

# Encoder/decoder mechanism, from machine translation to named entity recognition

Jingbo Xia

Huazhong Agricultural University

*xiajingbo.math@gmail.com*

March 9, 2019

# Table of contents I

1	RNN encoder/decoder mechanism and Attention mechanism	4
	• RNN encoder/decoder mechanism for machine translation	6
	• Attention mechanism: Bengio's new RNN e/d method for machine translation	8
2	Baseline methods for NER	15
	• Baseline method 1: Bi-LSTM-CRF Model for NER (Huang2015)	16
	• Baseline method 2: Bi-LSTM-ED Model for NER (Zheng2017)	17
	• Baseline method 3: GRU-Attention Model for NER (Liu2016)	18
3	Proposed Algorithm of Bi-LSTM-AEP for NER	19
4	Application of Bi-LSTM-AEP: A case study in trigger word detection	27


# Introduction

In this section, we introduced the basic idea of encoder/decoder mechanism, which was previously set up for machine learning. Later, Bengio gave a nice integration of attention and made Attention very popular since 2017.

Later, NER was achieved by using encoder/decoder mechanism. Two baseline without Attention and one baseline with Attention were presented subsequently.

This material is mainly to report an accepted work in CCL 2018 <sup>1</sup> .

---

<sup>1</sup>Kaiyin Zhou, Xinzhi Yao, Shuguang Wang, Jin-Dong Kim, Kevin Bretonnel Cohen, Ruiying Chen, Yuxing Wang and Jingbo Xia\*. Trigger Words Detection by Integrating Attention Mechanism into Bi-LSTM Neural Network — A Case study in PubMed-wide Trigger Words Detection for Pancreatic Cancer. CCL & NLP-NABD 2018. LNAI 11221, pp. 398-409, 2018. 

- 1 RNN encoder/decoder mechanism and Attention mechanism 4
  - RNN encoder/decoder mechanism for machine translation 6
  - Attention mechanism: Bengio's new RNN e/d method for machine translation 8
- 2 Baseline methods for NER 15
  - Baseline method 1: Bi-LSTM-CRF Model for NER (Huang2015) 16
  - Baseline method 2: Bi-LSTM-ED Model for NER (Zheng2017) 17
  - Baseline method 3: GRU-Attention Model for NER (Liu2016) 18
- 3 Proposed Algorithm of Bi-LSTM-AEP for NER 19
- 4 Application of Bi-LSTM-AEP: A case study in trigger word detection 27

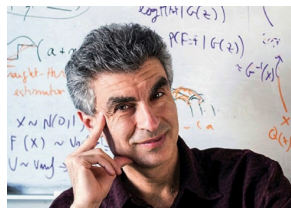
# RNN encoder/decoder mechanism and Attentionism

We borrow KyungHyun Cho and Yoshua Bengio's Encoder/Decoder illustration here first <sup>2</sup>, this is the first one stating e/d mechanism. Furthermore, Dzmitry Bahdanau and Yoshua Bengio presented clear description for RNN encoder and decoder mechanism <sup>3</sup>

In addition, Bengio's Attention mechanism is introduced in this paper. We are collecting these results in is section.



KyungHyun Cho



Yoshua Bengio

<sup>2</sup>Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

<sup>3</sup>Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473. 2014 Sep 1.

# RNN encoder/decoder mechanism for machine translation

In the Encoder-Decoder framework, an encoder reads the input sentence, a sequence of vectors  $x = (x_1, \dots, x_{T_x})$ , into a vector  $c$ . The most common approach is to use an RNN such that

$$h_t = f(x_t, h_{t-1})$$

and

$$c = q(\{h_1, \dots, h_{T_x}\}),$$

where  $h_t$  is hidden layer, and  $c$  is a vector generated from the sequence of the hidden states.  $f$  and  $q$  are some nonlinear functions.

# RNN encoder/decoder mechanism for machine translation

The decoder is often trained to predict the next word  $y_{t'}$  given the context vector  $c$  and all the previously predicted words  $\{y_1, \dots, y_{t'-1}\}$ . In other words, the decoder defines a probability over the translation  $y$  by decomposing the joint probability into the ordered conditionals:

$$p(\{y_1, \dots, y_{T_y}\}) = \prod_{t=1}^{T_y} p(y_t | \{y_1, \dots, y_{t-1}\}, c) \quad (1)$$

With an RNN, each conditional probability is modeled as

$$p(y_t | \{y_1, \dots, y_{t-1}\}, c) = g(y_{t-1}, s_t, c). \quad (2)$$

# Attention mechanism for machine translation

In a new model, Bengio et al define each conditional prob. in equation (1) as

$$p(y_i | y_1, \dots, y_{i-1}) = g(y_{i-1}, s_i, c_i) \quad (3)$$

where  $s_i$  is an RNN hidden state for time  $i$ , computed by

$$s_i = f(s_{i-1}, y_{i-1}, c_i). \quad (4)$$

Here the probability is conditioned on a distinct context vector  $c_i$  for each target word  $y_i$ .

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j, \quad (5)$$

where  $(h_1, \dots, h_{T_x})$  is a sequence of annotations, and weight  $\alpha_{ij}$  of each annotation  $h_j$  is:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}. \quad (6)$$

where  $e_{ij} = a(s_{i-1}, h_j)$  is an alignment model which scores how well the inputs around position  $i$  and output at position  $j$  match.



# Attention mechanism for machine translation

Input:  $(x_1, \dots, x_{x_T})$ , e.g., English;  
Output:  $(y_1, \dots, y_{y_T})$ , e.g., Francais.

$$p(y_i | y_1, \dots, y_{i-1}) = g(y_{i-1}, s_i, c_i) \quad (3)$$

$$s_i = f(s_{i-1}, y_{i-1}, c_i). \quad (4)$$

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j, \quad (5)$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}. \quad (6)$$

$e_{ij} = a(s_{i-1}, h_j)$ , to score how well  $x_j$  (e.g., Love) and  $y_i$  (Amour) match.

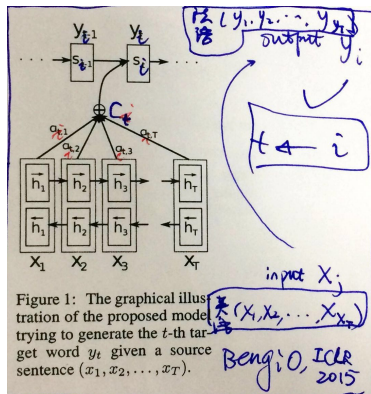


Figure 1: Bengio, Attention2015

# Attention mechanism for machine translation

The **new state**  $s_i$  of the RNN employing  $n$  gated hidden units is computed by equation (4), whose details are:

$$s_i = f(s_{i-1}, y_{i-1}, c_i) = (1 - z_i) \circ s_{i-1} + z_i \circ \tilde{s}_i \quad (7)$$

where **update state**  $\tilde{s}_i$  is

$$\tilde{s}_i = \tanh(W e(y_{i-1}) + U[r_i \circ s_{i-1}] + C c_i) \in (-1, 1)$$

where  $e(y_{i-1}) \in \mathbb{R}^m$  is a word embedding of a word  $y_{i-1}$ ,  $r_i \in (0, 1)$  is output of **reset gate**, which control how much and what information from the previous state  $s_{i-1}$  should be reset:

$$r_i = \sigma(W_r e(y_{i-1}) + U_r s_{i-1} + C_r c_i). \in (0, 1)$$

$\sigma$  is sigmoid activation function,  $e(y_{i-1})$  is treated as word embedding of French output,  $c_i$  is treated as attention of embedding of English input (from equation (5)),  $s_{i-1}$  is hidden layer of RNN, which also convey the word embedding of history info. **Update gate**  $z_i \in (0, 1)$  allows each hidden units maintain its previous activation.

# Attention mechanism for machine translation

So, if one wanna convey the idea of Attention mechanism from translation to NER, the symbol which need to change is the word embedding of  $y_{i-1}$ , i.e.,  $e(y_{i-1})$ . (So, this is the suggestion I gave to Kaiyin today. 2018/5/30. )

Two things that I omitted just now. First, **update gate**  $z_i$  is:

$$z_i = \sigma(W_z e(y_{i-1}) + U_z s_{i-1} + C_z c_i). \quad \in (0, 1)$$

While alignment of  $x_j$  and  $y_i$  is computed by **alignment model**:

$$e_{i,j} = a(s_{i-1}, h_j) = v_a^T \tanh(W_a s_{i-1} - U_a h_j).$$

Here we assume an expected match of  $x_j$  and  $y_i$  suffices to make a zero-approximate of linear combination of  $s_{i-1}$  and the  $j$ -th annotation of the sentence,  $h_j$ ?

# Attention mechanism for machine translation

## Detailed algorithm

ENCODER detail:

Input:  $x = (x_1, \dots, x_{T_x}), \vec{h}_0 = 0, \overleftarrow{h}_0 = 0,$

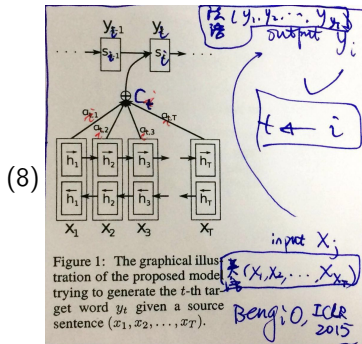
$$\vec{h}_i = \begin{cases} (1 - \vec{z}_i) \circ \vec{h}_{i-1} + \vec{z}_i \circ \vec{h}_i, & \text{if } i > 0 \\ 0, & \text{if } i = 0 \end{cases}$$

$$\vec{h}_i = \tanh(\vec{W} E x_i + \vec{U} [\vec{r}_i \circ \vec{h}_{i-1}])$$

$$\vec{z}_i = \sigma(\vec{W}_z + \vec{U}_z \vec{h}_{i-1}),$$

$$\vec{r}_i = \sigma(\vec{W}_r + \vec{U}_r \vec{h}_{i-1}),$$

Output:  $h_i = [\vec{h}_i, \overleftarrow{h}_i]$



# Attention mechanism for machine translation

## Detailed algorithm

### DECODER detail:

Input:  $\{h_1, h_2, \dots, h_{T_x}\}, y_{i-1}$

$$s_i = f(s_{i-1}, y_{i-1}, c_i) = (1 - z_i) \circ s_{i-1} + z_i \circ \tilde{s}_i$$

$$\tilde{s}_i = \tanh(W e(y_{i-1}) + U[r_i \circ s_{i-1}] + C c_i)$$

$$r_i = \sigma(W_r e(y_{i-1}) + U_r s_{i-1} + C_r c_i).$$

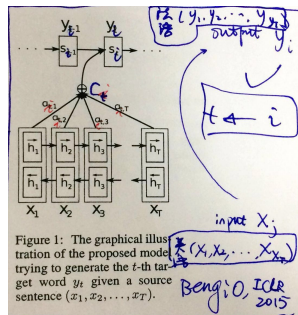
$$z_i = \sigma(W_z e(y_{i-1}) + U_z s_{i-1} + C_z c_i).$$

Attention output:  $c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$ ,

$$\alpha_{ij} = \text{softmax}(\vec{e}_i)_j = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}.$$

$$e_{i,j} = a(s_{i-1}, h_j) = v_a^T \tanh(W_a s_{i-1} - U_a h_j).$$

(9)



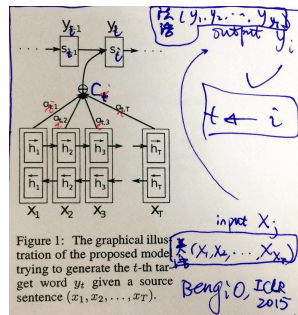
# Attention mechanism for machine translation

## Detailed algorithm

Prob approximation detail:

With the decoder state  $s_{i-1}$ , the context  $c_i$  and the last generated word  $y_{i-1}$ , we define the probability of a target word  $y_i$  as:

$$p(y_i | s_i, y_{i-1}, c_i) \propto \exp(y_i^T W_a t_i) \quad (10)$$
$$t_i = [\max\{\tilde{t}_{i,2j-1}, \tilde{t}_{i,2j}\}]_{j=1, \dots, l}^T$$
$$\tilde{t}_i = U_o s_{i-1} + V_o E y_{i-1} + C_o c_i$$



# Outline

- 1 RNN encoder/decoder mechanism and Attention mechanism 4
  - RNN encoder/decoder mechanism for machine translation 6
  - Attention mechanism: Bengio's new RNN e/d method for machine translation 8
- 2 **Baseline methods for NER** 15
  - Baseline method 1: Bi-LSTM-CRF Model for NER (Huang2015) 16
  - Baseline method 2: Bi-LSTM-ED Model for NER (Zheng2017) 17
  - Baseline method 3: GRU-Attention Model for NER (Liu2016) 18
- 3 Proposed Algorithm of Bi-LSTM-AEP for NER 19
- 4 Application of Bi-LSTM-AEP: A case study in trigger word detection 27

# Baseline method 1: Bi-LSTM-CRF Model

Bi-LSTM-CRF (Huang2015)<sup>4</sup> is a typical neural network model used in sequence labeling tasks. It carried on LSTM training with the data for two times, and the only difference for each time was that the order of the two times input data was completely reversed, then the results of each LSTM layer were concatenated as an output of words encoding results. Thus, Bi-LSTM model captured both the past and the future information respectively. Then, the output vectors of Bi-LSTM were fed to the CRF layer to jointly decode the best label sequence. This model has been proved to be reasonable and has achieved state-of-art scores on many sequence labeling tasks.

---

<sup>4</sup>Huang Z, Xu W, Yu K. Bidirectional LSTM-CRF models for sequence tagging[J]. arXiv preprint arXiv:1508.01991, 2015.



## Baseline method 2: Bi-LSTM-ED Model

Bi-LSTM-ED (Zheng 2017)<sup>5</sup> is another model used to carry on sequence labeling tasks. This model still used Bi-LSTM as encoding layers. Being different from other models, the decode layer in this model was a Variant LSTM. The units of the decoding LSTM were the same as the encoding LSTM except for the input gate, which was replaced by

$$\begin{cases} i_i = \sigma(W_{ii}E_{xi} + U_{ii}h_{i-1} + V_{ii}T_{i-1} + b_{ii}) \\ T_i = W_{is}h_i + b_{is} \end{cases} . \quad (11)$$

This model obtained good improvement in several sequence labeling tasks.

---

<sup>5</sup>Zheng S, Hao Y, Lu D, et al. Joint Entity and Relation Extraction Based on A Hybrid Neural Network[J]. Neurocomputing, 2017, 257(000):1-8.

## Baseline method 3: GRU-Attention Model

In past few years, GRU-Attention model <sup>6</sup> has been widely used in speech recognition, image caption generation, visual question answering, machine translation and other fields, while few people apply it to sequence labeling tasks. In 2016, Liu <sup>7</sup> converted it into a NER-purposed one, where  $s_i$  in encoder layer was computed by a GRU unit,

$$s_i = GRU(h_i, s_{i-1}, c_i), \quad (12)$$

where the detailed formulas are as below:

$$\begin{cases} s_i = GRU(h_i, s_{i-1}, c_i) = (1 - z_i) \circ s_{i-1} + z_i \circ \tilde{s}_i \\ \tilde{s}_i = \tanh(W h_{i-1} + U[r_i \circ s_{i-1}] + C c_i) \\ z_i = \sigma(W_z h_i + U_z s_{i-1} + C_z c_i) & \text{[update gate: } gate_{update}] \\ r_i = \sigma(W_r h_i + U_r s_{i-1} + C_r c_i) & \text{[reset gate: } gate_{reset}] \end{cases} \cdot \quad (13)$$

---

<sup>6</sup>Bahdanau D, Cho K, Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate[J]. Computer Science, 2014.

<sup>7</sup>Liu B, Lane I. Attention-based recurrent neural network models for joint intent detection and slot filling. arXiv preprint arXiv:1609.01454. 2016 Sep 6.

# Outline

- 1 RNN encoder/decoder mechanism and Attention mechanism 4
  - RNN encoder/decoder mechanism for machine translation 6
  - Attention mechanism: Bengio's new RNN e/d method for machine translation 8
- 2 Baseline methods for NER 15
  - Baseline method 1: Bi-LSTM-CRF Model for NER (Huang2015) 16
  - Baseline method 2: Bi-LSTM-ED Model for NER (Zheng2017) 17
  - Baseline method 3: GRU-Attention Model for NER (Liu2016) 18
- 3 Proposed Algorithm of Bi-LSTM-AEP for NER 19
- 4 Application of Bi-LSTM-AEP: A case study in trigger word detection 27

# Proposed Algorithm of Bi-LSTM-AEP for NER

A Bi-LSTM based encode/decode mechanism for named entity recognition was studied in this research.

In the proposed mechanism, Bi-LSTM was used for encoding, an Attention method was used in the intermediate layers, and an unidirectional LSTM was used as decoder layer. By using element wise product to modify the conventional decoder layers, the proposed model achieved better F-score, compared with other three baseline LSTM-based models.

For the purpose of algorithm application, a case study of causal gene discovery in terms of disease pathway enrichment was designed. In addition, the causal gene discovery rate of our proposed method was compared with another baseline methods. The result showed that trigger genes detection effectively increase the performance of a text mining system for causal gene discovery.

# Proposed Algorithm of Bi-LSTM-AEP for NER

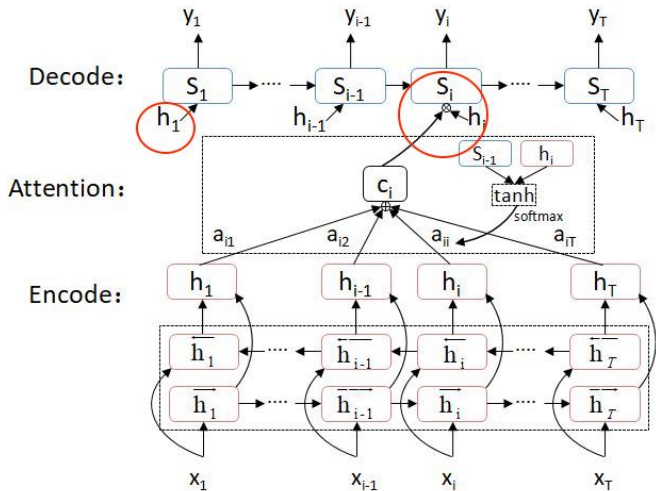


Figure 2: Network structure of Bi-LSTM-AEP

# Proposed Algorithm of Bi-LSTM-AEP for NER

- **Encode layer:**

Since we were desired to take both the last word and the next word into consideration, Bi-LSTM was employed as the encode layer, which contains a forward LSTM and a backward LSTM. At first, forward LSTM and backward LSTM read the words in a sentence  $x = (x_1, x_2, \dots, x_{T_x})$  respectively, and calculates the hidden states of each word:  $(\overrightarrow{h_1}, \overrightarrow{h_2}, \dots, \overrightarrow{h_{T_x}})$  and  $(\overleftarrow{h_1}, \overleftarrow{h_2}, \dots, \overleftarrow{h_{T_x}})$ .

Then the forward hidden states and the backward hidden states were combined in the third dimension as the annotation for each word:

$h_j = [\overrightarrow{h_j}, \overleftarrow{h_j}]$ . Therefore, annotation  $h_j$  represents not only the  $x_j$  itself but also the context information around it.

- **Attention-Mechanism:**

After encoding, the input words were transformed to annotation  $h$ . If attention mechanism was not taken into consideration, the annotation  $h$  for each input words (from  $h_1$  to  $h_j$ ) would be combined directly as a context vector which contained all the information in the sentence equally, and then the context vector was one of the inputs to decode layer. However, the importance of each annotation  $h$  should be different, so we introduced attention mechanism to calculate the different weight for each input by scoring to the alignment of input at position  $j$  and output at  $i$ .

# Proposed Algorithm of Bi-LSTM-AEP for NER

- At this part, the input is annotation  $h$  for each words in the sentence and the hidden state at last position of output sequence  $s_{i-1}$ , and the output is the context vector after considering the weight for each annotation  $h$ . The formulas of attention mechanism are shown below:

$$\left\{ \begin{array}{l} c_i = \sum_{j=1}^{T_x} a_{ij} h_j \\ a_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} \\ e_{ij} = V_a^T \tanh(W_a s_{i-1} + U_a h_j) \end{array} \right. , \quad (14)$$

where  $a_{ij}$  is the weight for different annotation  $h$ ,  $c_i$  is the context vector, and  $e_{ij}$  is the alignment scores.



# Proposed Algorithm of Bi-LSTM-AEP for NER

- Decode-layer: The decode layer is a unidirectional LSTM. In previous models, the output of encode results  $h = (h_1, h_2 \dots h_i, \dots h_{T_x})$  and the context vector  $c_i$  from attention mechanism were added at each time step  $s_i = GRU(h_i, s_{i-1}, c_i)$ . However, we proposed our decode-layer

$$s_i = LSTM(h_i * c_i, s_{i-1}), \quad (15)$$

where  $*$  is the element-wise product. The complete specific formulas are:

$$\left\{ \begin{array}{l} i_i = \sigma(W_{ei}[h_i * c_i] + U_{si}s_{i-1}) \\ f_i = \sigma(W_{ef}[h_i * c_i] + U_{sf}s_{i-1}) \\ z_i = \tanh(W_{ez}[h_i * c_i] + U_{sz}s_{i-1}) \\ \tilde{z}_i = f_i * \tilde{z}_{i-1} + i_i * z_i \\ o_i = \sigma(W_{eo}[h_i * c_i] + U_{so}s_{i-1}) \\ s_i = LSTM(h_i * c_i, s_{i-1}) = o_i \tanh(\tilde{z}_i) \end{array} \right. \quad (16)$$

For simplicity, the bias terms were omitted in the above formulas.

# Proposed Algorithm of Bi-LSTM-AEP for NER

Performance comparison

Method	Precision	Recall rate	F1-measure
Bi-LSTM-AEP (Ours)	<u>0.7576</u>	<u>0.4171</u>	<u>0.5160</u>
Bi-LSTM-Attention (Liu2016)	0.7092	0.3286	0.4368
Bi-LSTM-ED (zheng2017)	0.6604	0.3249	0.4263
Bi-LSTM-CRF (Huang2015)	0.5849	0.3051	0.3947

Data: AGAC corpus. Two class trigger detection task.

# Outline

- 1 RNN encoder/decoder mechanism and Attention mechanism 4
  - RNN encoder/decoder mechanism for machine translation 6
  - Attention mechanism: Bengio's new RNN e/d method for machine translation 8
- 2 Baseline methods for NER 15
  - Baseline method 1: Bi-LSTM-CRF Model for NER (Huang2015) 16
  - Baseline method 2: Bi-LSTM-ED Model for NER (Zheng2017) 17
  - Baseline method 3: GRU-Attention Model for NER (Liu2016) 18
- 3 Proposed Algorithm of Bi-LSTM-AEP for NER 19
- 4 Application of Bi-LSTM-AEP: A case study in trigger word detection 27

# Application of Bi-LSTM-AEP: A case study in trigger word detection

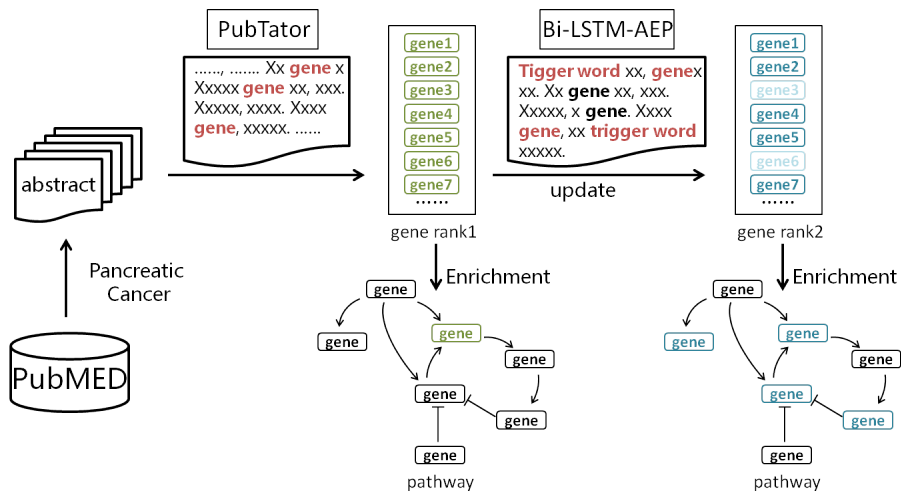


Figure 3: A Case study in trigger words detection in terms of pancreatic cancer.

# Application of Bi-LSTM-AEP —CONT

To compare the accuracy of the above mentioned two methods, Pubtator and Bi-LSTM-AEP. The result of the comparison is shown in Table 1.

	Extracted terms	Extracted pathway genes	Accuracy
Method 1(Pubtator)	28336	54	0.19%
Method 2(Bi-LSTM-AEP)	11675	52	0.45%

Table 1: The result of the comparison

# Application of Bi-LSTM-AEP —CONT

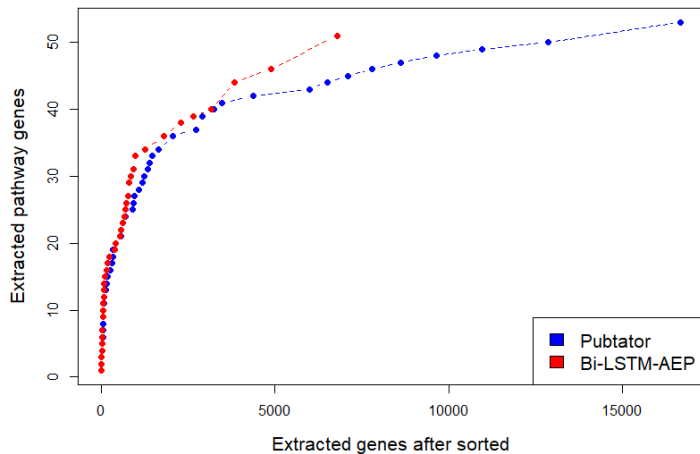


Figure 4: Distribution of KEGG Pathway Genes in the Results of Two Methods



감사합니다 Natick  
Grazie Danke Ευχαριστίες Dalu  
Thank You Köszönöm  
Tack  
Спасибо Dank Gracias  
谢谢 Merci Seé  
ありがとう Obrigado

Thank you!