

Original Paper

Novel Graph-Based Model With Biaffine Attention for Family History Extraction From Clinical Text: Modeling Study

Kecheng Zhan¹, MA; Weihua Peng², PhD; Ying Xiong¹, PhD; Huhao Fu¹, MA; Qingcai Chen^{1,3}, PhD; Xiaolong Wang¹, PhD; Buzhou Tang^{1,3}, PhD

¹Key Laboratory of Network Oriented Intelligent Computation, Harbin Institute of Technology, Shenzhen, China

²Baidu International Technology (Shenzhen) Co, Ltd, Shenzhen, China

³Peng Cheng Laboratory, Shenzhen, China

Corresponding Author:

Buzhou Tang, PhD

Key Laboratory of Network Oriented Intelligent Computation

Harbin Institute of Technology

University Town

Shenzhen, 518055

China

Phone: 86 13725525983

Email: tangbuzhou@hit.edu.cn

Abstract

Background: Family history information, including information on family members, side of the family of family members, living status of family members, and observations of family members, plays an important role in disease diagnosis and treatment. Family member information extraction aims to extract family history information from semistructured/unstructured text in electronic health records (EHRs), which is a challenging task regarding named entity recognition (NER) and relation extraction (RE), where named entities refer to family members, living status, and observations, and relations refer to relations between family members and living status, and relations between family members and observations.

Objective: This study aimed to introduce the system we developed for the 2019 n2c2/OHNLP track on family history extraction, which can jointly extract entities and relations about family history information from clinical text.

Methods: We proposed a novel graph-based model with biaffine attention for family history extraction from clinical text. In this model, we first designed a graph to represent family history information, that is, representing NER and RE regarding family history in a unified way, and then introduced a biaffine attention mechanism to extract family history information in clinical text. Convolution neural network (CNN)-Bidirectional Long Short Term Memory network (BiLSTM) and Bidirectional Encoder Representation from Transformers (BERT) were used to encode the input sentence, and a biaffine classifier was used to extract family history information. In addition, we developed a postprocessing module to adjust the results. A system based on the proposed method was developed for the 2019 n2c2/OHNLP shared task track on family history information extraction.

Results: Our system ranked first in the challenge, and the F1 scores of the best system on the NER subtask and RE subtask were 0.8745 and 0.6810, respectively. After the challenge, we further fine tuned the parameters and improved the F1 scores of the two subtasks to 0.8823 and 0.7048, respectively.

Conclusions: The experimental results showed that the system based on the proposed method can extract family history information from clinical text effectively.

(*JMIR Med Inform* 2021;9(4):e23587) doi: [10.2196/23587](https://doi.org/10.2196/23587)

KEYWORDS

family history information; named entity recognition; relation extraction; deep biaffine attention

Introduction

Family history information plays an important role in the diagnosis and treatment of diseases, especially genetic disorders.

Family history information is always embedded in electronic health records (EHRs) in a semistructured/unstructured format, which needs to be unlocked by natural language processing (NLP) technology.

In order to promote research on family history information extraction, Harvard Medical School and Mayo Clinic organized national NLP challenges on family history information extraction in 2018 and 2019. The family history information extraction task includes the following two subtasks: (1) recognizing family members, living status, and observations and (2) determining which family members the recognized living status and observations belong to, which correspond to two fundamental NLP tasks, namely named entity recognition (NER) and relation extraction (RE). The NER task is usually regarded as a sequence labeling task, while the RE task is the subsequent classification task, and they are tackled by pipeline methods.

For the NER task, traditional machine learning methods, such as hidden Markov model (HMM), conditional random field (CRF) [1], and structured support vector machine (SSVM) [2], and deep learning methods, such as Bidirectional Long Short Term Memory network (BiLSTM) CRF [3] and its variants [4,5], have been widely applied. For the RE task, the typical machine learning methods include support vector machine (SVM) [6], convolutional neural network (CNN) [7], and recurrent neural network [8]. The methods mentioned above have also been applied for clinical entity recognition and RE, such as the NLP challenges organized by i2b2 in 2009 [9], 2010 [10], 2012 [11], and 2014 [12], the NLP challenges organized by SemEval in 2015 [13] and 2016 [14], the NLP challenges organized by ShARe/CLEF in 2013 [15] and 2014 [16], and the NLP challenges organized by BioCreative/OHNLP in 2018 [17]. Most of these methods process NER and RE tasks in a pipeline way, which can suffer from error propagation [18].

A number of joint learning methods have been proposed [18,19] for NER and RE subtasks to avoid error propagation from NER to RE. In the case of family history information extraction, Shi et al [17] developed deep joint learning based on the BiLSTM that won the 2018 BioCreative/OHNLP challenge [20]. Joint learning methods generally used pretrained neural language models. Neural language models pretrained on large-scale unlabeled text have recently been proven to be surprisingly effective in many downstream tasks, and Bidirectional Encoder Representation from Transformers (BERT) [21] is one of the most popular neural language models.

Table 1. Normalized family member names.

Degree	Normalized family member names
1	Father, Mother, Parent, Sister, Brother, Daughter, Son, and Child
2	Grandmother, Grandfather, Grandparent, Cousin, Sibling, Aunt, and Uncle

Data Statistics

We conducted experiments on the corpus provided by the 2018 and 2019 n2c2/OHNLP shared task tracks on family history information extraction. The training set of the 2019 n2c2/OHNLP shared task together with the test set of the 2018 BioCreative/OHNLP shared task was used as the final training

In this study, we proposed a novel graph-based model with biaffine attention. Inspired by the dependency parsing task [22,23], we designed a novel graph-based schema to represent family history information and introduced deep biaffine attention [22,23] to extract family history information from clinical text. A system based on the proposed method was developed for the 2019 n2c2/OHNLP challenge on family history information extraction, and it achieved the highest F1 scores of 0.8823 on subtask1 and 0.7048 on subtask2.

Methods

Task Description

There were two subtasks in the 2019 n2c2/OHNLP challenge on family history information extraction. For subtask1, we need to recognize family members with the side of the family, living status mentioned in clinical text, and observations in the family history. All family members can be normalized to standard forms in Table 1. The property of family members named “side of family” includes the following three possible values: NA (“not applicable”), maternal, and paternal. Following the work of Shi et al [17], we compared two different strategies. The first strategy recognized three types of entities (family member, observation, and living status) and determined the “side of family” property for each family member entity through a postprocessing module. The second strategy recognized five types of entities (NA, maternal, paternal, observation, and living status), directly determining the “side of family” property of family members.

For subtask2, we need to extract the relations between family members, observations, and living status. Living status is used to represent the health status of family members, and it has the two properties of “Alive” and “Healthy.” Each property was measured by a real-valued score (yes: 2, NA: 1, and no: 0). The total living status score of family members was their alive score multiplied by their health score. We also need to predict the negation information (Negated and Non_Negated) for each observation, that is, to judge whether the family members have certain diseases or not.

set of 149 EHRs for model training. The test data set of the 2019 n2c2/OHNLP shared task, including 117 EHRs, was used for the model test. During model training, we randomly selected a development set of 14 EHRs from the training set for parameter optimization. The statistics of the corpus used in this study is shown in Table 2.

Table 2. Detailed data set statistics.

Item	Training set, n	Development set, n	Test set, n
Document	149	14	117
Sentence	770	71	644
FM ^a : overall	1128	94	— ^b
FM: NA ^c	631	55	—
FM: maternal	272	24	—
FM: paternal	225	15	—
OB ^d	1439	127	—
LS ^e	596	52	—
FM-OB: overall	1064	97	—
FM-OB: NA-OB	575	57	—
FM-OB: maternal-OB	265	23	—
FM-OB: paternal-OB	224	17	—
FM-LS: overall	605	53	—
FM-LS: NA-LS	334	29	—
FM-LS: maternal-LS	145	12	—
FM-LS: paternal-LS	126	12	—

^aFM: family member.

^bNot available.

^cNA: not applicable.

^dOB: observation.

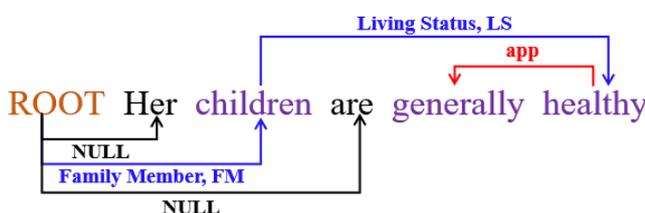
^eLS: living status.

Graph-Based Schema

Similar to the dependency parsing task where each token has a head token, we transformed the family history information extraction task to a dependency parsing problem, where a dummy root (denoted by “ROOT”) was appended to each sentence at the beginning and arcs denoted links between two tokens. In the “dependency parsing tree” of a sentence, tokens in each entity were connected together by an “app” arc from right to left, two entities with a relation were connected through linking the right most token by an arc labeled with the entity

type, and tokens not in any entity were connected with the “ROOT” node by “NULL” arcs. Figure 1 shows an example of using a “dependency parsing tree” to represent family history information extraction, where the family member entity “children” was determined by the “Family Member” arc from “ROOT” to “children,” the living status entity “generally healthy” was determined by “generally $\overline{\text{app}}$ generally,” and the relation between “children” and “generally healthy” was determined by the arc from “children” to “healthy” $\overline{\text{LS}}$.

Figure 1. Example of using a graph-based schema to represent family history information.



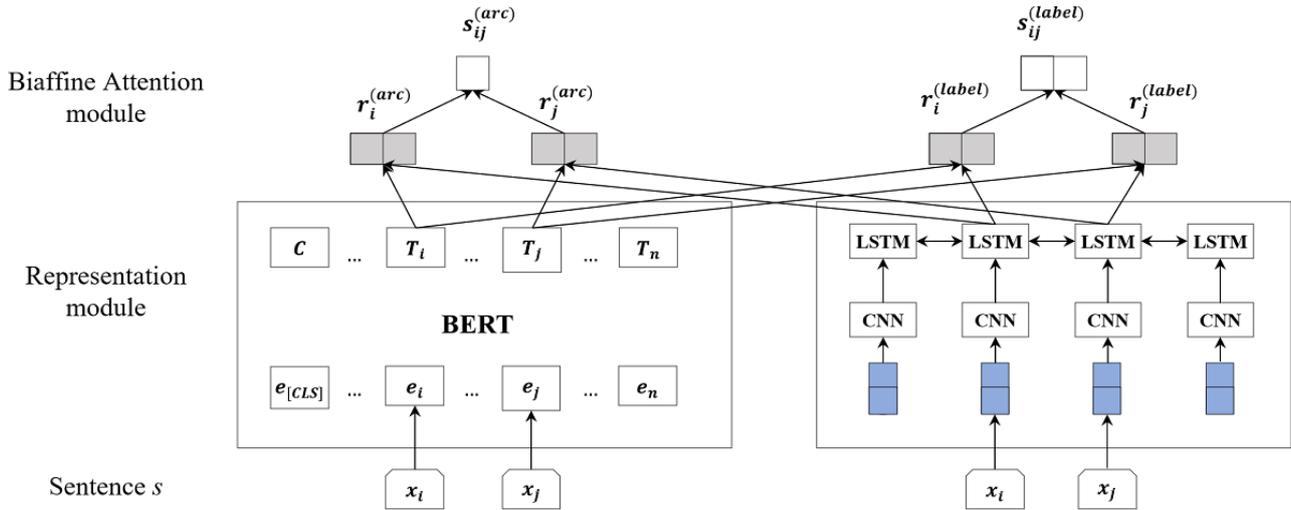
Entity: “children”, “generally healthy”
 Relation: (“children”, “generally healthy”)

Model Architecture

As shown in Figure 2, our model contained the following two main parts: (1) a representation module, which represented input text using BERT and CNN-BiLSTM and (2) a biaffine attention

module to predict label score vectors, including unlabeled arc prediction (top left in Figure 2) and arc label prediction (top right in Figure 2). We have presented them in the following sections in detail.

Figure 2. Overview architecture of our model.



Representation Layer

Given a sentence $s = x_1 \dots x_i \dots x_n$, where x_i is the i th token of s , we used BERT and CNN-BiLSTM to represent it separately as follows:

$$T_1 \dots T_i \dots T_n = \text{BERT}(s) \quad (1)$$

$$h_1 \dots h_i \dots h_n = \text{BiLSTM}(\text{CNN}(x_1) \dots \text{CNN}(x_i) \dots \text{CNN}(x_n)) \quad (2)$$

where CNN [4] is first used to get the character-level representation of each token, and BiLSTM is then used to get the contextual representation of each token in CNN-BiLSTM. The final representation of token x_i is

$$c_i = [T_i; h_i] \quad (3)$$

Biaffine Attention Layer

Unlabeled Arc Prediction

Considering the i th token and the j th token, we fed their corresponding representations into a bilinear transformation extension called a biaffine function to get the score of the arc from token i (head) to j (dependent) as follows:

$$r_j^{(arc-dep)} = \text{MLP}^{(arc-dep)}(c_j) \quad (4)$$

$$r_i^{(arc-head)} = \text{MLP}^{(arc-head)}(c_i) \quad (5)$$

$$s_{ij}^{(arc)} = r_i^{(arc-head)^T} U^{(arc)} r_j^{(arc-dep)} + r_i^{(arc-head)^T} u^{(arc)} \quad (6)$$

where $r_j^{(arc-dep)} \in \mathbb{R}^p$ and $r_i^{(arc-head)} \in \mathbb{R}^p$ are the outputs of multilayer perceptron, $U^{(arc)} \in \mathbb{R}^{p \times p}$ is a weight matrix controlling the strength of the arc from token i to j , and $u^{(arc)} \in \mathbb{R}^p$ is a bias vector.

Assume that $s_j^{(arc)} = [s_{1j}^{(arc)}; \dots; s_{nj}^{(arc)}]$ is the score vector of all possible heads of the j th token. We adopted the softmax function to compute the probability distribution d_j of all possible heads

of token j and the cross-entropy between the predicted d_j and gold standard $d_j^{(arc)}$ as the loss function as follows:

$$d_j = \text{softmax}(s_j^{(arc)}) \quad (7)$$

$$L_{arc} = -\sum d_j^{(arc)} \log(d_j) \quad (8)$$

Thereafter, the best head of token j was determined according to

$$\text{head}(j) = \arg \max s_j^{(arc)} \quad (9)$$

Arc Label Prediction

For each unlabeled arc, we need to determine its label. Assume that $s_{ij}^{(lab)} \in \mathbb{R}^{|L|}$ is the label score vector for each arc from token i to j , where $|L|$ is the size of the label set. We can compute $s_{ij}^{(lab)}$ as follows:

$$r_j^{(label-dep)} = \text{MLP}^{(label-dep)}(c_j) \quad (10)$$

$$r_i^{(label-head)} = \text{MLP}^{(label-head)}(c_i) \quad (11)$$

$$r_{ij}^{(label)} = r_i^{(label-head)} \oplus r_j^{(label-dep)} \quad (12)$$

$$s_{ij}^{(label)} = r_i^{(label-head)^T} U^{(label)} r_j^{(label-dep)} + W^{(label)} r_{ij}^{(label)} + u^{(label)} \quad (13)$$

where $r_j^{(label-dep)} \in \mathbb{R}^{|L| \times p}$ and $r_i^{(label-head)} \in \mathbb{R}^{|L| \times p}$ are outputs of the multilayer perceptron, $U^{(label)} \in \mathbb{R}^{|L| \times p \times p}$ is a third-order tensor, $W^{(label)} \in \mathbb{R}^{|L| \times 2p}$ is a weight matrix, and $u^{(label)} \in \mathbb{R}^{|L|}$ is a bias vector.

We also adopted the softmax function to compute the probability distribution d_{ij} of all possible labels of the arc from token i to j and the cross-entropy between the predicted d_{ij} and gold standard $d_{ij}^{(label)}$ as the loss function as follows:

$$d_{ij} = \text{softmax}(s_{ij}^{(label)}) \quad (14)$$

$$L_{label} = -\sum d_{ij}^{(label)} \log(d_{ij}) \quad (15)$$

Thereafter, the best label of the arc from token i to j was determined by

$$\text{label}(i, j) = \arg \max_{\text{label}} s_{ij}^{(\text{label})} \quad (16)$$

The total loss function was set as

$$L = L_{\text{arc}} + L_{\text{label}} \quad (17)$$

Postprocessing Rules

We designed a rule-based postprocessing module to adjust the outputs of our model. It included the following five parts:

1. Converting the output to entities and relations.

(1) Combining all tokens connected by “app” arcs to form entities and assigning them the label of their last token.

(2) If there was an arc between two entities, but not an “app” arc, there was a relation between them.

2. Normalizing family members.

(1) Converting family member entities into normalized forms as shown in Table 1. For example, we converted the recognized “father’s father” into “grandfather” and “aunt’s son” into “cousin.”

(2) Excluding unnecessary family members. For example, a patient’s nonblood relatives, such as “father” in section “partner’s father,” should be removed. If the family member “father” belonged to section “partner’s father,” we removed “father” since father-in-law was not in Table 1.

3. Determining the side of family members when using the strategy of three types of entities.

(1) If a family member was a first-degree relative, the side of the family was set as “NA.”

(2) If a family member was in the section “maternal family history” or “paternal family history,” the side of the family was set as maternal or paternal.

(3) If there was an indicator (“maternal” or “paternal”) near a family member, the side of the family was determined by the indicator.

(4) Otherwise, the side of the family of a family member was set as “NA.”

4. Determining the living status score of family members following the work of Shi et al [17].

(1) Determining the scores of the properties “Alive” and “Healthy” of a family member through searching the keywords listed in Table 3 from the family member’s living status. If a living status entity contained some keywords listed in Table 2, we assigned its property scores with the corresponding scores; otherwise, both its alive score and healthy score were set as NA=1.

(2) The total living status score was determined according to the alive score and healthy score. For a relative with “Alive=Yes” and “Healthy=Yes,” for example, the living status score should be 4.

5. Determining the negation information of observations.

(1) Determining the negation information of an observation through searching keywords (no, never, not, none, negative, neither, nor, unremarkable, and deny) from the observation’s context. If the context of an observation contained a keyword mentioned above, we set its negation information as “Negated;” otherwise, it was set as “Non_Negated.”

(2) Reversing the negation information of an observation if there were specific phrases, such as “apart from” and “except for,” in the observation’s context. For example, the negation information of the observation entity “Meniere disease” in “there is no history of hearing loss apart from the father’s history of Meniere disease” was set as “Non_Negated” rather than “Negated.”

Table 3. Keywords used to determine the properties “Alive” and “Healthy.”

Property	Keywords
Alive: Yes=2	Alive and living
Alive: No=0	Dead, die, deceased, death, died, stillborn, and passed away
Healthy: Yes=2	Good, health, without problems, healthy, and well

Experimental Settings

The hyperparameters used in our experiments are listed in Table 4, and all other parameters were optimized in the validation set. The pretrained BERT model we used was [BERT-Base, Uncased] [24].

We first investigated our model in the following two settings:

(1) a pipeline model that tackled unlabeled arc prediction and arc label prediction separately and (2) a joint model that tackled unlabeled arc prediction and arc label prediction simultaneously. The joint model predicated the arc and label of each token in our model jointly. The pipeline model first trained one model

to predict the head of each token and then trained another model to predict the head of each token according to the result of the predicted head. Thereafter, we compared our model with the BERT-based model using the same architecture as that of the model by Shi et al [17], except that we used BERT instead of word embeddings in the input layer (denoted by BERT-2BiLSTM). Finally, we looked into the effect of the sentence representation based on CNN-BiLSTM on our model and the effect of different data sets on our model. The performance of all models for the two subtasks was measured by precision, recall, and F1 score (F1) as follows:

$$P = TP / (TP + FP) \quad (18)$$

$$R = TP / (TP + FN) \quad (19)$$

$$F_1 = 2PR / (P + R) \quad (20)$$

where TP denotes the number of true-positive samples, FP denotes the number of false-positive samples, and FN denotes

the number of false-negative samples. We used the tool provided by the organizers [25] to calculate them. The tool accepted partial matching of the observations, for example, the recognized observation “diabetes” whose gold standard observation is “type 2 diabetes” was considered as a true-positive sample. The source code is available at GitHub [26].

Table 4. Major hyperparameters.

Parameter	Value
BiLSTM ^a size	256
Arc MLP ^b size	500
Label MLP size	100
BERT ^c size	768
Char embedding size	25
CNN ^d kernel size	(3, 4, 5)
Char-level CNN size	50
Dropout	0.5
Optimizer	Adam
Learning rate	2e-5
Batch size	32
Max epoch	100

^aBiLSTM: Bidirectional Long Short Term Memory network.

^bMLP: multilayer perceptron.

^cBERT: Bidirectional Encoder Representation from Transformers.

^dCNN: convolutional neural network.

Results

As shown in Table 5, the performance of the model considering five types of entities was better than that considering three types of entities. The joint model considering five types of entities achieved the highest F1 score of 0.8823 on the NER subtask and 0.7048 on the RE subtask, which were higher than the values for the joint model considering three types of entities by 1.20% on the NER subtask and 1.87% on the RE subtask.

Compared to the pipeline model, the joint model performed better on both the NER and RE. For example, when considering five types of entities, the joint model outperformed the pipeline model by 1.21% in the F1 score on the NER subtask and 1.97% in the F1 score on the RE subtask. It indicated that error propagation was partially alleviated in our joint model. When considering five types of entities, the joint model achieved higher F1 scores than BERT-2BiLSTM on the NER subtask and RE subtask by 1.18% and 0.39%, respectively.

Table 5. Performance of different models.

Subtask	Model	Three types of entities			Five types of entities		
		Precision	Recall	F1 score	Precision	Recall	F1 score
NER ^a	Pipeline	0.9254	0.8062	0.8617	0.9241	0.8223	0.8702
NER	Joint	0.9012	0.8415	0.8703	0.9154	0.8514	0.8823
NER	BERT ^b -2BiLSTM ^c	— ^d	—	—	0.9096	0.8347	0.8705
RE ^e	Pipeline	0.7909	0.6005	0.6827	0.7895	0.6051	0.6851
RE	Joint	0.7679	0.6200	0.6861	0.7717	0.6487	0.7048
RE	BERT-2BiLSTM	—	—	—	0.7686	0.6441	0.7009

^aNER: named entity recognition.

^bBERT: Bidirectional Encoder Representation from Transformers.

^cBiLSTM: Bidirectional Long Short Term Memory network.

^dNot available.

^eRE: relation extraction.

The performance of our best model on each type of family member information and relation (except living status not provided in the test set) is listed in [Table 6](#). On the NER subtask, our model performed better on observations than family members by 3.80% in terms of the F1 score. Among the three types of family members, our model achieved the highest F1 score of 0.8702 for maternal family member and the lowest F1 score of 0.8411 for paternal family member. On the RE subtask,

the F1 score of our model on the family member-living status relation was nearly the same as that of our model on the family member-observation relation. Among the family member-observation relations, our model performed worse on the maternal-observation relation than the other two types of relations. Among the family member-living status relations, our model performed worse on the paternal-living status relation than the other two types of relations.

Table 6. Performance of the best model on each type of family member information.

Subtask	Type	Precision	Recall	F1 score
NER ^a	FM ^b : overall	0.8814	0.8386	0.8594
NER	FM: NA ^c	0.8699	0.8515	0.8606
NER	FM: maternal	0.9185	0.8267	0.8702
NER	FM: paternal	0.8738	0.8108	0.8411
NER	OB ^d	0.9385	0.8598	0.8974
NER	LS ^e	— ^f	—	—
NER	Overall	0.9154	0.8514	0.8823
RE ^g	FM-OB: overall	0.7843	0.6397	0.7047
RE	FM-OB: NA-OB	0.8595	0.6098	0.7134
RE	FM-OB: maternal-OB	0.7067	0.6601	0.6826
RE	FM-OB: paternal-OB	0.7077	0.7150	0.7113
RE	FM-LS: overall	0.7627	0.6553	0.7050
RE	FM-LS: NA-LS	0.7627	0.6553	0.7050
RE	FM-LS: maternal-LS	0.7108	0.7375	0.7239
RE	FM-LS: paternal-LS	0.6825	0.6825	0.6825
RE	Overall	0.7717	0.6487	0.7048

^aNER: named entity recognition.

^bFM: family member.

^cNA: not applicable.

^dOB: observation.

^eLS: living status.

^fNot available.

^gRE: relation extraction.

As shown in Table 7, without using the additional data for BioCreative/OHNLNLP 2018, our model considering five types of entities achieved an F1 score of 0.8648 on the NER subtask

and 0.6612 on the RE subtask (the F1 score was significantly reduced both on the NER subtask and RE subtask), showing the importance of the data.

Table 7. Performance of our model with different data.

Subtask	Data set	Three types of entities			Five types of entities		
		Precision	Recall	F1 score	Precision	Recall	F1 score
NER ^a	2019	0.8767	0.8409	0.8584	0.8847	0.8458	0.8648
NER	2018+2019 ^b	— ^c	—	—	0.9154	0.8372	0.8745
NER	2018+2019 ^d	0.9012	0.8415	0.8703	0.9154	0.8514	0.8823
RE ^e	2019	0.7240	0.5973	0.6545	0.7270	0.6064	0.6612
RE	2018+2019 ^b	—	—	—	0.7459	0.6265	0.6810
RE	2018+2019 ^d	0.7679	0.6200	0.6861	0.7717	0.6487	0.7048

^aNER: named entity recognition.

^b2018+2019: the challenge submission performances of our model.

^cNot available.

^d2018+2019: the performances of our best model after challenge.

^eRE: relation extraction.

Discussion

Effect of Sentence Representation

In order to investigate the effect of sentence representation based on CNN-BiLSTM on our model, we evaluated the model without using the representation and obtained an F1 score of 0.8802 on the NER subtask and an F1 score of 0.7059 on the RE subtask when considering five types of entities. The sentence representation based on CNN-BiLSTM can bring improvement in the NER subtask, but a little loss in the RE subtask. Possibly, we can only share BERT on NER and RE for further improvement.

Impact of Different Decoders on the NER Subtask

Traditional approaches regarded the NER task as a sequence labeling task, in which each token was assigned with a combined label of entity boundary and type. The entity boundaries were represented by the BIO schema, where “B” indicates the

beginning of an entity, “I” indicates the inside of an entity, and “O” indicates the outside of an entity. Using a graph schema, we can also convert NER into a graph in the following way: (1) connect all tokens with “ROOT,” that is, the heads of all tokens are set to 0 and (2) set the label of the nonentity token to “NULL,” set the label of the last token in the entity to the entity type, and set the label of the remaining token in the entity to “app.”

We compared different decoders, that is, CRF for sequence labeling, biaffine for NER only (biaffine-NER), and biaffine for joint NER and RE (biaffine-Joint). As shown in Table 8, the performance of biaffine-NER was slightly better than that of CRF, while biaffine-Joint was considerably better than the other two models. Although the head prediction was not directly related to the NER task, the arcs of different types among tokens provided global information that was beneficial to the NER task.

Table 8. Comparison of different decoders on the named entity recognition subtask.

Decoder	Three types of entities			Five types of entities		
	Precision	Recall	F1 score	Precision	Recall	F1 score
CRF ^a	0.8989	0.8316	0.8639	0.9070	0.8390	0.8717
Biaffine-NER ^b	0.9001	0.8310	0.8641	0.8895	0.8570	0.8729
Biaffine-Joint	0.9012	0.8415	0.8703	0.9154	0.8514	0.8823

^aCRF: conditional random field.

^bNER: named entity recognition.

Error Analysis

We performed error analysis on our model considering five types of entities in the development data set. In the case of the NER subtask, 88.24% of errors were boundary errors because of wrong “app” arc prediction, while the remaining 11.76% of errors were type errors that have a correct boundary but wrong entity type. For example, in the sentence “The paternal grandmother, age 53, has wind sucking attributed to not having intestinal during her life,” the paternal entity “grandmother” with the observation entity “wind sucking” was wrongly recognized as a family member entity. In the RE subtask, all errors were caused by incorrect entities. For example, in the sentence “The patient’s father is 43 years old and healthy. His father is 72 years old and was diagnosed with esophageal cancer at age 70,” the family member entity “grandfather” with the observation entity “esophageal cancer” was wrongly extracted as the family member entity “father” with the observation entity

“esophageal cancer” as our model could not understand that “his” refers to “the patient’s father,” which needs strong indirect relative reasoning.

Limitations and Future Work

The rule-based postprocessing module in our system cannot handle all cases properly, as shown by the example in the error analysis section. In future work, we will try to solve indirect relative reasoning for further improvement.

Conclusions

In this study, we proposed a novel graph-based model with biaffine attention, where a graph-based schema was design to represent entities and relations regarding family history in a unified way and deep biaffine attention was adopted to extract the entities and relations from clinical text. Our system based on the proposed model achieved the highest F1 score of the challenge to date.

Acknowledgments

This paper was supported in part by the following grants: National Natural Science Foundations of China (U1813215, 61876052, and 61573118), Special Foundation for Technology Research Program of Guangdong Province (2015B010131010), National Natural Science Foundations of Guangdong, China (2019A1515011158), Guangdong Province Covid-19 Pandemic Control Research Fund (2020KZDZX1222), Strategic Emerging Industry Development Special Funds of Shenzhen (JCYJ20180306172232154 and JCYJ20170307150528934), and Innovation Fund of Harbin Institute of Technology (HIT.NSRIF.2017052).

Conflicts of Interest

None declared.

References

1. Lafferty J, McCallum A, Pereira F. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: Proceedings of the Eighteenth International Conference on Machine Learning. 2001 Presented at: Eighteenth International Conference on Machine Learning; June 28-July 1, 2001; Williamstown, MA, USA p. 282-289.
2. Tang B, Cao H, Wu Y, Jiang M, Xu H. Recognizing clinical entities in hospital discharge summaries using Structural Support Vector Machines with word representation features. *BMC Med Inform Decis Mak* 2013;13(Suppl 1):S1. [doi: [10.1186/1472-6947-13-s1-s1](https://doi.org/10.1186/1472-6947-13-s1-s1)]
3. Huang Z, Xu W, Yu K. Bidirectional LSTM-CRF Models for Sequence Tagging. arXiv. 2015. URL: <http://arxiv.org/abs/1508.01991> [accessed 2021-03-29]
4. Ma X, Hovy E. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2016 Presented at: 54th Annual Meeting of the Association for Computational Linguistics; 2016; Berlin, Germany p. 1064-1074. [doi: [10.18653/v1/p16-1101](https://doi.org/10.18653/v1/p16-1101)]
5. Tang B, Hu J, Wang X, Chen Q. Recognizing Continuous and Discontinuous Adverse Drug Reaction Mentions from Social Media Using LSTM-CRF. *Wireless Communications and Mobile Computing* 2018;2018:1-8. [doi: [10.1155/2018/2379208](https://doi.org/10.1155/2018/2379208)]
6. Cortes C, Vapnik V. Support-vector networks. *Mach Learn* 1995 Sep;20(3):273-297. [doi: [10.1007/bf00994018](https://doi.org/10.1007/bf00994018)]
7. Sahu S, Anand A, Oruganty K. Relation extraction from clinical texts using domain invariant convolutional neural network. In: Proceedings of the 15th Workshop on Biomedical Natural Language Processing. 2016 Presented at: 15th Workshop on Biomedical Natural Language Processing; 2016; Berlin, Germany p. 206-215. [doi: [10.18653/v1/w16-2928](https://doi.org/10.18653/v1/w16-2928)]
8. Luo Y. Recurrent neural networks for classifying relations in clinical notes. *J Biomed Inform* 2017 Aug;72:85-95 [FREE Full text] [doi: [10.1016/j.jbi.2017.07.006](https://doi.org/10.1016/j.jbi.2017.07.006)] [Medline: [28694119](https://pubmed.ncbi.nlm.nih.gov/28694119/)]
9. Uzuner O, Solti I, Cadag E. Extracting medication information from clinical text. *J Am Med Inform Assoc* 2010;17(5):514-518 [FREE Full text] [doi: [10.1136/jamia.2010.003947](https://doi.org/10.1136/jamia.2010.003947)] [Medline: [20819854](https://pubmed.ncbi.nlm.nih.gov/20819854/)]
10. Tang B, Wu Y, Jiang M, Chen Y, Denny JC, Xu H. A hybrid system for temporal information extraction from clinical text. *J Am Med Inform Assoc* 2013 Sep 01;20(5):828-835 [FREE Full text] [doi: [10.1136/amiajnl-2013-001635](https://doi.org/10.1136/amiajnl-2013-001635)] [Medline: [23571849](https://pubmed.ncbi.nlm.nih.gov/23571849/)]
11. Liu Z, Chen Y, Tang B, Wang X, Chen Q, Li H, et al. Automatic de-identification of electronic medical records using token-level and character-level conditional random fields. *J Biomed Inform* 2015 Dec;58 Suppl:S47-S52 [FREE Full text] [doi: [10.1016/j.jbi.2015.06.009](https://doi.org/10.1016/j.jbi.2015.06.009)] [Medline: [26122526](https://pubmed.ncbi.nlm.nih.gov/26122526/)]
12. Zhang Y, Wang J, Tang B. UTH_CCB: A report for SemEval 2014 – Task 7 Analysis of Clinical Text. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014). 2014 Presented at: 8th International Workshop on Semantic Evaluation (SemEval 2014); 2014; Dublin, Ireland p. 802-806. [doi: [10.3115/v1/S14-2142](https://doi.org/10.3115/v1/S14-2142)]
13. Bethard S, Derczynski L, Savova G, Pustejovsky J, Verhagen M. SemEval-2015 Task 6: Clinical TempEval. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015). 2015 Presented at: 9th International Workshop on Semantic Evaluation (SemEval 2015); June 4-5, 2015; Denver, Colorado p. 806-814. [doi: [10.18653/v1/s15-2136](https://doi.org/10.18653/v1/s15-2136)]
14. Kelly L, Goeriot L, Suominen H, Névéol A, Palotti J, Zuccon G. Overview of the CLEF eHealth Evaluation Lab 2016. In: Experimental IR Meets Multilinguality, Multimodality, and Interaction. CLEF 2016. Lecture Notes in Computer Science, vol 9822. Cham: Springer; 2016:255-266.
15. Suominen H, Salanterä S, Velupillai S, Chapman WW, Savova G, Elhadad N, et al. Overview of the ShARe/CLEF eHealth Evaluation Lab 2013. In: Information Access Evaluation. Multilinguality, Multimodality, and Visualization. CLEF 2013. Lecture Notes in Computer Science, vol 8138. Berlin, Heidelberg: Springer; 2013:212-231.
16. Goeriot L, Kelly L, Li W, Palotti J, Pecina P, Zuccon G, et al. ShARe/CLEF eHealth Evaluation Lab 2014, Task 3: User-centred health information retrieval. In: CLEF 2014 Online Working Notes. 2014 Presented at: CEUR Workshop; September 15-18, 2014; Sheffield, UK p. 43-61.
17. Shi X, Jiang D, Huang Y, Wang X, Chen Q, Yan J, et al. Family history information extraction via deep joint learning. *BMC Med Inform Decis Mak* 2019 Dec 27;19(Suppl 10):277 [FREE Full text] [doi: [10.1186/s12911-019-0995-5](https://doi.org/10.1186/s12911-019-0995-5)] [Medline: [31881967](https://pubmed.ncbi.nlm.nih.gov/31881967/)]
18. Li Q, Ji H. Incremental Joint Extraction of Entity Mentions and Relations. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2014 Presented at: 52nd Annual Meeting of the Association for Computational Linguistics; June 22-27, 2014; Baltimore, MD, USA p. 402-412. [doi: [10.3115/v1/p14-1038](https://doi.org/10.3115/v1/p14-1038)]
19. Miwa M, Bansal M. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2016 Presented at: 54th Annual Meeting of the Association for Computational Linguistics; August 7-12, 2016; Berlin, Germany p. 1105-1116. [doi: [10.18653/v1/p16-1105](https://doi.org/10.18653/v1/p16-1105)]
20. Liu S, Mojarad MR, Wang Y, Wang L. Overview of the BioCreative/OHNLP 2018 Family History Extraction Task. In: Proceedings of the BioCreative Workshop. 2018 Presented at: BioCreative Workshop; 2018; Washington, DC, USA.

21. Devlin J, Chang M, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). 2019 Presented at: 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers); June 2-7, 2019; Minneapolis, MN, USA p. 4171-4186. [doi: [10.18653/v1/n19-1423](https://doi.org/10.18653/v1/n19-1423)]
22. Dozat T, Manning C. Deep Biaffine Attention for Neural Dependency Parsing. In: Proceedings of the 5th International Conference on Learning Representations. 2017 Presented at: 5th International Conference on Learning Representations; April 24-26, 2017; Toulon, France.
23. Yan H, Qiu X, Huang X. A Graph-based Model for Joint Chinese Word Segmentation and Dependency Parsing. Transactions of the Association for Computational Linguistics 2020 Dec;8:78-92. [doi: [10.1162/tacl_a_00301](https://doi.org/10.1162/tacl_a_00301)]
24. TensorFlow code and pre-trained models for BERT. GitHub. URL: <https://github.com/google-research/bert> [accessed 2021-03-05]
25. OHNLP/BioCreative 2018 Task 1: Family History Extraction. GitHub. URL: https://github.com/ohnlp/fh_eval [accessed 2021-03-05]
26. 2019 n2c2/OHNLP Track2 family history extraction. GitHub. URL: https://github.com/zkczzj/2019_n2c2_FHExtraction [accessed 2021-03-26]

Abbreviations

BERT: Bidirectional Encoder Representation from Transformers

BiLSTM: Bidirectional Long Short Term Memory

CNN: convolutional neural network

CRF: conditional random field

EHR: electronic health record

NER: named entity recognition

NLP: natural language processing

RE: relation extraction

Edited by Y Wang, F Shen; submitted 17.08.20; peer-reviewed by I Mircheva, X Yang; comments to author 12.10.20; revised version received 18.01.21; accepted 09.02.21; published 21.04.21

Please cite as:

Zhan K, Peng W, Xiong Y, Fu H, Chen Q, Wang X, Tang B

Novel Graph-Based Model With Biaffine Attention for Family History Extraction From Clinical Text: Modeling Study

JMIR Med Inform 2021;9(4):e23587

URL: <https://medinform.jmir.org/2021/4/e23587>

doi: [10.2196/23587](https://doi.org/10.2196/23587)

PMID:

©Kecheng Zhan, Weihua Peng, Ying Xiong, Huhao Fu, Qingcai Chen, Xiaolong Wang, Buzhou Tang. Originally published in JMIR Medical Informatics (<https://medinform.jmir.org>), 21.04.2021. This is an open-access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work, first published in JMIR Medical Informatics, is properly cited. The complete bibliographic information, a link to the original publication on <http://medinform.jmir.org/>, as well as this copyright and license information must be included.