

METHOD AND APPARATUS FOR EVENT EXTRACTION, ELECTRONIC DEVICE AND STORAGE MEDIUM

TECHNICAL FIELD

[0001] The disclosure relates to the field of natural language processing (NLP) technologies, and relates to, but not limited to, an event extraction method and apparatus, and an electronic device and a storage medium.

BACKGROUND

[0002] Event extraction is an important problem in the field of information extraction. Event extraction is the extraction of corresponding trigger words and event arguments from a text with pre-given event types and argument roles. Event extraction models have important applications in many fields. For example, in large-scale texts in specific fields (for example, legal, medical, financial fields), corresponding structured information is extracted, or relevant event-based knowledge is extracted to build a large-scale knowledge graph.

[0003] However, many current event extraction methods focus on extraction from a single-sentence level text, and an event extraction task is considered as a question answering task. These methods have proven to be inadequate in event extraction from a document-level text. The main problems are as follows: In one aspect, in event extraction from a document-level text, many event arguments are distributed across a plurality of sentences in a document, and existing models still cannot adequately model relationships between sentences. In another aspect, one same event has a plurality of event arguments.

Some event arguments may belong to one same argument role. During matching of an event argument and an argument role in an existing model, argument roles corresponding to different event arguments still cannot be adequately recognized. In addition, it is also very difficult to implement simultaneous extraction of a plurality of arguments.

SUMMARY

[0004] In view of this, embodiments of the disclosure provide a method and apparatus for event extraction, and an electronic device and a storage medium.

[0005] According to the first aspect, embodiments of the disclosure provide a method for event extraction. The method includes: acquiring an inputted text, where the inputted text includes at least one trigger word and at least one event argument; extracting, through an encoder and a decoder, a first text feature vector set corresponding to the inputted text; acquiring a target extraction template corresponding to the at least one trigger word; decoding, through the decoder, the target extraction template and the inputted text to obtain an argument feature vector of each event argument; and determining, based on the first text feature vector set and the argument feature vector of each event argument, a value range corresponding to each event argument.

[0006] According to a second aspect, embodiments of the disclosure provide a method for training an event extraction model. The event extraction model includes an encoder, a decoder, and an argument detection module. The method includes: acquiring a training text and a target extraction template corresponding to the training text; extracting, through the encoder and the decoder, a first text feature vector set corresponding to the training text; encoding, through the encoder, the target extraction template and the training text to obtain an argument feature vector of each event argument; determining, through the argument detection module, a prediction range corresponding to each event argument based on the first text feature vector set and the argument feature vector of each event argument by using the argument detection module; matching the prediction range

of each event argument against a true value range to obtain an optimal prediction range that has a loss function value satisfying a second preset condition and corresponds to each event argument; determining a loss function value of the event extraction model based on the optimal prediction range of each event argument; and training the event extraction model based on the loss function value.

[0007] According to a third aspect, embodiments of the disclosure provide an apparatus for event extraction. The apparatus includes: a first acquisition module, configured to acquire an inputted text, where the inputted text includes at least one trigger word and at least one event argument; a first extraction module, configured to extract, through an encoder and a decoder, a first text feature vector set corresponding to the inputted text; a second acquisition module, configured to acquire a target extraction template corresponding to the at least one trigger word; a first decoding module, configured to decode, through the decoder, the target extraction template and the inputted text to obtain an argument feature vector of each event argument; and a first determination module, configured to determine, based on the first text feature vector set and the argument feature vector of each event argument, a value range corresponding to each event argument based on the first text feature vector set and the argument feature vector of each event argument.

[0008] According to a fourth aspect, embodiments of the disclosure provide an apparatus for training an event extraction model. The apparatus includes: a fourth acquisition module, configured to acquire a training text and a target extraction template corresponding to the training text; a second extraction module, configured to extract, through the encoder and the decoder, a first text feature vector set corresponding to the training text; a second encoding module, configured to encode, through the encoder, the target extraction template and the training text to obtain an argument feature vector of each event argument; a sixth determination module, configured to determine, through the argument detection module, a prediction range corresponding to each event argument based on the first text feature vector set and the argument feature vector of each event argument; a matching module, configured to match the prediction range of each event argument against a true value range to obtain an optimal prediction range that has a loss function value satisfying a second preset condition and corresponds to each event

argument; a seventh determination module, configured to determine a loss function value of the event extraction model based on the optimal prediction range of each event argument; and a training module, configured to train the event extraction model based on the loss function value.

[0009] According to a fifth aspect, embodiments of the disclosure provide an electronic device. The device includes a memory and a processor, the memory storing computer programs executable on the processor, the processor executing the computer programs to implement the steps in the foregoing method.

[0010] According to a sixth aspect, embodiments of the disclosure provide a computer-readable storage medium, storing computer programs, where a processor is configured to execute the computer programs to implement the steps in the foregoing method.

[0011] According to a seventh aspect, embodiments of the disclosure provide a computer program product, comprising computer readable codes, wherein when the computer readable codes run in a device, a processor in the device executes instructions for implementing the steps in the foregoing method.

[0012] In the embodiments of the disclosure, firstly, a first text feature vector set corresponding to an inputted text is extracted through an encoder and a decoder, and then a target extraction template and the inputted text are decoded through the decoder to obtain an argument feature vector of each event argument. Because the target extraction template and the inputted text are decoded together by using the decoder, the argument feature vector of each event argument obtained through decoding by the decoder is merged with information of the inputted text. The first text feature vector set contains text feature vectors reflecting context. In this case, a value range that corresponds to each event argument and is determined based on the first text feature vector set and the argument feature vector of each event argument is a value range reflecting the context, so that the accuracy is higher, thereby improving the accuracy of event argument extraction (EAE). In addition, because a target extraction template includes argument roles and relationships between the argument roles, the establishment of cross-sentence relationships between a plurality of event arguments is implemented. In this way, the

problem of a long distance between event arguments in event extraction in sentence-level and document-level texts is resolved, and a plurality of event arguments may be predicted at once, so that the speed of event extraction is greatly increased, and during the construction of a large-scale event knowledge graph, the construction efficiency and scale of the event knowledge graph are improved. In addition, a plurality of slots are introduced for one argument role in a target extraction template, thereby effectively reducing cases in which event extraction cannot adequately match that one argument role corresponds to a plurality of event arguments, thereby improving the accuracy of matching.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] In the accompanying drawings (which are not necessarily drawn to scale), similar reference numerals may depict similar parts in different views. Similar reference numerals with different letter suffixes may indicate different examples of similar parts. The accompanying drawings illustrate generally, by way of example and not limitation, the various embodiments discussed herein.

[0014] FIG. 1 is a schematic flowchart of a method for event extraction according to embodiments of the disclosure;

[0015] FIG. 2 is a schematic flowchart of another method for event extraction according to embodiments of the disclosure;

[0016] FIG. 3 is a schematic flowchart of determining a value range corresponding to each event argument according to embodiments of the disclosure;

[0017] FIG. 4 is a schematic flowchart of a method for training an event extraction model according to embodiments of the disclosure;

[0018] FIG. 5 is a schematic flowchart of another method for event extraction according to embodiments of the disclosure;

- [0019] FIG. 6 is a schematic structural diagram of an apparatus for event extraction according to embodiments of the disclosure;
- [0020] FIG. 7 is a schematic structural diagram of an apparatus for training an event extraction model according to embodiments of the disclosure; and
- [0021] FIG. 8 is a schematic diagram of a hardware entity of an electronic device according to embodiments of the disclosure.

DETAILED DESCRIPTION

- [0022] The technical solutions of the disclosure are further described below in detail with reference to the accompanying drawings and the embodiments. Apparently, the described embodiments are merely some rather than all of the embodiments of the disclosure. All other embodiments obtained by persons of ordinary skill in the art based on the embodiments of the disclosure without creative efforts fall within the protection scope of the disclosure.
- [0023] In the following description, reference is made to "some embodiments", which describes a subset of all possible embodiments, but it may be understood that "some embodiments" may be the same subset or a different subset of all possible embodiments, and may be combined with each other without conflict.
- [0024] In the description that follows, suffixes such as "module," "part," or "unit" are used to denote components only to facilitate the description of the disclosure and have no specific meanings of their own. Therefore, "module", "part" or "unit" may be used interchangeably.
- [0025] It needs to be noted that references to the terms "first, second, and third" in the embodiments of the disclosure are only to distinguish similar objects and do not denote a specific order of objects, but rather the terms "first, second, and third" are used to

interchange specific orders or sequences, where appropriate, to enable embodiments of the disclosure described herein to be practiced in an order other than the order shown or described herein.

[0026] Embodiments of the disclosure provide an event extraction method. As shown in FIG. 1, the method includes the following steps.

[0027] In S101, an inputted text of a sentence-level or document-level text is acquired, where the inputted text includes at least one trigger word and at least one event argument.

[0028] The inputted text is used for describing event content, and may be a sentence-level or document-level text. The sentence-level text is a text in the form of a single sentence. The document-level text is a text in the form of a plurality of sentences.

[0029] Components of an event may include a trigger word, an event type, an event argument, and an argument role. The trigger word refers to a core word for the occurrence of the event, and is usually a verb or a noun, for example, "attack", or "discuss". The event type refers to a type of the event, for example, a movement event, or an attack event. Event types corresponding to different data sets have different classes. For example, eight event types and 33 subtypes are defined in the data set ACE2005. 33 event types are used for the extraction of most events. The event argument refers to a participant of the event, for example, Sam Zhang (that is, a specific person), or a school. The argument role refers to a role that the event argument plays in the event, for example, "attacker", "position", or "time", etc.

[0030] The following event is used as an example for description below. The private Dogan news agency reported that more than 30 of those wounded in A, primarily children, were transferred to a hospital in B.

[0031] It showed footage of ambulances arriving at C hospital and medical personnel <t>unloading</t> a girl on stretchers.

[0032] The event type is a movement event triggered by a trigger word "unloading". Event arguments are "A", "ambulances", "C hospital", "medical personnel", and "a girl". All event arguments play specific roles in the event. The roles include a source (that is, a

place indicated by "A"), a vehicle (that is, a vehicle indicated by "ambulances"), a destination (that is, a place indicated by "C hospital"), a transporter (that is, a transporter indicated by "medical personnel"), and a passenger (that is, a passenger indicated by "a girl").

[0033] In some embodiments, the inputted text may include a trigger word and at least one event argument. The at least one may be one or at least two, which may be determined according to content of the inputted text.

[0034] During description, an event extraction task is first formulated. A series of event types E are then predefined. The predefined event type may be defined based on a data set. For example, for the data set ACE2005, there are a total of 33 event types. Each event type $t \in E$. Each event type t includes a series of argument roles R^t , where $R^t = \{r_1^t, r_2^t, r_3^t, \dots, r_m^t\}$. For ease of description, a superscript that indicates the event type t is removed below.

[0035] Next, an inputted text X of a sentence-level or document-level text is acquired. $X = \{x_1, x_2, x_3, \dots, x_n\}$. Each x_i corresponds to one lemma. When the inputted text X is in a vector form, each x_i corresponds to one lemma vector. The inputted text X includes a trigger word x_{trig} and at least one event argument A_{ri} . An event argument set $A^t = \{A_{r1}, A_{r2}, \dots, A_{rm}\}$. The event extraction task is that in the inputted text X , an event argument A_{ri} corresponding to each argument role r_i is extracted for the argument role. Each argument role A_{ri} is content corresponding to a span of at least one lemma in X . For example, the event argument is "Sam Zhang". Each of "Sam" and "Zhang" in "Sam Zhang" corresponds to one lemma. The event argument "Sam Zhang" is the content corresponding to a span of the two lemmas "Sam" and "Zhang".

[0036] In S102, a first text feature vector set corresponding to the inputted text is extracted through an encoder and a decoder.

[0037] The first text feature vector set contains text feature vectors reflecting context. A feature vector corresponding to each lemma is an h -dimensional vector. The encoder and the decoder may respectively use an encoder and decoder in a Transformer model. The Transformer model is a Natural Language Processing (NLP) classic model

proposed in 2017. The Transformer model uses a self-attention mechanism. The encoder includes a self-attention layer. The decoder includes a self-attention layer and a cross-attention layer. The self-attention layer is used for paying attention to relationships between a plurality of lemmas in a single sentence in the inputted text. The cross-attention layer is used for paying attention to relationships between a plurality of sentences of the inputted text. The inputted text is subjected to the attention mechanism of the attention layer to generate interaction between different lemmas in the inputted text, to obtain the first text feature vector set considering the context.

[0038] In some embodiments, the encoder and the decoder may also use a BERT model or a BART model. The BERT model is a Pretrained Language Model (PLM) that only uses an encoder structure of a Transformer model (i.e., a Transformer-Encoder). The BART model is a PLM that uses an overall structure of a transformer model.

[0039] In some embodiments, the implementation of step S102 may include S1021 and S1022.

[0040] In S1021, the at least one trigger word of the inputted text is acquired, and the at least one trigger word of the inputted text is marked.

[0041] The trigger word of the inputted text may be manually recognized or automatically recognized by a device and marked. In some embodiments, the trigger word may be marked by using a special symbol (referring to the following examples) or in another manner. A manner of marking the trigger word in the inputted text is not limited in the embodiments of the disclosure.

[0042] An example of marking a trigger word is provided below.

[0043] The private Dogan news agency reported that more than 30 of those wounded in A, primarily children, were transferred to a hospital in B.

[0044] It showed footage of ambulances arriving at C hospital and medical personnel <t>unloading</t> a girl on stretchers.

[0045] "Unloading" is a trigger word, and is marked by using < t > and </t>.

- [0046] An example of marking a trigger word is provided below.
- [0047] The inputted text is $X = \{x_1, x_2, x_3 \dots x_n\}$.
- [0048] The inputted text with the trigger word marked is $\hat{X} = \{x_1, x_2, < t >, x_3, x_4, < /t >, \dots, x_n\}$.
- [0049] In S1022, the first text feature vector set corresponding to the marked inputted text is extracted through the encoder and the decoder.
- [0050] In the embodiments of the disclosure, by means of a method for marking a trigger word in the inputted text, the first text feature vector set corresponding to the inputted text can be effectively extracted.
- [0051] In S103, a target extraction template corresponding to the at least one trigger word is acquired.
- [0052] The target extraction template is a template formed by splicing argument roles in an event type based on relationships between the argument roles. For example, argument roles in an attack event include an attacker, a victim, and a place. The splicing may be performed according to relationships between a plurality of argument roles to obtain a target extraction template: (attacker) attacks (victim) at (place).
- [0053] A trigger word and a target extraction template may have a one-to-one relationship, or may have a multiple-to-one relationship. That is, one trigger word corresponds to one target extraction template, or a plurality of trigger words correspond to one target extraction template. For example, trigger words are "contact" and "negotiate". Both the trigger words correspond to one target extraction template: participant communicated with participant about topic at place. A position corresponding to each argument role is called a slot. The target extraction template has a total of four slots "participant", "participant", "topic", and "place". Two slots correspond to one argument role "participant". That is, one argument role may correspond to a plurality of slots in the target extraction template.
- [0054] Table 1 is an example of the target extraction template:

[0055] Table 1

Data set	Representation of an event type	Representation of a natural language template
ACE05	Personnel.Nominate	<u>Agent</u> (and <u>Agent</u>) nominated <u>Person</u> with <u>Position</u> at <u>Place</u> (and <u>Place</u>)
ACE05	Personnel.End-Position	<u>Person</u> stopped working at <u>Position</u> of <u>Entity</u> (and <u>Entity</u>) at <u>Place</u>
RAMS	government.spy.spy	<u>spy</u> spied on <u>observed entity</u> to the benefit of <u>beneficiary</u> in <u>place</u>
RAMS	justice.investigate.n/a	<u>investigator</u> investigated <u>defendant</u> in <u>place</u>
WIKIEVENT	Conflict.Defeat.Unspecified	<u>Victor</u> (and <u>Victor</u>) defeated in <u>Conflict</u> or <u>Election</u> at <u>Place</u> (and <u>Place</u>)
WIKIEVENT	Life.Consume.Unspecified	<u>Consuming Entity</u> consumed <u>Consumed Thing</u> at <u>Place</u> (and <u>Place</u>)

[0056] Each slot is represented by an underlined word. Parentheses represent an N-to-one argument role (that is, one argument role corresponds to a plurality of event

arguments).

[0057] In some embodiments, the target extraction template may include a plurality of argument roles, and a plurality of event arguments corresponding to the plurality of argument roles may be distributed in a plurality of sentences. For example, the inputted text is: "The private Dogan news agency reported that more than 30 of those wounded in A, primarily children, were transferred to a hospital in B.

[0058] It showed footage of ambulances arriving at C hospital and medical personnel <t>unloading</t> a girl on stretchers".

[0059] An event argument "A" is in the first sentence. An event argument "a girl" is in the second sentence. Therefore, when event argument is extracted by using a target extraction template, a plurality of event arguments may be extracted from the target extraction template at once. In addition, extraction is performed on a plurality of sentences, so that the problems of a long distance between event arguments in event extraction in sentence-level and document-level texts and multi-event argument are resolved.

[0060] In some embodiments, the target extraction template corresponding to the trigger word may be acquired manually or automatically acquired by a device. During description, the target extraction template may be determined based on the trigger word.

[0061] Correspondingly, before the implementation of step S103, the method may include S1021 and S1022.

[0062] In S1031, a target event type is determined based on the at least one trigger word.

[0063] In S1032, the target extraction template is determined from a preset template library based on the target event type.

[0064] Before the implementation of step S1032 "determining the target extraction template from a preset template library based on the target event type", the method may include S103a to S103d.

- [0065] In S103a, an event type set corresponding to a text set is determined, where the event type set includes at least one event type.
- [0066] The text set is all text data in the data set. For example, the data set is ACE2005, and there are 400 corresponding texts, so that the text set includes the 400 texts.
- [0067] In S103b, at least two argument roles in each event type and relationship information between the at least two argument roles are determined.
- [0068] In S103c, the at least two argument roles are spliced based on the relationship information between the at least two argument roles for each event type to obtain an extraction template of each event type.
- [0069] In S103d, the preset template library is formed based on the extraction template of each event type in the event type set.
- [0070] In S104, the target extraction template and the inputted text are decoded through the decoder to obtain an argument feature vector of each event argument.
- [0071] The argument feature vector is a feature vector merged with information of the inputted text. Each argument feature vector corresponds to a respective argument role. The argument feature vector is $\psi_{ri}^k \in R^h$. R^h denotes an h-dimensional vector, ri denotes the argument role, and k denotes each slot in the target extraction template. It needs to be noted that in the embodiments of the disclosure, for one argument role, a plurality of slots may be introduced into the target extraction template. Because each argument feature vector corresponding to each slot is a feature vector merged with the information of the inputted text, and there is a slight difference in context for each slot. For example, the inputted text is "Sam Zhang and Sid Lee are chatting in the People's Park". Because a slot corresponding to "Sam Zhang" and a slot corresponding to "Sid Lee" have different positions in the context, there is a slight difference in context between the slot corresponding to "Sam Zhang" and the slot corresponding to "Sid Lee", so that argument feature vectors having different meanings tend to be generated, thereby facilitating recognition in a case that one argument role corresponds to a plurality of event

arguments, to better implement accurate extraction.

[0072] Inner products need to be calculated for the following start search vector and end search vector of the argument feature vector and the vector corresponding to each lemma in the following first text feature vector. Therefore, a dimensionality of a vector corresponding to each lemma in the first text feature vector should be the same as a dimensionality of the start search vector and the end search vector. The inputted text is subsequently converted into a vector form (that is, the following text word vector set), and is then converted into the first text feature vector. The argument feature vector is converted into the start search vector and the end search vector, Therefore, in some embodiments, a dimensionality of the argument feature vector may be the same as a dimensionality of a vector corresponding to one lemma in the inputted text in a vector form after conversion, to facilitate subsequent operation processing.

[0073] In some embodiments, the inputted text may be a vector set obtained after the inputted text is encoded by the encoder. The decoder decodes the target extraction template and the vector set to obtain the argument feature vector of each event argument.

[0074] In S105, a value range corresponding to each event argument is determined based on the first text feature vector set and the argument feature vector of each event argument.

[0075] The value range refers to indexes of a start lemma and an end lemma corresponding to the event argument A_{ri} in the inputted text X . For example, indexes of the start lemma and the end lemma of the event argument A_{ri} in the inputted text X are respectively 3 and 4. The value range of the event argument A_{ri} is (3, 4).

[0076] In some embodiments, each word corresponding to a value range of each event argument A_{ri} may be determined from the inputted text according to the value range, to obtain each event argument. For example, the inputted text is "Sam Zhang and Sid Lee are chatting in the People's Park". The value range of the event argument A_{ri} is (1, 2). (1, 2) corresponds to "Sam Zhang" in the inputted text. The event argument A_{ri} is "Sam Zhang".

[0077] In some embodiments, there may be more argument roles in the target extraction template than in the inputted text. In this case, a result of the extracted event argument A_{r_i} may be that there is no answer. For example, the target extraction template is "(participant A) and (participant B) are communicating about (topic) at (place)". The inputted text is "Sam Zhang and Sid Lee are chatting in the People's Park". In this case, an event argument corresponding to an argument role "topic" is empty. During the extraction of the content of the event argument corresponding to the argument role "topic", the result is that there is no answer.

[0078] An expression of the extracted event argument A_{r_i} is:

$$A_{r_i} = \begin{cases} \emptyset & J_{r_i} = 0 \\ \left\{ \left(s_{r_i}^{(j)}, e_{r_i}^{(j)} \right) \right\}_{j=1}^{J_{r_i}} & J_{r_i} \geq 1 \end{cases}$$

[0079] Here, J_{r_i} is a quantity of event arguments A_{r_i} corresponding to the argument role r_i , and $s_{r_i}^{(j)}$ and $e_{r_i}^{(j)}$ are respectively indexes of a start lemma and an end lemma corresponding to a $J_{r_i}^{\text{th}}$ event argument A_{r_i} in the inputted text X .

[0080] An example in which the inputted text X is "Sam Zhang and Sid Lee are chatting in the People's Park" is still used for description. The argument roles r_i corresponding to event arguments A_{r_i} "Sam Zhang" and "Sid Lee" are both participants. J_{r_i} is the quantity of corresponding event arguments A_{r_i} when the argument role r_i is a participant, that is, 2, corresponding to two event arguments "Sam Zhang" and "Sid Lee". In the inputted text X , "Sam Zhang" is the first event argument, and $s_{r_i}^{(j)}$ and $e_{r_i}^{(j)}$ are respectively indexes of a start lemma and an end lemma of the first event argument "Sam Zhang" in the inputted text X . For example, in the inputted text X , "Zhang" is the first lemma, and "Sam" is the second lemma. In this case, $(s_{r_i}^{(j)}, e_{r_i}^{(j)})$ is (1, 2).

[0081] In some embodiments, the argument feature vector of each event argument may be converted into a start search vector representing a start position and an end search vector representing an end position. The value range corresponding to each event argument is determined in a manner of respectively calculating inner products of the start search vector and the end search vector with a feature vector corresponding to each

lemma in the first text feature vector set.

[0082] In the embodiments of the disclosure, firstly, a first text feature vector set corresponding to an inputted text is extracted through an encoder and a decoder, and then a target extraction template and the inputted text are decoded through the decoder to obtain an argument feature vector of each event argument. Because the target extraction template and the inputted text are decoded together through the decoder, the argument feature vector of each event argument obtained through decoding by the decoder is merged with information of the inputted text. The first text feature vector set includes text feature vectors reflecting context. In this case, a value range that corresponds to each event argument and is determined based on the first text feature vector set and the argument feature vector of each event argument is a value range reflecting the context, so that the accuracy is higher, thereby improving the accuracy of EAE. In addition, because a target extraction template includes argument roles and relationships between the argument roles, the establishment of cross-sentence relationships between a plurality of event arguments is implemented. In this way, the problem of a long distance between event arguments in event extraction in sentence-level and document-level texts is resolved, and a plurality of event arguments may be predicted at once, so that the speed of event extraction is greatly increased, and during the construction of a large-scale event knowledge graph, the construction efficiency and scale of the event knowledge graph are improved. In addition, a plurality of slots are introduced for one argument role in a target extraction template, thereby effectively reducing cases in which event extraction cannot adequately match that one argument role corresponds to a plurality of event arguments, thereby improving the accuracy of matching.

[0083] Embodiments of the disclosure further provide a method for event extraction. As shown in FIG. 2, the method includes S201 to S209.

[0084] In S201: an inputted text is acquired, where the inputted text includes at least one trigger word and at least one event argument.

[0085] In S202, the at least one trigger word of the inputted text is acquired, and the at least one trigger word of the inputted text is marked.

- [0086] Similar to the above, for example, the inputted text is $X = \{x_1, x_2, x_3, \dots, x_n\}$.
- [0087] The inputted text with the trigger word marked is $\hat{X} = \{x_1, x_2, < t >, x_3, x_4, < /t >, \dots, x_n\}$.
- [0088] In S203, the inputted text with the at least one trigger word marked is converted into a text word vector set.
- [0089] A dimensionality of each lemma in the text word vector set is not limited in the embodiments of the disclosure. For example, for a basic BART model, a dimensionality of each lemma may be 768 dimensions. For an advanced BART model, a dimensionality of each lemma may be 1024 dimensions. The basic BART model uses six layers of encoders and decoders. The advanced BART model uses twelve layers of encoders and decoders.
- [0090] In S204, the text word vector set is encoded through the encoder to obtain a second text feature vector set.
- [0091] The second text feature vector set is a feature vector set merged with information of the inputted text after encoding by a self-attention layer in the encoder.
- [0092] An encoder in the BART model is used as an example. An expression of the second text feature vector set is:
- [0093] $H^{(enc)} = \text{BART-Encoder}(\hat{X})$.
- [0094] In S205, the text word vector set and the second text feature vector set are decoded through the decoder to obtain a first text feature vector set corresponding to the inputted text.
- [0095] The first text feature vector set is a feature vector set merged with the information of the inputted text after decoding by a self-attention layer and a cross-attention layer in the decoder.
- [0096] A decoder in the BART model is used as an example. An expression of the first text feature vector set is:

- [0097] $H^{(dec)} = \text{BART-Decoder}(\hat{X}, H^{(enc)})$.
- [0098] In S206, a target extraction template corresponding to the at least one trigger word is acquired.
- [0099] In S207, the target extraction template is converted into a template word vector set.
- [00100] A dimensionality corresponding to each lemma in the template word vector set may be the same as a dimensionality of each lemma in the text word vector set. For example, if the dimensionality of each lemma in the text word vector set is 768 dimensions, the dimensionality corresponding to each lemma in the template word vector set may also be 768 dimensions, to facilitate subsequent operation processing.
- [00101] In S208, the template word vector set and the second text feature vector set are decoded through the decoder to obtain an argument feature vector of each event argument.
- [00102] In S209, a value range corresponding to each event argument is determined based on the first text feature vector set and the argument feature vector of each event argument.
- [00103] For steps S208 and S209, reference may be made to steps S104 and S105.
- [00104] In the embodiments of the disclosure, the inputted text and the target extraction template are respectively converted into the text word vector set and the template word vector set, and are then inputted into the encoder and the decoder, to facilitate processing of data by the encoder and the decoder. In addition, after the text word vector set is inputted into the encoder, the second text feature vector set is outputted by the encoder, and the second text feature vector set is then inputted into the decoder for decoding, to obtain the first text feature vector set and the argument feature vector of each event argument.
- [00105] In some embodiments, the implementation of step S208 "decoding the template word vector set and the second text feature vector set through the decoder to

obtain the argument feature vector of each event argument" includes S2081 and S2082.

[00106] In S2081, the template word vector set and the second text feature vector set are decoded through the decoder to obtain a template feature vector set merged with information of the inputted text.

[00107] Because the template feature vector set is obtained after the decoder decodes the template word vector set and the second text feature vector set, and the second text feature vector set is a feature vector set merged with the information of the inputted text. Therefore, the template feature vector set is also merged with the information of the inputted text.

[00108] A decoder in the BART model is used as an example. An expression of the template feature vector set is:

[00109] $Pt^{(dec)} = \text{BART-Decoder}(Pt^{(t)}, H^{(enc)}),$

[00110] where $Pt^{(t)}$ is the template word vector set.

[00111] In S2082, the argument feature vector of each event argument corresponding to each argument role is determined from the template feature vector set based on position information of each argument role in the template word vector set.

[00112] The template word vector set is obtained by converting the target extraction template into a vector form. The target extraction template includes the position information of each argument role. Therefore, the position information of each argument role in the template word vector set may be known.

[00113] In addition, because the template feature vector set is obtained after the decoder decodes the template word vector set and the second text feature vector set, the argument feature vector of each event argument corresponding to each argument role may be determined from the template feature vector set based on the position information of each argument role in the template word vector set.

[00114] For example, for the target extraction template "participant communicated with participant about topic at place", a position corresponding to the first lemma is a position

of an argument role "participant". After the target extraction template is converted into the template word vector set, the template word vector set may be searched for a vector corresponding to the first lemma, to obtain the position information of the argument role "participant" in the template word vector set.

[00115] The template feature vector set is obtained after the template word vector set is decoded by the decoder. Similarly, a vector in which the argument role "participant" is located in the template word vector set. That is, an argument feature vector of an event argument corresponding to the argument role "participant", may be found in the template feature vector set.

[00116] In some embodiments, the implementation of step S2082 "determining the argument feature vector of each event argument corresponding to each argument role from the template feature vector set based on position information of each argument role in the template word vector set" includes S282a and S282b.

[00117] In S282a, a feature vector of each event argument corresponding to each argument role is extracted from the template feature vector set for the position information of the argument role in the template word vector set.

[00118] In S282b, average pooling is performed on the feature vector of each event argument to obtain the argument feature vector of the event argument.

[00119] The function of average pooling is to average a dimensionality of the feature vector of each event argument along a text length, to keep the dimensionality of the feature vector of each event argument consistent. Each event argument includes a different quantity of lemmas. For example, the inputted text is "Sam Zhang and Sid Lee are chatting in the People's Park". A quantity of lemmas corresponding to the event argument "Sam Zhang" may be 2, and a quantity of lemmas corresponding to the event argument "People's Park" may be 4. Therefore, the dimensionalities of the event arguments "Sam Zhang" and "People's Park" along the text length are respectively 2 and 4. After average pooling, the dimensionalities of the event arguments "Sam Zhang" and "People's Park" along the text length may be 1. That is, after average pooling, the dimensionalities of the event arguments "Sam Zhang" and "People's Park" along the text

length are the same. In some embodiments, the dimensionality of the feature vector of each event argument along the text length may be the same as the dimensionality of each lemma in the first text feature vector set along the text length.

[00120] In the embodiments of the disclosure, average pooling is performed on the feature vector of each event argument, so that the dimensionality of the feature vector of each event argument is the same as a dimensionality of a vector corresponding to each lemma in the first text feature vector set, to facilitate subsequent operation processing.

[00121] In some embodiments, the implementation of step S105 "determining a value range corresponding to each event argument based on the first text feature vector set and the argument feature vector of each event argument" includes S1051 and S1052.

[00122] In step S1051, at least one candidate start-to-end range of each event argument is determined based on a quantity of corresponding lemmas in the first text feature vector set.

[00123] A start position of each event argument in the first text feature vector set should be less than an end position of each event argument in the first text feature vector set.

[00124] Based on this, the at least one candidate start-to-end range of each event argument may be determined according to the quantity of corresponding lemmas in the first text feature vector set. For example, the quantity of corresponding lemmas in the first text feature vector set is 5. The at least one candidate start-to-end range of each event argument may be (1, 2), (1, 3), (1, 4), (1, 5), (2, 3), (2, 4), (2, 5), (3, 4), (3, 5), or (4, 5), where numbers in the parentheses denote indexes of a corresponding lemma in the first text feature vector set, the first number denote an index corresponding to the start position, and the second number denote an index of the end position,

[00125] In step S1052, for each event argument, the value range corresponding to the event argument is determined based on the at least one candidate start-to-end range of the event argument and the argument feature vector of the event argument.

[00126] In some embodiments, as shown in FIG. 3, the implementation of step S1052

"determining the value range corresponding to the event argument based on the at least one candidate start-to-end range of the event argument and the argument feature vector of the event argument" includes S1521 to S1524.

[00127] In step S1521, the argument feature vector of each event argument is converted into a start search vector representing a start position and an end search vector representing an end position.

[00128] During description, the argument feature vector of each event argument is denoted as ψ_k . Expressions of a start search vector ψ_k^s and an end search vector ψ_k^e are:

[00129] $\psi_k^s = \psi_k \circ w^{(s)} \in R^h$, and

[00130] $\psi_k^e = \psi_k \circ w^{(e)} \in R^h$,

[00131] where R^h represents an h-dimensional vector, $w^{(s)}$ and $w^{(e)}$ are two learnable parameters, and are h-dimensional vectors, and $\circ$ denotes elementwise multiplication.

[00132] Because the argument feature vector ψ_k also belongs to R^h , elementwise multiplication may be performed on the argument feature vector ψ_k and $w^{(s)}$ or $w^{(e)}$. The obtained ψ_k^s and ψ_k^e still belong to R^h .

[00133] In step S1522, a first inner product result of a feature vector corresponding to each lemma in the first text feature vector set and each start search vector is determined to obtain a first inner product result set of candidate start positions of each event argument.

[00134] Because the feature vector corresponding to each lemma in the first text feature vector set belongs to R^h and ψ_k^s also belongs to R^h , an inner product may be calculated for the feature vector corresponding to each lemma in the first text feature vector set and the start search vector ψ_k^s to obtain a first inner product result. An inner product is calculated for each event argument, to obtain the first inner product result set of the candidate start positions of each event argument.

[00135] An expression of calculating the first inner product result set is:

- [00136] $\text{logit}_k^s = \psi_k^s H^{(dec)}$.
- [00137] k is each slot in the target extraction template, $\text{logit}_k^s \in R^L$, and L is the quantity of corresponding lemmas in the first text feature vector set.
- [00138] In step S1523, a second inner product result of the feature vector corresponding to each lemma in the first text feature vector set and each end search vector is determined to obtain a second inner product result set of candidate end positions of each event argument.
- [00139] Because the feature vector corresponding to each lemma in the first text feature vector set belongs to R^h and ψ_k^e also belongs to R^h , an inner product may be calculated for the feature vector corresponding to each lemma in the first text feature vector set and the end search vector ψ_k^e to obtain a second inner product result. An inner product is calculated for each event argument, to obtain the second inner product result set of the candidate end positions of each event argument.
- [00140] An expression of calculating the second inner product result set is:
- [00141] $\text{logit}_k^e = \psi_k^e H^{(dec)}$.
- [00142] k is each slot in the target extraction template, $\text{logit}_k^e \in R^L$, and L is the quantity of corresponding lemmas in the first text feature vector set.
- [00143] In step S1524, the value range corresponding to each event argument is determined based on the first inner product result set of the candidate start positions and the second inner product result set of the candidate end positions of each event argument.
- [00144] The value range corresponding to each event argument may be determined by calculating a sum of the first inner product result and the second inner product result in each candidate start-to-end range corresponding to each event argument. In some embodiments, a maximum sum of the first inner product result and the second inner product result in the candidate start-to-end range corresponds to the value range of the corresponding event argument.
- [00145] For example, if the candidate start-to-end range is (1, 2), an inner product is

calculated for the start search vector ψ_k^s and the feature vector corresponding to the first lemma in the first text feature vector set to obtain the first inner product result. An inner product is calculated for the end search vector ψ_k^e and the feature vector corresponding to the second lemma in the first text feature vector set to obtain the second inner product result. The first inner product result and the second inner product result are added. A candidate start-to-end range corresponding to the maximum sum of the first inner product result and the second inner product result is the value range of the corresponding event argument.

[00146] In some embodiments, the value range corresponding to each event argument may be determined in the candidate start-to-end range by using greedy search.

[00147] An expression of determining the value range corresponding to each event argument may be:

$$[00148] \quad (\hat{s}, \hat{e}) = \arg \max_{(i,j) \in L^2} \text{logit}_k^{(s)}(i) + \text{logit}_k^{(e)}(j), i < j,$$

[00149] where $(\hat{s}, \hat{e})$ is a found value range corresponding to each event argument, L is the quantity of corresponding lemmas in the first text feature vector set, i denotes an index of a lemma corresponding to the start position of each event argument in the first text feature vector set, and j denotes an index of a lemma corresponding to the end position of each event argument in the first text feature vector set.

[00150] In the embodiments of the disclosure, an argument feature vector of an event argument is converted into a start search vector representing a start position and an end search vector representing an end position. Then inner products are respectively calculated for the start search vector and the end search vector and a feature vector corresponding to each lemma in the first text feature vector set to obtain a first inner product result and a second inner product result. A start-to-end range with the largest sum of the first inner product result and the second inner product result is determined in the at least one candidate start-to-end range by using greedy search, to obtain a start-to-end range corresponding to the event argument. Finally, a start-to-end range of each event argument is sequentially determined, to obtain start-to-end ranges of all event arguments. In this way, accurate extraction of event arguments is implemented, and the speed of

extraction is also increased.

[00151] Embodiments of the disclosure provide a method for training an event extraction model. The event extraction model includes an encoder, a decoder, and an argument detection module. As shown in FIG. 4, the method includes S401 to S407.

[00152] In S401, a training text and a target extraction template corresponding to the training text are acquired.

[00153] In S402, a first text feature vector set corresponding to the training text is extracted through the encoder and the decoder.

[00154] In S403, the target extraction template and the training text are encoded through the decoder to obtain an argument feature vector of each event argument.

[00155] In S404, a prediction range corresponding to each event argument is determined based on the first text feature vector set and the argument feature vector of each event argument through the argument detection module.

[00156] For the implementation of step S401 to step S404, reference may be made to step S101 to step S105.

[00157] In S405, the prediction range of each event argument is matched against a true value range to obtain an optimal prediction range that has a loss function value satisfying a second preset condition and corresponds to each event argument.

[00158] The true value range is an accurate range corresponding to each event argument. In some embodiments, the prediction range of each event argument may be matched against the true value range through a bipartite graph matching method, to obtain different matching solutions, and losses corresponding to different matching solutions are evaluated by using a loss function.

[00159] The second preset condition may be that the loss function value is minimal. Based on a matching solution corresponding to the minimum loss function value, the optimal prediction range corresponding to each event argument may be obtained.

- [00160] In S406, a loss function value of the event extraction model is determined based on the optimal prediction range of each event argument.
- [00161] The loss function value of the event extraction model may be determined by using a cross-entropy method.
- [00162] In S407, the event extraction model is trained based on the loss function value.
- [00163] When the loss function value is smaller, the effect of the event extraction model is better. In some embodiments, parameters of the model may be optimized by using a gradient descent method, thereby improving the effect of the model.
- [00164] In some embodiments, the implementation of step S405 "matching the prediction range of each event argument against a true value range to obtain an optimal prediction range that has a loss function value satisfying a second preset condition and corresponds to each event argument" includes S4051 to S4053.
- [00165] In S4051, the prediction range of each event argument is matched against the true value range by using a bipartite graph matching method to obtain a candidate matching group, where the candidate matching group includes at least one matching pair, and one matching pair includes a prediction range and a true value range of one event argument.
- [00166] A true value range of all event arguments in data z (that is, an inputted text X) may be denoted as $y_r^z = [(s_0, e_0), (s_1, e_1), (s_2, e_2), \dots, (s_n, e_n)]$, where r is an argument role.
- [00167] A prediction range of all event arguments in the data z is denoted as $\hat{y}_r^z = [(\hat{s}_0, \hat{s}_0), (\hat{s}_1, \hat{s}_1), (\hat{s}_2, \hat{s}_2), \dots, (\hat{s}_m, \hat{s}_m)]$, where m is a quantity of times that the argument role r appears in the target extraction template, that is, a quantity of slots in the target extraction template.
- [00168] The prediction range of each event argument is matched against the true value range by using the bipartite graph matching method to obtain a candidate matching group. For example, $y_r^z = [(s_0, e_0), (s_1, e_1)]$, and $\hat{y}_r^z = [(\hat{s}_0, \hat{s}_0), (\hat{s}_1, \hat{s}_1)]$. Two candidate

matching groups may be obtained. The first candidate matching group is: (s_0, e_0) corresponds to $(\hat{s}_0, \hat{s}_0)$, and (s_1, e_1) corresponds to $(\hat{s}_1, \hat{s}_1)$. The second candidate matching group is: (s_0, e_0) corresponds to $(\hat{s}_1, \hat{s}_1)$, and (s_1, e_1) corresponds to $(\hat{s}_0, \hat{s}_0)$. In the first candidate matching group, (s_0, e_0) and $(\hat{s}_0, \hat{s}_0)$ are one matching pair, and (s_1, e_1) and $(\hat{s}_1, \hat{s}_1)$ are one matching pair.

[00169] In S4052, a loss function value of each candidate matching group is determined, where the loss function value is used for representing a sum of distances in all matching pairs in the candidate matching group.

[00170] The loss function value may be $\sum_i^N L_1((s, e)_i, (\hat{s}, \hat{e})_{\sigma(i)})$, used for representing a sum of distances in all matching pairs in the candidate matching group. i is an index number of (s, e) in the true value range y_r^Z . $\sigma(i)$ is an index number of $(\hat{s}, \hat{e})$ in the prediction range corresponding to an i^{th} (s, e) in the true value range. For example, the first (s, e) in the true value range matches the second $(\hat{s}, \hat{e})$ in the prediction range. $\sigma(i)$ is 2, and $((s, e)_i, (\hat{s}, \hat{e})_{\sigma(i)})$ is $((s, e)_1, (\hat{s}, \hat{e})_2)$.

[00171] When a quantity n of (s, e) in the true value range is greater than a quantity m of $(\hat{s}, \hat{e})$ in the prediction range and N is m , only the matched (s, e) and $(\hat{s}, \hat{e})$ are used in the loss function. When the quantity n of (s, e) in the true value range is less than the quantity m of $(\hat{s}, \hat{e})$ in the prediction range, $(0, 0)$ is inserted in the true value range, to enable a quantity of (s, e) in the true value range to be equal to quantity of $(\hat{s}, \hat{e})$ in the true value range. For example, the quantity m of $(\hat{s}, \hat{e})$ in the prediction range is 5, and the quantity n of (s, e) in the true value range is 3, so that two $(0, 0)$ are inserted in the true value range, to represent a case that there is no answer, to avoid overprediction.

[00172] In S4053, a candidate matching group that has the loss function value satisfying the second preset condition is selected to obtain the optimal prediction range corresponding to each event argument.

[00173] The second preset condition may be that the loss function value is minimal, a candidate matching group corresponding to the minimum loss function value is selected, to obtain the optimal prediction range corresponding to each event argument.

[00174] An expression of selecting a candidate matching group that has the loss function value satisfying the second preset condition is:

$$[00175] \quad \hat{\sigma} = \operatorname{argmin}_{\sigma \in \Gamma_N} \sum_i^N L_1((s, e)_i, (\hat{s}, \hat{e})_{\sigma(i)}),$$

[00176] where i is an index number of (s, e) in the true value range y_r^Z , $\sigma(i)$ is an index number of $(\hat{s}, \hat{e})$ in the prediction range corresponding to an i th (s, e) in the true value range, and Γ_N is a matching manner of $N(s, e)$ and $N(\hat{s}, \hat{e})$.

[00177] In some embodiments, the implementation of step S406 "determining a loss function value of the event extraction model based on the optimal prediction range of each event argument" includes S4061 to S4063.

[00178] In S4061, a first inner product result corresponding to a start position and a second inner product result corresponding to an end position of the optimal prediction range of each event argument are acquired.

[00179] During description, for example, the optimal prediction range corresponding to the event argument $(8,9)_2$ is $(8,10)_1$. A first inner product result corresponding to a start position 8 and a second inner product result corresponding to an end position 10 in $(8,10)_1$ are acquired.

[00180] In S4062, the first inner product result and the second inner product result are normalized.

[00181] The acquired first inner product result and second inner product result may be normalized by using a softmax function.

[00182] In S4063, a loss function value of a model which the method is used for is determined by using the normalized first inner product result and second inner product result.

[00183] The loss function value of the model may be calculated by using cross entropy. In some embodiments, an expression of a loss function corresponding to one argument role r may be:

[00184]
$$L_{bi}(y_r^z, \hat{y}_r^z) = \sum_{i=1}^N \frac{-\log \hat{p}_{\hat{\sigma}(i)}^s(s_i) - \log \hat{p}_{\hat{\sigma}(i)}^e(e_i)}{2},$$

[00185] where r is the argument role, z is data, $\hat{p}_{\hat{\sigma}(i)}^s(s_i)$ denotes a value (that is, a result obtained in step S4062) after the first inner product result corresponding to a start position s_i of the optimal prediction range is normalized, $\log \hat{p}_{\hat{\sigma}(i)}^e(e_i)$ denotes a value (that is, a result obtained in step S4062) after the second inner product result corresponding to an end position e_i of the optimal prediction range is normalized, and i is an index number of (s, e) in the true value range y_r^z .

[00186] Correspondingly, an expression of a loss function of the model which the method is used for may be:

[00187]
$$L(y, \hat{y}) = \sum_z \sum_r L_{bi}(y_r^z, \hat{y}_r^z),$$
 where z is data, and r is the argument role.

[00188] That is, loss function values of all argument roles in all data are calculated.

[00189] In this embodiment of the disclosure, a prediction range that best matches a true value range is found by using a bipartite graph matching loss method. In this way, a case of a matching error because one argument role corresponds to a plurality of event arguments can be further mitigated. In addition, a loss function value of the entire model is determined based on the optimal match and by using the cross-entropy method, to test the validity of the model, thereby facilitating training and adjustment of the model.

[00190] Embodiments of the disclosure further provide a method for training an event extraction model. As shown in FIG. 5, the method includes S501 to S509.

[00191] In step S501, a training text and a target extraction template corresponding to the training text are acquired.

[00192] The training text is "In the first debate, Li Ming disputed Han Meimei's comment that stop and frisk was ruled unconstitutional in New York".

[00193] In some embodiments, a trigger word may be marked from the training text, so that subsequently a decoder effectively extracts a first text feature vector set corresponding to the training text.

- [00194] The training text with the trigger word marked is:
- [00195] "In the first <t>debate</t>, Li Ming disputed Han Meimei's comment that stop and frisk was ruled unconstitutional in New York".
- [00196] The target extraction template corresponding to the training text is "participant communicated with participant about topic at place". A participant 1, a participant 2, a topic, and a place correspond to argument roles.
- [00197] In some embodiments, the training text with the trigger word marked and the target extraction template corresponding to the training text may be converted into a vector form, to facilitate operation by the encoder and the decoder. The training text that is marked with the trigger word and is in the vector form after conversion is a text word vector set $\hat{X}$. The target extraction template in a vector form after conversion is a template word vector set $Pt^{(t)}$.
- [00198] In step S502, a text word vector set is encoded through the encoder to obtain a second text feature vector set.
- [00199] For step S502, reference may be made to step S204.
- [00200] In step S503, the text word vector set and the second text feature vector set are decoded through a decoder to obtain a first text feature vector set corresponding to an inputted text;
- [00201] For step S503, reference may be made to step S205. 101 in FIG. 5 is the first text feature vector set.
- [00202] In step S504, the target extraction template in a vector form after conversion and the second text feature vector set are decoded by using the decoder to obtain a template feature vector set merged with information of the inputted text.
- [00203] For step S504, reference may be made to step S2081. 102 in FIG. 5 is the template feature vector set merged with the information of the inputted text.
- [00204] In step S505, an argument feature vector of each event argument corresponding

to each argument role from the template feature vector set is determined based on position information of each argument role in the target extraction template in a vector form after conversion.

[00205] In step S505, reference may be made to step S2082. In the target extraction template "participant 1 communicated with participant 2 about topic at place", positions corresponding to the first three lemmas are positions of an argument role "participant 1". After the target extraction template is converted into the template word vector set, the template word vector set may be searched for vectors corresponding to the first three lemmas, to obtain the position information of the argument role "participant 1" in the template word vector set.

[00206] The template feature vector set is obtained after the template word vector set is decoded by the decoder. Similarly, a vector in which the argument role "participant 1" is located in the template word vector set may be found in the template feature vector set, that is, an argument feature vector of an event argument corresponding to the argument role "participant 1", to obtain the argument feature vector of each event argument.

[00207] In some embodiments, the implementation of step S505 may include the following steps.

[00208] In S505a, a feature vector of each event argument corresponding to each argument role from the template feature vector set is extracted based on the position information of each argument role in the template word vector set.

[00209] In S505b, average pooling is performed on the feature vector of each event argument to obtain the argument feature vector of the event argument.

[00210] For step S505a and step S505b, reference may be made to step S282a and step S282b.

[00211] 103 in FIG. 5 is an argument feature vector of each event argument, where a numeral 1 is an argument feature vector of an event argument corresponding to the first argument role "participant 1" in the target extraction template, a numeral 2 is an argument feature vector of an event argument corresponding to the second argument role

"participant 2" in the target extraction template, a numeral 3 is an argument feature vector of an event argument corresponding to the third argument role "topic" in the target extraction template, and a numeral 4 is an argument feature vector of an event argument corresponding to the fourth argument role "place" in the target extraction template.

[00212] In step S506, a prediction range corresponding to each event argument is determined based on the first text feature vector set and the argument feature vector of each event argument.

[00213] The implementation of step S506 may include the following steps.

[00214] In step S5061, at least one candidate start-to-end range of each event argument is determined based on a quantity of corresponding lemmas in the first text feature vector set.

[00215] For step S5061, reference may be made to step S1051.

[00216] In step S5062, the argument feature vector of each event argument is converted into a start search vector representing a start position and an end search vector representing an end position.

[00217] In step S5063, a first inner product result of a feature vector corresponding to each lemma in the first text feature vector set and each start search vector is determined to obtain a first inner product result set of candidate start positions of each event argument.

[00218] In step S5064, a second inner product result of the feature vector corresponding to each lemma in the first text feature vector set and each end search vector is determined to obtain a second inner product result set of candidate end positions of each event argument.

[00219] In step S5065, the prediction range corresponding to each event argument is determined based on the first inner product result set of the candidate start positions and the second inner product result set of the candidate end positions of each event argument.

[00220] For step S5062 to step S5065, reference may be made to step S1521 to step

S1524.

[00221] In some embodiments, a maximum sum of the first inner product result and the second inner product result in the candidate start-to-end range corresponds to the prediction range of the corresponding event argument.

[00222] Step S506 corresponds to "multi-event argument value range prediction" in FIG. 5. As can be known from figure, obtained prediction ranges are (8, 10), (6, 7), (11, 11), and (18, 20). A position number corresponding to a prediction range is an index of a corresponding lemma in the first text feature vector set. (8, 10), (6, 7), (11, 11), and (18, 20) respectively correspond to "Han Meimei", "Li Ming", "stop and frisk", and "New York". An event argument extracted for the first argument role "participant 1" is "Han Meimei". An event argument extracted for the second argument role "participant 2" is "Li Ming". An event argument extracted for the third argument role "topic" is "stop and frisk". An event argument extracted for the fourth argument role "place" is "New York". It can be seen that event arguments extracted for the argument roles "participant 1" and "participant 2" in the prediction range are interchanged. In fact, the event argument extracted for the first argument role "participant 1" should be "Li Ming", and the event argument extracted for the second argument role "participant 2" should be "Han Meimei".

[00223] In step S507, the prediction range of each event argument is matched against a true value range to obtain an optimal prediction range that has a loss function value satisfying a second preset condition and corresponds to each event argument.

[00224] In some embodiments, the implementation of step S507 may include the following steps.

[00225] In S5071, the prediction range of each event argument is matched against the true value range by using a bipartite graph matching method to obtain a candidate matching group, where the candidate matching group includes at least one matching pair, and one matching pair includes a prediction range and a true value range of one event argument.

[00226] A true value range of all event arguments in the training text is denoted as $y_r^k =$

$[(6,7)_1, (8,9)_2, (11,14)_3, \text{ and } (18,20)_4]$, where r is the argument role.

[00227] A prediction range of all event arguments in the training text is denoted as $\hat{y}_r^z$
= $[(8,10)_1, (6,7)_2, (11,11)_3, \text{ and } (18,20)_4]$.

[00228] The prediction range of each event argument is matched against the true value range by using the bipartite graph matching method to obtain N candidate matching groups. For example, one candidate matching group may be: $(6,7)_1$ in the true value range is paired with $(6,7)_2$ in the prediction range, $(8,9)_2$ in the true value range is paired with $(8,10)_1$ in the prediction range, $(11,14)_3$ in the true value range is paired with $(11,11)_3$ in the prediction range, $(18,20)_4$ in the true value range is paired with $(18,20)_4$ in the prediction range, and $(6,7)_1$ in the true value range and $(6,7)_2$ in the prediction range are one matching pair.

[00229] In S5072, a loss function value of each candidate matching group is determined, where the loss function value is used for representing a sum of distances in all matching pairs in the candidate matching group.

[00230] In S5073, a candidate matching group that has the loss function value satisfying the second preset condition is selected to obtain the optimal prediction range corresponding to each event argument.

[00231] For step S5072 and step S5073, reference may be made to step S4052 and step S4053.

[00232] As shown in FIG. 5, for the training text "In the first debate, Li Ming disputed Han Meimei's comment that stop and frisk was ruled unconstitutional in New York", true value ranges are (6, 7), (8, 9), (11, 14), and (18, 20). By means of the foregoing bipartite graph matching loss method, it is obtained that optimal prediction ranges are (6, 7), (8, 10), (11, 11), and (18, 20). That is, the event argument extracted for the first argument role "participant 1" corresponds to "Li Ming", and the event argument extracted for the second argument role "participant 2" corresponds to "Han Meimei". Mismatching of the argument roles "participant 1" and "participant 2" in step S506 is changed. In this way, a case of a matching error because one argument role corresponds to a plurality of event

arguments can be effectively mitigated.

[00233] In S508, a loss function value of the event extraction model is determined based on the optimal prediction range of each event argument.

[00234] In some embodiments, the implementation of step S508 may include the following steps.

[00235] In S5081, a first inner product result corresponding to a start position and a second inner product result corresponding to an end position of the optimal prediction range of each event argument are acquired.

[00236] In S5082, the first inner product result and the second inner product result are normalized.

[00237] In S5083, a loss function value of a model which the method is used for is determined by using the normalized first inner product result and second inner product result.

[00238] For step S5081 to step S5083, reference may be made to step S4061 to step S4063.

[00239] In S509, the event extraction model is trained based on the loss function value.

[00240] In step S509, reference may be made to step S407.

[00241] Based on the foregoing method, corresponding verification is performed for three data sets ACE 2005, RAMS, and WIKIEVENT.

[00242] ACE 2005 focuses on event extraction from a sentence-level text (Sentence-Level EAE, (S-EAE)), and is mainly from news articles. RAMS and WIKIEVENT focus on event extraction from a document-level text (Document-Level EAE, (D-EAE)), and are respectively from news and Wikipedia. Table 2 shows detailed statistical data in a process of event extraction for different data sets. Data set preprocessing and division method in the related art are followed. However, the labeling of any entity is not required.

[00243] Table 2

Data set	ACE2005	RAMS	WIKIEVENT
Quantity of event arguments			
Training set	4859	17026	4552
Verification set	605	2188	428
Test set	576	2023	566
Event type	33	139	50
Argument role	36	65	59
Quantity of event arguments of each event	1.19	2.33	1.40

[00244] A standard F1 score (Arg-I) for element recognition and a standard F1 score (Arg-C) for element classification are used as evaluation indicators in the embodiments of the disclosure. Arg-I is used for measuring a position of a prediction range, and Arg-C is used for measuring whether an argument role is accurately predicted. For WIKIEVENT, a head F1 is further evaluated, and only matching of heads is involved.

[00245] In a specific method, a basic PLM or an advanced PLM is separately trained on single graphics cards NVIDIA V100 and 1080Ti. AdamW is selected as an optimization method in the embodiments of the disclosure, and a linear learning rate with a warm up rate of 0.1 is used as a scheduling policy. Because documents occupy large memory, a quantity of batch processing is reduced, and the maximum number of training iterations is increased from 10000 to 20000. A window size of approximately 500 trigger words is used to train the model to adapt to a Graphics Processing Unit (GPU). A miss rate of true value labeling in a training process is less than 0.1%. For fair comparison, all documents in RAMS are used for evaluation. Learning rates of S-EAE and D-EAE are respectively $4e^{-5}$ and $3e^{-5}$.

[00246] Benchmark: The reason of selecting the following benchmarks is that the benchmarks have implemented the most advanced performance in EAE, and are also based on a PLM.

[00247] 1. EEQA is the first (machine) reading comprehension (Question Answering, QA)-based model in an S-EAE task.

[00248] 2. FEAE considers event argument interaction through knowledge distillation, and extends a QA-based method to D-EAE.

[00249] 3. ONEIE jointly extracts an entity, a relationship, and an event, and different from a QA-based model, extracted entities are used as candidate parameters.

[00250] 4. BART-Gen generates event arguments in an order-to-order model sequence, rather than to extract a start-to-end range of event arguments from articles.

[00251] Table 3 shows overall performance of different models.

[00252] Table 3

Model	PLM	ACE05		RAMS		WIKIEVENT		
		Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C	Head-C
FEAE	BERT-B			53.49	47.40			
EEQA*	BERT-B	66.32	62.08	46.18	39.61	51.61	48.61	
EEQA*	BERT-L	68.04	64.57	46.23	41.15	52.04	48.47	
EEQA*	BART-B	67.91	66.54	47.11	43.77	56.41	52.27	
ONEIE*	BERT-B	69.42	66.90					
BART-Gen	BART-L	69.92	66.67		48.64	57.32		64.57
PAIE-B	BART-B	73.77	70.98	53.64	50.40	61.08	63.68	67.11

PAIE-L	BART-L	75.68	73.28	55.59	53.04	69.61	65.67	69.23
--------	--------	-------	-------	-------	-------	-------	-------	-------

[00253] BERT-B is a basic version of BERT, BERT-L is an advanced version of BERT, BART-B is a basic version of BART, BART-L is an advanced version of BART, PAIE-B is a basic version of the model provided in the embodiments of the disclosure, and PAIE-L is an advanced version of the model provided in the embodiments of the disclosure. Arg-I is used for measuring a position of a prediction range, Arg-C is used for measuring whether an argument role is accurately predicted, and Head-C is used for measuring the matching of a head.

[00254] When values of Arg-I, Arg-C, and Head-C in different data sets (ACE05, RAMS, and WIKIEVENT) corresponding to different models are larger, it indicates that the accuracy of prediction is higher. That is, when the value of Arg-I is larger, the prediction range is more accurate. When the value of Arg-C is larger, a probability that an argument role is accurately predicted is higher. When the value of Head-C is larger, the matching of the head is more accurate.

[00255] It may be observed that: (1) In all data sets including an S-EAE task and a D-EAE task, PAIE has the best performance as compared with a benchmark method, this proves that the method provided in the embodiments of the disclosure has adequate generalization. The method inherits advantages of a basic QA-based EAE model, and successfully resolves the problem of a long distance between event arguments in event extraction in document-level texts in particular. (2) As a backbone Network of a PLM, the performance of BART is much better than that of BERT. The method in the embodiments of the disclosure may further implement an improvement through BART-L, indicating that decoding using a template can achieve adequate performance. (3) Considering that models (FEAE, BARTGen, and PAIE) of event argument interaction has better performance than other models, especially in a D-EAE task. This point matches the expectation of the embodiments of the disclosure. That is, a cross-sentence event argument requires long-distance dependence, and implicit or explicit introduction of dependence relationship information between event arguments improves the accuracy of model prediction.

[00256] Based on the foregoing embodiments, embodiments of the disclosure provide an event extraction apparatus. The modules included in the apparatus, the sub-modules included in the modules, the units included in the sub-modules, and the subunits included in the units may all be implemented by an electronic device, or certainly may be implemented by a specific logical circuit. During implementation, a processor may be a Central Processing Unit (CPU), a Microprocessor Unit (MPU), a Digital Signal Processor (DSP), a Field Programmable Gate Array (FPGA), or the like.

[00257] FIG. 6 is a schematic structural diagram of an event extraction apparatus according to an embodiment of the disclosure. As shown in FIG. 6, an event extraction apparatus 600 includes a first acquisition module 601, a first extraction module 602, a second acquisition module 603, a decoding module 604, and a first determination module 605.

[00258] The first acquisition module 601 is configured to acquire an inputted text, where the inputted text includes at least one trigger word and at least one event argument.

[00259] The first extraction module 602 is configured to extract a first text feature vector set corresponding to the inputted text through an encoder and a decoder.

[00260] The second acquisition module 603 is configured to acquire a target extraction template corresponding to the at least one trigger word.

[00261] The first decoding module 604 is configured to decode the target extraction template and the inputted text by using the decoder to obtain an argument feature vector of each event argument.

[00262] The first determination module 605 is configured to determine a value range corresponding to each event argument based on the first text feature vector set and the argument feature vector of each event argument.

[00263] In some embodiments, the apparatus further includes: a conversion module 606 configured to respectively convert the inputted text and the target extraction template into a text word vector set and a template word vector set.

- [00264] In some embodiments, the first extraction module 602 includes: an encoding sub-module, configured to encode the text word vector set by using the encoder to obtain a second text feature vector set; and a first decoding sub-module, configured to decode the text word vector set and the second text feature vector set by using the decoder to obtain the first text feature vector set corresponding to the inputted text.
- [00265] Correspondingly, the first decoding module 604 includes: a second decoding sub-module, configured to decode the template word vector set and the second text feature vector set by using the decoder to obtain the argument feature vector of each event argument.
- [00266] In some embodiments, the second decoding sub-module includes: a decoding unit, configured to decode the template word vector set and the second text feature vector set by using the decoder to obtain a template feature vector set merged with information of the inputted text; and a first determination unit, configured to determine the argument feature vector of each event argument corresponding to each argument role from the template feature vector set based on position information of each argument role in the template word vector set.
- [00267] In some embodiments, the first determination unit includes: an extraction subunit, configured to extract a feature vector of each event argument corresponding to each argument role from the template feature vector set for the position information of the argument role in the template word vector set; and a pooling subunit, configured to perform average pooling on the feature vector of each event argument to obtain the argument feature vector of the event argument.
- [00268] In some embodiments, the apparatus further includes: a second determination module, configured to determine a target event type based on the trigger word; and a third determination module, configured to determine the target extraction template from a preset template library based on the target event type.
- [00269] In some embodiments, the apparatus further includes: a fourth determination module, configured to determine an event type set corresponding to a text set, where the event type set includes at least one event type; a fifth determination module, configured

to determine at least two argument roles in each event type and relationship information between the at least two argument roles; a splicing module, configured to splice the at least two argument roles based on the relationship information between the at least two argument roles for each event type to obtain an extraction template of each event type; and a formation module, configured to form the preset template library based on the extraction template of each event type in the event type set.

[00270] In some embodiments, the apparatus further includes: a third acquisition module, configured to: acquire the at least one trigger word of the inputted text, and mark the at least one trigger word of the inputted text. Correspondingly, the first extraction module 602 is configured to extract the first text feature vector set corresponding to the marked inputted text by using the encoder and the decoder.

[00271] In some embodiments, the first determination module 605 includes: a first determination sub-module, configured to determine at least one candidate start-to-end range of each event argument based on a quantity of corresponding lemmas in the first text feature vector set; and a second determination sub-module, configured to determine, for each event argument, the value range corresponding to the event argument based on the at least one candidate start-to-end range of the event argument and the argument feature vector of the event argument.

[00272] In some embodiments, the second determination sub-module includes: a conversion unit, configured to convert the argument feature vector of each event argument into a start search vector representing a start position and an end search vector representing an end position; a second determination unit, configured to determine a first inner product result of a feature vector corresponding to each lemma in the first text feature vector set and each start search vector to obtain a first inner product result set of candidate start positions of each event argument; a third determination unit, configured to determine a second inner product result of the feature vector corresponding to each lemma in the first text feature vector set and each end search vector to obtain a second inner product result set of candidate end positions of each event argument; and a fourth determination unit, configured to determine the value range corresponding to each event argument based on the first inner product result set of the candidate start positions and

the second inner product result set of the candidate end positions of each event argument.

[00273] FIG. 7 is a schematic structural diagram of an apparatus for training an event extraction model according to an embodiment of the disclosure. As shown in FIG. 7, the apparatus 700 for training an event extraction model includes:

[00274] a fourth acquisition module 701, configured to acquire a training text and a target extraction template corresponding to the training text;

[00275] a second extraction module 702, configured to extract a first text feature vector set corresponding to the training text through an encoder and a decoder;

[00276] a second encoding module 703, configured to encode the target extraction template and the training text through the decoder to obtain an argument feature vector of each event argument;

[00277] a sixth determination module 704, configured to determine a prediction range corresponding to each event argument based on the first text feature vector set and the argument feature vector of each event argument through an argument detection module;

[00278] a matching module 705, configured to match the prediction range of each event argument against a true value range to obtain an optimal prediction range that has a loss function value satisfying a second preset condition and corresponds to each event argument;

[00279] a seventh determination module 706, configured to determine a loss function value of the event extraction model based on the optimal prediction range of each event argument; and

[00280] a training module 707, configured to train the event extraction model based on the loss function value.

[00281] In some embodiments, the matching module 705 includes: a matching sub-module, configured to match the prediction range of each event argument against the true value range by using a bipartite graph matching method to obtain a candidate matching group, where the candidate matching group includes at least one matching pair, and one

matching pair includes a prediction range and a true value range of one event argument; a third determination sub-module, configured to determine a loss function value of each candidate matching group, where the loss function value is used for representing a sum of distances in all matching pairs in the candidate matching group; and a selection sub-module, configured to select a candidate matching group that has the loss function value satisfying the second preset condition to obtain the optimal prediction range corresponding to each event argument.

[00282] In some embodiments, the seventh determination module 706 includes: an acquisition sub-module, configured to acquire a first inner product result corresponding to a start position and a second inner product result corresponding to an end position of the optimal prediction range of each event argument; a normalization sub-module, configured to normalize the first inner product result and the second inner product result; and a fourth determination sub-module, configured to determine, based on the normalized first inner product result and second inner product result, a loss function value of a model which the method is used for.

[00283] The above description of the apparatus embodiments is similar to the description of the method embodiments above, and has beneficial effects similar to those of the method embodiments. For technical details not disclosed in the apparatus embodiments of the disclosure, refer to the description of the method embodiments of the disclosure for comprehension.

[00284] It needs to be noted that in the embodiments of the disclosure, when the foregoing event extraction method and the related training method are implemented in the form of a software functional unit and sold or used as an independent product, the method may be stored in a computer-readable storage medium. Based on such an understanding, the technical solutions in the embodiments of the disclosure essentially, or the part contributing to the prior art may be implemented in the form of a software product. The software product is stored in the storage medium and includes several instructions for instructing an electronic device (which may be a personal computer, a server or the like) to perform all or some of the steps of the method described in the embodiments of the disclosure. The foregoing storage medium includes various media

that can store program code, such as a Universal Serial Bus (USB) flash drive, a removable hard disk, a Read-Only Memory (ROM), a magnetic disk or an optical disc. In this way, the embodiments of the disclosure are not limited to any specific combination of hardware and software.

[00285] Correspondingly, embodiments of the disclosure provide an electronic device, including a memory and a processor, the memory storing computer programs executable on the processor, the processor executing the programs to implement the steps in the event extraction method and the related training method provided in the foregoing embodiments.

[00286] Correspondingly, embodiments of the disclosure provide a readable storage medium, storing computer programs, where a processor is configured to execute the computer programs to implement the steps in the foregoing event extraction method and related training method.

[00287] It needs to be noted that the above description of the embodiments of the storage medium and platform is similar to the description of the method embodiments above, and has beneficial effects similar to those of the method embodiments. For technical details not disclosed in the embodiments of the storage medium and platform of the disclosure, refer to the description of the method embodiments of the disclosure for comprehension.

[00288] It needs to be noted that FIG. 8 is a schematic diagram of a hardware entity of an electronic device according to an embodiment of the disclosure. As shown in FIG. 8, a hardware entity of an electronic device 800 includes a processor 801, a communication interface 802, and a memory 803.

[00289] The processor 801 usually controls the overall operations of the electronic device 800.

[00290] The communication interface 802 may enable the electronic device 800 to communicate with another platform or electronic device or server through a network.

[00291] The memory 803 is configured to store instructions and applications

executable by the processor 801, and may also cache data (for example, image data, audio data, voice communication data, and video communication data) to be processed or already processed by the processor 801 and the modules in the electronic device 800, either through a flash memory or a Random Access Memory (RAM).

[00292] Embodiments of the disclosure further provide a computer program product, comprising computer readable codes, wherein when the computer readable codes run in a device, a processor in the device executes instructions for implementing the steps in the foregoing method.

[00293] In several embodiments provided in the disclosure, it should be understood that the disclosed device and method may be implemented in other forms. The described device embodiments are merely an example. For example, the unit division is merely logical function division and may be other division in actual implementation. For example, a plurality of units or components may be combined or integrated into another system, or some features may be ignored or not performed. In addition, the shown or discussed mutual couplings or direct couplings or communication connections between the components may be implemented through some interfaces, indirect couplings or communication connections between the apparatuses or units, or electrical connections, mechanical connections, or connections in other forms.

[00294] The units described as separate parts may or may not be physically separate, and parts shown as units may or may not be physical units, that is, may be located in one position, or may be distributed on a plurality of network units. Some or all of the units may be selected according to actual needs to achieve the objects of the solutions in the embodiments.

[00295] In addition, functional units in the embodiments of the disclosure may all be integrated into one processing module, or each of the units may exist alone, or two or more units are integrated into one unit. The integrated unit may be implemented in the form of hardware or a software functional unit in addition to hardware. A person of ordinary skill in the art may understand that all or some of the steps in implementing the foregoing method embodiments may be accomplished by a program instructing relevant hardware. The foregoing program may be stored in a computer-readable storage medium.

The problem, when executed, performs the steps in the foregoing method embodiments. The foregoing storage medium includes various media that can store program code such as a removable storage device, a ROM, a RAM, a magnetic disk or an optical disc.

[00296] The methods disclosed in several method embodiments provided in the disclosure may be arbitrarily combined with each other without causing any conflict to obtain new method embodiments.

[00297] The features disclosed in several product embodiments provided in the disclosure may be arbitrarily combined with each other without causing any conflict to obtain new product embodiments.

[00298] The features disclosed in several method or device embodiments provided in the disclosure may be arbitrarily combined with each other without causing any conflict to obtain new method embodiments or device embodiments.

[00299] The foregoing descriptions are merely specific implementations of the disclosure, but are not intended to limit the protection scope of the disclosure. Any variation or replacement that may be readily figured out by a person skilled in the art within the technical scope disclosed in the disclosure shall fall within the protection scope of the disclosure. Therefore, the protection scope of the disclosure shall be subject to the protection scope of the claims.

CLAIMS

1. A method for event extraction, comprising:

acquiring an inputted text, wherein the inputted text comprises at least one trigger word and at least one event argument;

extracting, through an encoder and a decoder, a first text feature vector set corresponding to the inputted text;

acquiring a target extraction template corresponding to the at least one trigger word;

decoding, through the decoder, the target extraction template and the inputted text to obtain an argument feature vector of each event argument; and

determining, based on the first text feature vector set and the argument feature vector of each event argument, a value range corresponding to each event argument.

2. The method according to claim 1, further comprising:

respectively converting the inputted text and the target extraction template into a text word vector set and a template word vector set, wherein

the extracting, through the encoder and the decoder, the first text feature vector set corresponding to the inputted text comprises:

encoding the text word vector set through the encoder to obtain a second text feature vector set; and

decoding the text word vector set and the second text feature vector set through the decoder to obtain the first text feature vector set corresponding to the inputted text; and

correspondingly, the decoding, through the decoder, the target extraction template and the inputted text to obtain the argument feature vector of each event argument comprises:

decoding, through the decoder, the template word vector set and the second text feature vector set to obtain the argument feature vector of each event argument.

3. The method according to claim 2, wherein the decoding, through the decoder, the template word vector set and the second text feature vector set to obtain the argument feature vector of each event argument comprises:

decoding, by using the decoder, the template word vector set and the second text feature vector set to obtain a template feature vector set merged with information of the inputted text; and

determining, from the template feature vector set, the argument feature vector of each event argument corresponding to each argument role based on position information of each argument role in the template word vector set.

4. The method according to claim 3, wherein the determining, from the template feature vector set, the argument feature vector of each event argument corresponding to each argument role based on position information of each argument role in the template word vector set comprises:

for the position information of each argument role in the template word vector set, extracting, from the template feature vector set, a feature vector of each event argument corresponding to the argument role ; and

performing average pooling on the feature vector of each event argument to obtain the argument feature vector of the event argument.

5. The method according to any one of claims 1 to 4, further comprising:

determining a target event type based on the at least one trigger word; and

determining the target extraction template from a preset template library based on the

target event type.

6. The method according to claim 5, further comprising:

determining an event type set corresponding to a text set, wherein the event type set comprises at least one event type;

determining at least two argument roles in each event type and relationship information between the at least two argument roles;

for each event type, splicing the at least two argument roles based on the relationship information between the at least two argument roles to obtain an extraction template of each event type; and

forming the preset template library based on the extraction template of each event type in the event type set.

7. The method according to any one of claims 1 to 4, further comprising:

acquiring the at least one trigger word of the inputted text, and marking the at least one trigger word of the inputted text, wherein

the extracting, through the encoder and the decoder, the first text feature vector set corresponding to the inputted text comprises:

extracting, through the encoder and the decoder, the first text feature vector set corresponding to the marked inputted text .

8. The method according to any one of claims 1 to 4, wherein the determining, based on the first text feature vector set and the argument feature vector of each event argument, the value range corresponding to each event argument comprises:

determining at least one candidate start-to-end range of each event argument based on

a quantity of corresponding lemmas in the first text feature vector set; and

determining, for each event argument, the value range corresponding to the event argument based on the at least one candidate start-to-end range of the event argument and the argument feature vector of the event argument.

9. The method according to claim 8, wherein determining, for each event argument, the value range corresponding to the event argument based on the at least one candidate start-to-end range of the event argument and the argument feature vector of the event argument comprises:

converting the argument feature vector of each event argument into a start search vector representing a start position and an end search vector representing an end position;

determining a first inner product result of a feature vector corresponding to each lemma in the first text feature vector set and each start search vector to obtain a first inner product result set of candidate start positions of each event argument;

determining a second inner product result of the feature vector corresponding to each lemma in the first text feature vector set and each end search vector to obtain a second inner product result set of candidate end positions of each event argument; and

determining the value range corresponding to each event argument based on the first inner product result set of the candidate start positions and the second inner product result set of the candidate end positions of each event argument.

10. A method for training an event extraction model, wherein the event extraction model comprises an encoder, a decoder, and an argument detection module, and the method comprises:

acquiring a training text and a target extraction template corresponding to the training text;

extracting, through the encoder and the decoder, a first text feature vector set

corresponding to the training text;

encoding, through the encoder, the target extraction template and the training text to obtain an argument feature vector of each event argument;

determining, through the argument detection module, a prediction range corresponding to each event argument based on the first text feature vector set and the argument feature vector of each event argument;

matching the prediction range of each event argument against a true value range to obtain an optimal prediction range that has a loss function value satisfying a second preset condition and corresponds to each event argument;

determining a loss function value of the event extraction model based on the optimal prediction range of each event argument; and

training the event extraction model based on the loss function value.

11. The method according to claim 10, wherein the matching the prediction range of each event argument against the true value range to obtain the optimal prediction range that has the loss function value satisfying the second preset condition and corresponds to each event argument comprises:

matching the prediction range of each event argument against the true value range by using a bipartite graph matching method to obtain a candidate matching group, wherein the candidate matching group comprises at least one matching pair, and one matching pair comprises a prediction range and a true value range of one event argument;

determining a loss function value of each candidate matching group, wherein the loss function value is used for representing a sum of distances in all matching pairs in the candidate matching group; and

selecting a candidate matching group that has a loss function value satisfying the second preset condition to obtain the optimal prediction range corresponding to each event argument.

12. The method according to claim 10 or 11, wherein the determining the loss function value of the event extraction model based on the optimal prediction range of each event argument comprises:

acquiring a first inner product result corresponding to a start position of the optimal prediction range of each event argument and a second inner product result corresponding to an end position of the optimal prediction range of each event argument;

normalizing the first inner product result and the second inner product result; and

determining, based on the normalized first inner product result and second inner product result, a loss function value of a model which the method is used for.

13. An apparatus for event extraction, comprising:

a first acquisition module, configured to acquire an inputted text of a sentence-level or document-level text, wherein the inputted text comprises at least one trigger word and at least one event argument;

a first extraction module, configured to extract, through an encoder and a decoder, a first text feature vector set corresponding to the inputted text;

a second acquisition module, configured to acquire a target extraction template corresponding to the at least one trigger word;

a first decoding module, configured to decode, through the decoder, the target extraction template and the inputted text to obtain an argument feature vector of each event argument; and

a first determination module, configured to determine, based on the first text feature vector set and the argument feature vector of each event argument, a value range corresponding to each event argument.

14. The apparatus according to claim 13, further comprising:

a first conversion module, configured to respectively convert the inputted text and the target extraction template into a text word vector set and a template word vector set, wherein

the first extraction module further comprises:

an encoding sub-module, configured to encode the text word vector set through the encoder to obtain a second text feature vector set; and

a first decoding sub-module, configured to decode the text word vector set and the second text feature vector set by using the decoder to obtain the first text feature vector set corresponding to the inputted text; and

correspondingly, the first decoding module further comprises:

a second decoding sub-module, configured to decode, by using the decoder, the template word vector set and the second text feature vector set to obtain the argument feature vector of each event argument.

15. The apparatus according to claim 14, wherein the second decoding sub-module further comprises:

a decoding unit, configured to decode, by using the decoder, the template word vector set and the second text feature vector set to obtain a template feature vector set merged with information of the inputted text; and

a first determination unit, configured to determine, from the template feature vector set, the argument feature vector of each event argument corresponding to each argument role based on position information of each argument role in the template word vector set.

16. The apparatus according to claim 15, wherein the first determination unit further comprises:

an extraction sub-unit, configured to, for the position information of each argument role in the template word vector set, extract, from the template feature vector set, a feature vector of each event argument corresponding to the argument role ; and

a pooling sub-unit, configured to perform average pooling on the feature vector of each event argument to obtain the argument feature vector of the event argument.

17. An apparatus for training an event extraction model, comprising:

a fourth acquisition module, configured to acquire a training text and a target extraction template corresponding to the training text;

a second extraction module, configured to extract, through the encoder and the decoder, a first text feature vector set corresponding to the training text;

a second encoding module, configured to encode, through the encoder, the target extraction template and the training text to obtain an argument feature vector of each event argument;

a sixth determination module, configured to determine, through the argument detection module, a prediction range corresponding to each event argument based on the first text feature vector set and the argument feature vector of each event argument;

a matching module, configured to match the prediction range of each event argument against a true value range to obtain an optimal prediction range that has a loss function value satisfying a second preset condition and corresponds to each event argument;

a seventh determination module, configured to determine a loss function value of the event extraction model based on the optimal prediction range of each event argument; and

a training module, configured to train the event extraction model based on the loss function value.

18. An electronic device, comprising a memory and a processor, the memory storing

computer programs executable on the processor, the processor executing the programs to implement the steps in the method according to any one of claims 1 to 12.

19. A computer-readable storage medium, storing computer programs, wherein a processor is configured to execute the computer programs to implement the steps in the method according to any one of claims 1 to 12.

20. A computer program product, comprising computer readable codes, wherein when the computer readable codes run in a device, a processor in the device executes instructions for implementing the steps in the method according to any one of claims 1 to 12.