recurrent-Extreme-Learning-Machine.

Table 1
Hyper-parameters setting.

Parameter	Parameter name	Value
d^{word}	Word-embedding dimension	200
m_1	Character-embedding dimension	20
d^{char}	Character-level embedding dimension	20
L	Hidden layer dimension of WOR-ELM	100
C	Regularization parameter	0.001

Table 2
The effect of segments representation of the proposed method on the ADE corpus.

Segment	Direction	P (%)	R (%)	F1 (%)
IOU	Left-to-right	86.9	88.0	87.5
	Bidirectional	68.8	65.9	67.4
IO	Left-to-right	81.3	82.1	81.7
	Bidirectional	72.4	60.8	66.1
IOL	Left-to-right	85.4	80.7	83.0
	Bidirectional	49.0	22.4	30.8
BIO	Left-to-right	89.8	71.5	79.6
	Bidirectional	78.4	20.8	32.9
BILOU	Left-to-right	74.6	64.2	69.1
	Bidirectional	68.5	29.8	41.5

Table 3
The effect of different input representations of the proposed method on the ADE corpus.

Feature	P (%)	R (%)	F1 (%)
char	66.7	62.9	64.8
word	85.8	86.8	86.3
word+char	86.9	88.0	87.5

Table 4
Mention-level evaluation of the proposed method on the ADE corpus.

Feature	Mention	P (%)	R (%)	F1 (%)
char	Drugs	65.7	58.5	61.9
	Diseases	67.4	66.3	66.8
word	Drugs	94.3	90.8	92.5
	Diseases	79.3	83.3	81.3
word+char	Drugs	94.9	92.2	93.5
	Diseases	80.7	84.3	82.5

4.4. Results

To investigate the effect of segments representation of the proposed method, we perform a preliminary experiment on the ADE corpus. We trained our method by using two directions with five segments representation: IOU, IO, IOL, BIO and BILOU where B (Begin) represents the first word of the mention, I (Inside) indicates a part of the mention, L (Last) is the last word of the mention, O (Outside) represents a word that is not a part of the mention and U (Unit) is a single word mention. As shown in Table 2, the overall performance of the proposed method heavily depends on the appropriate choice of segments for representing multiword mentions which affects the final output and the learning process. Indeed, the IOU segment produced better results than all other ones whatever the direction used for span detection stage. The left-to-right direction provides better results in all segment while the IOU segment outperforms other ones by an average of 4.5% in terms of F-score.

We conduct several experiments to evaluate the effectiveness of the features adopted in our ADE mentions extraction method. We explore two features set, including word and character features. Then, we take the concatenation of them. Tables 3 and 4 show the effects of these features on the overall performance of the proposed method. It can clearly be seen from Table 3 that the concatenation of the two features allow our method to boost

Table 5
Comparison in terms of precision, recall and F-score of the proposed method with the current state-of-the-art ones on the ADE corpus.

Method	P (%)	R (%)	F1 (%)
Li et al. [28]	79.5	79.6	79.5
Li et al. [29]	82.7	86.7	84.6
Proposed method	86.9	88.0	87.5

its performance for ADE mentions extraction since the F-score increases by 1.2% when character-level embedding with a modified OR-ELM is added to the word-level embedding.

According to the results showed in Table 4, the contribution of the embeddings at mention-level is different. The character-level embedding has more effect in classification of diseases mentions than of drugs ones unlike other features combination. Moreover, the concatenation of word-level and character-level embeddings bring the performance enhancement of up to 1.2% and 1% in terms of F-score for the diseases and drugs mentions, respectively.

Table 5, on the other hand, shows a comparison in terms of precision, recall and F-score between our method and the current state-of-the-art ones on the ADE corpus presented in [28,29]. Li et al. [28] explored a deep neural system to jointly identify drug-related disease mentions from biomedical texts, achieving an F-score of 79.5%. Li et al. [29] adopted a Bi-LSTM network to identify ADE mentions from text and they reached an F-score of 84.6%.

As shown in Table 5, the proposed method achieves the best performance on the test set for adverse drug effect mentions extraction. Moreover, compared with the method proposed in [29] which appears to be the state-of-the-art in ADE extraction, our method gives better results (an average improvement of 3% in terms of F-score). The increased performance was statistically significant (the p-value is 0.0196, the result is significant at $p < 0.05$ and the deviation of precision, recall and F-score are 1.85, 0.63, and 0.96, respectively). In addition, we observe that our method has both higher precision and recall of up to 4.2% and 1.6%, respectively, while other methods have comparatively lower precision.

5. Discussion

In this paper, we presented an adverse drug effect mentions extraction method based on weighted online recurrent extreme learning machine to identify and classify drugs and diseases mentions from biomedical texts through using our character-level and the pretrained word embeddings.

5.1. Performance evaluation with other methods

The boundary of mention is more difficult to be identified which affect negatively the performance of such method. In fact, our method achieves good results by dividing it into two stages to handle this issue on the ADE corpus compared with other state-of-the-art methods. It gives better performance by training WOR-ELM without using additional features to correctly identify the boundary of mentions compared with the method proposed in [29] that requires researchers to integrate the last entity label as another feature to improve the overall performance by 2.5%. Indeed, the output sequence in [29] is represented by combining each tag with a mention type, which increases the number of features because of the expanded number of tags. Another advantage of the proposed method is the ability to identify the best sequence mention for given sentence in one direction from left to right. In fact, the inverse direction does not contribute to an increase in terms of the overall performance which affects negatively the bidirectional one. It means that the discriminant information of WOR-ELM is captured directly from the forward direction rather than backward one. Furthermore, our method can learn relevant features automatically by only incorporating both word-level and character-level embeddings that are tuned consequently during training the WOR-ELM.

5.2. Features evaluation

We investigated the effect of two types of features, including word-level and character-level embeddings for our adverse drug effect mention extraction method. Features evaluation showed that each type of features has a significant effect, with an F-score enhancement ranging from 1.8% to 3%. Our results also demonstrated that combining the two features further improved the overall performance, indicating that these features were complementary to each other. Therefore, the proposed character-level embedding with the modified OR-ELM contributes well when it is concatenated with word-level embedding. Indeed, compared with the method proposed in [29], which employs the CNN to learn character-level representation of the words, improving the performance by only 0.3% when it is concatenated with other feature embeddings, our proposed character-level embedding contribute to a significant increase in terms of F-score by an average of 1.2% with only the word-level feature. In fact, the modified OR-ELM seems to be able to project discriminant morphological features of word to a new space presented in the hidden layer since it incorporates ELM-AE and LN procedure. Indeed, ELM-AE proved its discriminative ability for representational learning [45] according to the universal approximation capability [46] and Johnson and Lindenstrauss [47] Lemmas. Hence, ELM-AE is used to extract important hidden layer features of the modified OR-ELM. Moreover, the LN procedure tends to improve the overall performance as it increases the stability and alleviate the internal covariate shift. As a result, the hidden layer of the modified OR-ELM provides an efficient features space which can be independent from the input character embedding and can prevent the overfitting problem. Moreover, the improvements by the two features varied among mention types, where the drug mentions achieved the highest improvement compared with disease ones. This is due to the word formation of disease mentions is much complex. For example, the disease mentions are often represented as the forms “like Bilateral acoustic (VIII) nerve palsy”, “(sub) fulminant hepatitise”, etc which contain letters and symbols while it appears infrequently for drug mentions.

5.3. Effect of segment representations

The segment representations are an efficient way to encode multi-word mention by assigning exactly one tag to each word. However, the obtained results showed that the inappropriate choice of segments affects negatively the overall performance of our ADE mentions extraction method. Indeed, the F-score tends to improve as the number of tags decrease where the IOL and IOU segments achieve the highest performance compared with BILOU segment. The main causes are as follow: first, the method leads to the tag sparsity problem because just a few part of data is named mention. Therefore, the use of a BILOU segment is not much effective for our method and bring performance degradation. Second, by using the BILOU segment, the error rate increase as a result of the ambiguity problem which a word can appear as either an entire or part of mentions in different contexts. For instance, given the following sentence “the use of MP developed transient renal failure following an MP pulse therapy”, compared with BILOU segment, the IOU segment tends to recognize “transient renal failure” as mention while BILOU segment just considers “renal failure” as mention since it appears as an entire one in other contexts. Therefore, the IOU segment will be most suitable for our proposed method.

Table 6
Errors rates of the proposed method on the ADE corpus.

Error type	rate (%)
FP	13.02
FN	11.97

5.4. Error analysis

To better illustrate the behavior of the proposed method, we examine the error rates, particularly false-positives (FP) and false-negatives (FN) presented in Table 6. The majority of errors in both FP and FN were caused by the span detection stage, which is very critical to the overall performance since the identification errors propagate into the ADE mentions classification one. The common causes are as follows: (1) integrity span detection errors, for instance, the true mention is “adriamycin toxicity”, our method tends to break this mention into two parts “adriamycin” and “toxicity”. This due to the corpus annotation incoherence: The true mention is annotated as an entire mention in some contexts and these components “adriamycin” and “toxicity” are annotated as multiple single mention in other ones. (2) Left boundary errors which are identified with wrong left boundary and correct right one. For instance, in the true mention “acute myopathy”, our method missed the adjective word “acute” and detected the mention as “myopathy”. (3) Missing spans which include the annotated mentions not matched with any gold ones. For example, some abbreviations, such as “TCA” and “LPD”, cannot be identified by our proposed method in some context and some general words (e.g. “syndrome” and “reaction”) are detected as mentions. In future work, we anticipate to integrate the contextual features to improve the overall result. In addition, the main error of the ADE mentions classification stage is caused by a misclassification of the disease mention to the drug mention due to the sense ambiguity and the lack of semantic knowledge source of mentions. In the future, we will reduce these errors by incorporating information from ontologies as additional features for our method.

6. Conclusion and future works

In this paper we proposed a weighted online recurrent extreme learning machine based method to extract adverse drug effect mentions from texts. The overall architecture of the proposed method is composed of two main stages: (1) span detection for identifying the boundary of the mentions irrespective of their types, and (2) ADE mentions classification for classifying the identified mentions to the appropriate type. In both stages, we combined character-level and word-level embeddings as features for WOR-ELM. The character-level embedding has been developed using the modified online recurrent extreme learning machine to capture morphological features. Our experimental results on the ADE corpus, a standard dataset, showed that the proposed method can be successfully used to detect and extract ADR mentions without any features engineering effort, and it demonstrates superior results as compared to the state-of-the-art methods in ADE mention extraction. We explored several segment representations in an attempt to encode multi-word mentions. Our experiments showed that significant improvements in classification accuracies can be achieved by using IOU segment. In the future, we intend to incorporate information from ontologies and metathesaurus such as UMLS as additional features for the weighted online recurrent extreme learning machine to improve the overall performance of the proposed method. We would also like to apply this method for the task of ADE relation extraction, which is the next step in our ADE monitoring pipeline. Considering that we achieved significantly better results than past research in the field, we expect that using the proposed method for the later stages of our work will produce similar results.

Conflicts of interest statement

None.

References

[1] S. Karimi, C. Wang, A. Metke-Jimenez, R. Gaire, C. Paris, Text and data mining techniques in adverse drug reaction detection, ACM Comput. Surv. 47 (2015) 1–39, doi:10.1145/2719920.

[2] Y. Ji, H. Ying, P. Dews, A. Mansour, J. Tran, R.E. Miller, R.M. Massanari, A potential causal association mining algorithm for screening adverse drug reactions in postmarketing surveillance, IEEE Trans. Inf. Technol. Biomed. 15 (3) (2011) 428–437, doi:10.1109/TITB.2011.2131669.

[3] E. Russo, C. Palleria, C. Leporini, S. Chimirri, G. Marrazzo, S. Sacchetta, L. Bruno, R. Lista, O. Staltari, A. Scuteri, F. Scicchitano, Limitations and obstacles of the spontaneous adverse drugs reactions reporting: two ”challenging” case reports, J. Pharmacol. Pharmacother. 4 (2013) 66, doi:10.4103/0976-500X.120955.

[4] C.L. Gurudatt, Case reports: brief overview of reporting and submission to biomedical journal, Indian J. Anaesth. 60 (2016) 695–699, doi:10.4103/0019-5049.190629.

[5] R. Harpaz, A. Callahan, S. Tamang, Y. Low, D. Odgers, S. Finlayson, K. Jung, P. LePendu, N.H. Shah, Text mining for adverse drug events: the promise, challenges, and state of the art, Drug Saf. 37 (2014) 777–790, doi:10.1007/s40264-014-0218-z.

[6] S.J. Athenikos, H. Han, Biomedical question answering: a survey, Comput. Methods Programs Biomed. 99 (1) (2010) 1–24, doi:10.1016/j.cmpb.2009.10.003.

[7] M. Sarrouti, S. Ouatik El Alaoui, A passage retrieval method based on probabilistic information retrieval and UMLS concepts in biomedical question answering, J. Biomed. Inform. 68 (2017) 96–103, doi:10.1016/j.jbi.2017.03.001.

[8] M. Sarrouti, S.O.E. Alaoui, A machine learning-based method for question type classification in biomedical question answering, Methods Inf. Med. 56 (03) (2017) 209–216, doi:10.3414/ME16-01-0116.

[9] M. Sarrouti, A. Lachkar, A new and efficient method based on syntactic dependency relations features for ad hoc clinical question classification, Int. J. Bioinform. Res. Appl. 13 (2) (2017) 161, doi:10.1504/IJBRA.2017.083150.

[10] M. Sarrouti, S.O.E. Alaoui, A yes/no answer generator based on sentiment-word scores in biomedical question answering, Int. J. Healthc. Inf.Syst. Inform. 13 (3) (2017) 12, doi:10.4018/IJHISI.2017070104.

[11] M. Sarrouti, S.O.E. Alaoui, A biomedical question answering system in BioASQ 2017, BioNLP 2017, Association for Computational Linguistics, 2017, doi:10.18653/v1/w17-2337.

[12] M. Sarrouti, S.O.E. Alaoui, A generic document retrieval framework based on UMLS similarity for biomedical question answering system, in: Intelligent Decision Technologies 2016, Springer International Publishing, 2016, pp. 207–216, doi:10.1007/978-3-319-39627-9_18.

[13] T. Alves, R. Rodrigues, H. Costa, M. Rocha, Development of an information retrieval tool for biomedical patents, Comput. Methods Programs Biomed. 159 (2018) 125–134, doi:10.1016/j.cmpb.2018.03.012.

[14] R.A.A. Seoud, M.S. Mabrouk, TMT-HCC: A tool for text mining the biomedical literature for hepatocellular carcinoma (HCC) biomarkers identification, Comput. Methods Programs Biomed. 112 (3) (2013) 640–648, doi:10.1016/j.cmpb.2013.07.014.

[15] E. El-allaly, M. Sarrouti, N. En-Nahnahi, S.O.E. Alaoui, Adverse drug reaction mentions extraction from drug labels: an experimental study, in: Advanced Intelligent Systems for Sustainable Development (AI2SD'2018) Vol 4: Advanced Intelligent Systems Applied to Health, vol. 914, Springer International Publishing, 2019, pp. 216–231, doi:10.1007/978-3-030-11884-6_21.

[16] D. Campos, S. Matos, J.L. Oliveira, Biomedical named entity recognition: a survey of machine-learning tools, Theory Appl. Adv. Text Min.g (2012) 175–195, doi:10.5772/51066.

[17] T.C. Rindflesch, L. Tanabe, J.N. Weinstein, L. Hunter, EDGAR: extraction of drugs, genes and relations from the biomedical literature., Pac. Symp. Biocomput. (2000) 517–528, doi:10.1016/j.bbi.2008.05.010.

[18] M. Song, H. Yu, W.S. Han, Developing a hybrid dictionary-based bio-entity recognition technique, BMC Med. Inform. Decis. Mak. 15 (2015) S9, doi:10.1186/1472-6947-15-S1-S9.

[19] Q.T. Zeng, S. Goryachev, S. Weiss, M. Sordo, S.N. Murphy, R. Lazarus, Extracting principal diagnosis, co-morbidity and smoking status for asthma research: evaluation of a natural language processing system, BMC Med. Inform. Decis. Mak. 6 (2006) 1–9, doi:10.1186/1472-6947-6-30.

[20] D. Hanisch, K. Fundel, H.T. Mevissen, R. Zimmer, J. Fluck, ProMiner: rule-based protein and gene entity recognition, BMC Bioinform. 6 (SUPPL.1) (2005) 1–9, doi:10.1186/1471-2105-6-S1-S14.

[21] K. Takeuchi, N. Collier, Bio-medical entity extraction using support vector machines, Artif. Intell. Med. 33 (2005) 125–137, doi:10.1016/j.artmed.2004.07.019.

[22] S. Zhao, Named entity recognition in biomedical texts using an HMM model, in: Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications - JNLPBA '04, 2004, p. 84, doi:10.3115/1567594.1567613.

[23] B. Settles, Biomedical named entity recognition using conditional random fields and rich feature sets, in: The International Joint Workshop, 2004, pp. 104–107, doi:10.3115/1567594.1567618.
[24] J.L. Elman, Finding structure in time, Cognit. Sci. 14 (1990) 179–211, doi:10.1207/s15516709cog1402_1.
[25] S. Hochreiter, J.J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1–32, doi:10.1162/neco.1997.9.8.1735.
[26] H. Fabregat, L. Araujo, J. Martinez-Romo, Deep neural models for extracting entities and relationships in the new RDD corpus relating disabilities and rare diseases, Comput. Methods Programs Biomed. 164 (2018) 121–129, doi:10.1016/j.cmpb.2018.07.007.
[27] C. Lyu, B. Chen, Y. Ren, D. Ji, Long short-term memory RNN for biomedical named entity recognition, BMC Bioinform. 18 (2017) 462, doi:10.1186/s12859-017-1868-5.
[28] F. Li, Z. Yue, Z. Meishan, D. Ji, Joint models for extracting adverse drug events from biomedical text, in: IJCAI International Joint Conference on Artificial Intelligence, 2016, pp. 2838–2844.
[29] F. Li, M. Zhang, G. Fu, D. Ji, A neural joint model for entity and relation extraction from biomedical text, BMC Bioinform. 18 (2017) 198, doi:10.1186/s12859-017-1609-9.
[30] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: a new learning scheme of feedforward neural networks, in: IEEE International Joint Conference on Neural Networks, vol. 2, IEEE, 2004, pp. 985–990, doi:10.1109/IJCNN.2004.1380068.
[31] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (2006) 489–501, doi:10.1016/J.NEUCOM.2005.12.126.
[32] J.M. Park, J.H. Kim, Online recurrent extreme learning machine and its application to time-series prediction, in: Proceedings of the International Joint Conference on Neural Networks, 2017-May, 2017, pp. 1983–1990, doi:10.1109/IJCNN.2017.7966094.
[33] N.-Y. Liang, G.-B. Huang, P. Saratchandran, N. Sundararajan, A fast and accurate online sequential learning algorithm for feedforward networks, IEEE Trans. Neural Netw. 17 (2006) 1411–1423, doi:10.1109/TNN.2006.880583.
[34] L.L.C. Kasun, H. Zhou, G.-B. Huang, C. Vong, Representational learning with extreme learning machine for big data, IEEE Intell. Syst. (2013) 1–4, doi:10.1109/MIS.2013.140.
[35] H. Gurulingappa, A.M. Rajput, A. Roberts, J. Fluck, M. Hofmann-Apitius, L. Toldo, Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports, J. Biomed. Inform. 45 (2012) 885–892, doi:10.1016/j.jbi.2012.04.008.
[36] H. Gurulingappa, A. MateenRajput, L. Toldo, Extraction of potential adverse drug events from medical case reports, J. Biomed. Semant. 3 (2012) 15, doi:10.1186/2041-1480-3-15.
[37] C. Giuliano, A. Lavelli, D. Pighin, L. Romano, Fbk-irst: Kernel methods for semantic relation extraction, in: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007), Association for Computational Linguistics, 2007, pp. 141–144.
[38] N. Kang, B. Singh, C. Bui, Z. Afzal, E.M. van Mulligen, J.A. Kors, Knowledge-based extraction of adverse drug events from biomedical text, BMC Bioinform. 15 (2014) 64, doi:10.1186/1471-2105-15-64.
[39] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, CoRR (2013).
[40] S. Pyysalo, F. Ginter, H. Moen, T. Salakoski, S. Ananiadou, Distributional semantics resources for biomedical text processing, in: Proceedings of LBM 2013, 2013, pp. 39–44.
[41] M. Gridach, Character-level neural network for biomedical named entity recognition, J. Biomed. Inform. 70 (2017) 85–91, doi:10.1016/j.jbi.2017.05.002.
[42] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, ICML, 2015.
[43] P.K. Wong, C.M. Vong, X.H. Gao, K.I. Wong, Adaptive control using fully online sequential-extreme learning machine and a case study on engine air-fuel ratio regulation, Math. Probl. Eng. 2014 (2014), doi:10.1155/2014/246964.
[44] B. Mirza, Z. Lin, K.-A. Toh, Weighted online sequential extreme learning machine for class imbalance learning, Neural Process. Lett. 38 (3) (2013) 465–486, doi:10.1007/s11063-013-9286-9.
[45] L.L.C. Kasun, Y. Yang, G.-B. Huang, Z. Zhang, Dimension reduction with extreme learning machine, IEEE Trans. Image Process. 25 (8) (2016) 3906–3918, doi:10.1109/tip.2016.2570569.
[46] G.-B. Huang, L. Chen, C.-K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes, IEEE Trans. Neural Netw. 17 (4) (2006) 879–892, doi:10.1109/tnn.2006.875977.
[47] W.B. Johnson, J. Lindenstrauss, Extensions of Lipschitz mappings into a hilbert space, Contemp. Math. (1984), doi:10.1090/conm/026/737400.