(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
05 April 2018 (05.04.2018)

W!P O | PCT

(10) International Publication Number
WO 2018/060450 Al

(51) International Patent Classification:
G06F 17/30 (2006.01)

(21) International Application Number:
PCT/EP20 17/0748 18

(22) International Filing Date:
29 September 2017 (29.09.2017)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/401,293 29 September 2016 (29.09.2016) US

(71) Applicant: **KONINKLIJKE PHILIPS N.V.** [NL/NL]; High Tech Campus 5, 5656 AE Eindhoven (NL).

(72) Inventors: **GHAEINI, Reza;** High Tech Campus 5, 5656 AE Eindhoven (NL). **AL HASAN, Sheikh, Sadid;** High Tech Campus 5, 5656 AE Eindhoven (NL). **FARRI, Oladimeji, Feyisetan;** High Tech Campus 5, 5656 AE Eindhoven (NL). **LEE, Kathy, Mi, Young;** High Tech Campus 5, 5656 AE Eindhoven (NL). **DATLA, Vivek, Varma;** High Tech Campus 5, 5656 AE Eindhoven (NL). **QADIR, Ashequl;** High Tech Campus 5, 5656 AE Eindhoven (NL). **LIU, Junyi;** High Tech Campus 5, 5656 AE Eindhoven (NL). **PRAKASH, Adi;** High Tech Campus 5, 5656 AE Eindhoven (NL).

(74) Agent: **VAN VELZEN, Maaike, Mathilde** et al; Philips International B.V., Intellectual Property & Standards, High Tech Campus 5, 5656 AE Eindhoven (NL).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available):* AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available):* ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ,

(54) Title: QUESTION GENERATION

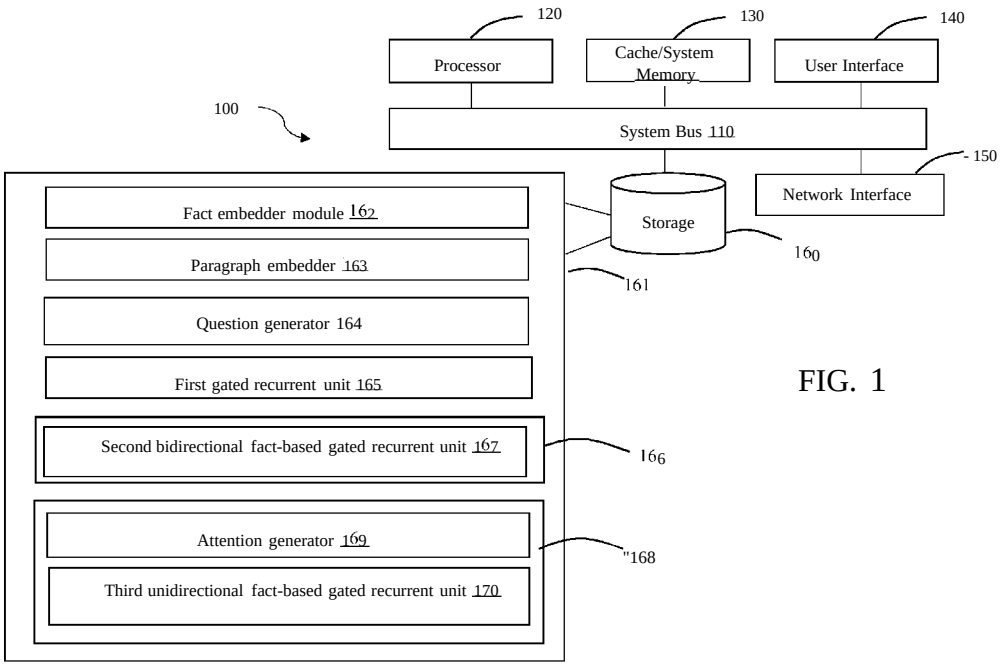

FIG. 1

(57) Abstract: Methods and systems for generating a question from free text. The system is trained on a corpus of data and receives a tuple consisting of a paragraph (free text), a focused fact, and a question type. The system implements a language model to find the most optimal combination of words to return a question for the paragraph about the focused fact.

WO 2018/060450 A1

[Continued on next page]

UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

Published:

— *with international search report (Art. 21(3))*

QUESTION GENERATION

TECHNICAL FIELD

[0001] Embodiments described herein generally relate to systems and methods for generating questions and, more particularly but not exclusively, to systems and methods for generating questions from free text using deep learning networks.

BACKGROUND

[0002] Search engines cannot always satisfy an end user's desire to have more direct access to relevant documents or information. Question answering (QA) systems have attempted to improve this experience by directly retrieving relevant answers to natural language questions.

[0003] However, one of the main requirements of a QA system is that it must receive a concise and well-described question as an input to generate the best possible answer as an output. Prior studies have revealed that humans do not always ask succinct questions on a specific topic of interest. For example, variations in expressive abilities among users may impact the ability of QA systems to "understand" the input queries and therefore the QA system may not truly understand the information that a user is seeking. Accordingly, the performance of the QA system may be adversely affected.

[0004] Using existing search engines is therefore time consuming and often unsatisfying as they are unable to handle complex queries well. Accordingly, accurate answers may not be retumed. While existing QA systems have addressed some of these limitations, they generally do not perform well with questions that are related to many topics of interest and that may be unrelated.

[0005] A need exists, therefore, for question generation systems and methods that overcome the disadvantages of existing techniques.

SUMMARY

[0006] This summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description section. This summary is not

intended to identify or exclude key features or essential features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter.

[0007] In one aspect, embodiments relate to a method of generating a question from text. The method includes receiving textual content using an interface, receiving a factual statement associated with the textual content using the interface, and generating, using a processor executing instructions stored on a memory to provide a question generator module, a question from the textual content relating to the factual statement.

[0008] In some embodiments, the method further includes receiving a question type related to the textual content using the interface.

[0009] In some embodiments, the factual statement consists of a plurality of words, and the method further comprises mapping, using a processor executing instructions stored on a memory to provide a fact embedder module, a sequence of the plurality of words to word embeddings. In some embodiments, the method further includes processing the word embeddings of the fact embedder module using a first gated recurrent unit to result in a set of computed weights. In some embodiments, the method further includes providing the received textual content to a second bidirectional fact-based gated recurrent unit whose weighting is determined by the set of computed weights. In some embodiments, the method further includes providing output of the second bidirectional fact-based gated recurrent unit to at least one attention generator, each attention generator computing normalized weights for all sequences of the second gated recurrent unit. In some embodiments, the method further includes using the computed normalized weights and a third unidirectional fact-based gated recurrent unit to generate a plurality of words forming the question. In some embodiments, the at least one attention generator utilizes the set of computed weights in determining the normalized weights. In some embodiments, the second bidirectional fact-based gated recurrent unit comprises a convolutional neural network feeding forward into a plurality of recurrent neural networks.

[0010] According to yet another aspect, embodiments relate to a system for generating a question from text. The system includes an interface for receiving textual content and a factual statement associated with the textual content, and a processor executing instructions stored on a memory to provide a question generator module configured to generate a question from the textual content relating to the factual statement.

[0011] In some embodiments, the interface is further configured to receive a question type related to the textual content.

[0012] In some embodiments, the factual statement consists of a plurality of words, and the system further includes a processor executing instructions stored on a memory to provide a fact embedder module configured to map a sequence of the plurality of words to word embeddings. In some embodiments, the system further includes a first gated recurrent unit configured to process the word embeddings into a set of computed weights. In some embodiments, the system further includes a second bidirectional fact-based gated recurrent unit configured to receive the textual content whose weighting is determined by the set of computed weights. In some embodiments, the system further includes at least one attention generator configured to compute normalized weights for sequences outputted by the second bidirectional fact-based gated recurrent unit. In some embodiments, the system further includes a third unidirectional fact-based gated recurrent unit configured to generate a plurality of words forming the question using the normalized weights. In some embodiments, the at least one attention generator utilizes the set of computed weights in computing the normalized weights. In some embodiments, the second bidirectional fact-based gated recurrent unit comprises a convolutional neural network feeding forward into a plurality of recurrent neural networks.

[0013] In some embodiments, the question generator module receives input that is a concatenation of the factual statement and a paragraph. In some embodiments, the system further includes an attention generator that considers the factual statement and previous hidden states of the question generator module. In some embodiments, the question generator module includes an encoder and a decoder, and the question generator module uses a single representation of the factual statement for the encoder and the decoder.

[0014] In some embodiments, the question generator module includes an encoder and a decoder, and an input to the decoder is the element-wise product of an encoder output and an extracted fact representation from the encoder.

[0015] According to yet another aspect, embodiments relate to a computer readable medium containing computer-executable instructions for performing a method of generating a question from text. The medium includes computer-executable instructions for receiving textual content using an interface; computer-executable instructions for receiving a factual statement associated

with the textual content using the interface, and computer-executable instructions for generating, using a processor executing instructions stored on a memory to provide a question generator module, a question from the textual content relating to the factual statement.

BRIEF DESCRIPTION OF DRAWINGS

[0016] Non-limiting and non-exhaustive embodiments of the invention are described with reference to the following figures, wherein like reference numerals refer to like parts throughout the various views unless otherwise specified.

[0017] FIG. 1 illustrates a system for generating a question from text in accordance with one embodiment;

[0018] FIG. 2 illustrates the workflow of operation of the system of FIG. 1 in accordance with one embodiment;

[0019] FIG. 3 illustrates a more detailed view of operation of the system of FIG. 1 in accordance with one embodiment;

[0020] FIG. 4 illustrates recurrent neural network architectures used in one embodiment of the invention;

[0021] FIG. 5 illustrates the attention mechanism of FIG. 4 in accordance with one embodiment;

[0022] FIG. 6 depicts a flowchart of a method of generating a question from text in accordance with one embodiment; and

[0023] FIG. 7 depicts a flowchart of a method of generating a question from text in accordance with another embodiment.

DETAILED DESCRIPTION

[0024] Various embodiments are described more fully below with reference to the accompanying drawings, which form a part hereof, and which show specific exemplary embodiments. However, the concepts of the present disclosure may be implemented in many different forms and should not be construed as limited to the embodiments set forth herein; rather, these embodiments are provided as part of a thorough and complete disclosure, to fully convey the scope of the concepts, techniques and implementations of the present disclosure to

those skilled in the art. Embodiments may be practiced as methods, systems or devices. Accordingly, embodiments may take the form of a hardware implementation, an entirely software implementation or an implementation combining software and hardware aspects. The following detailed description is, therefore, not to be taken in a limiting sense.

[0025] Reference in the specification to "one embodiment" or to "an embodiment" means that a particular feature, structure, or characteristic described in connection with the embodiments is included in at least one example implementation or technique in accordance with the present disclosure. The appearances of the phrase "in one embodiment" in various places in the specification are not necessarily all referring to the same embodiment.

[0026] Some portions of the description that follow are presented in terms of symbolic representations of operations on non-transient signals stored within a computer memory. These descriptions and representations are used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. Such operations typically require physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical, magnetic or optical signals capable of being stored, transferred, combined, compared and otherwise manipulated. It is convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like. Furthermore, it is also convenient at times, to refer to certain arrangements of steps requiring physical manipulations of physical quantities as modules or code devices, without loss of generality.

[0027] However, all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the following discussion, it is appreciated that throughout the description, discussions utilizing terms such as "processing" or "computing" or "calculating" or "determining" or "displaying" or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system memories or registers or other such information storage, transmission or display devices. Portions of the present disclosure include processes and instructions that may be embodied in software, firmware or

hardware, and when embodied in software, may be downloaded to reside on and be operated from different platforms used by a variety of operating systems.

[0028] The present disclosure also relates to an apparatus for performing the operations herein. This apparatus may be specially constructed for the required purposes, or it may comprise a general-purpose computer selectively activated or reconfigured by a computer program stored in the computer. Such a computer program may be stored in a computer readable storage medium, such as, but is not limited to, any type of disk including floppy disks, optical disks, CD-ROMs, magnetic-optical disks, read-only memories (ROMs), random access memories (RAMs), EPROMs, EEPROMs, magnetic or optical cards, application specific integrated circuits (ASICs), or any type of media suitable for storing electronic instructions, and each may be coupled to a computer system bus. Furthermore, the computers referred to in the specification may include a single processor or may be architectures employing multiple processor designs for increased computing capability.

[0029] The processes and displays presented herein are not inherently related to any particular computer or other apparatus. Various general-purpose systems may also be used with programs in accordance with the teachings herein, or it may prove convenient to construct more specialized apparatus to perform one or more method steps. The structure for a variety of these systems is discussed in the description below. In addition, any particular programming language that is sufficient for achieving the techniques and implementations of the present disclosure may be used. A variety of programming languages may be used to implement the present disclosure as discussed herein.

[0030] In addition, the language used in the specification has been principally selected for readability and instructional purposes and may not have been selected to delineate or circumscribe the disclosed subject matter. Accordingly, the present disclosure is intended to be illustrative, and not limiting, of the scope of the concepts discussed herein.

[0031] Features of various embodiments of systems and methods described herein utilize a novel deep learning-based question generation solution to generate relevant questions from textual content (e.g., a paragraph about specific topic(s) of interest). Various embodiments use associated focused factual statements and question types to generate the relevant question(s).

[0032] Specifically, the proposed models may use a novel attention-based recurrent neural network (RNN) encoder-decoder architecture with control gates for focused facts and question types. The deep RNN in accordance with various embodiments can look back on more than just previously generated words and/or characters to decide which word and/or characters to generate next as part of the generated question. Various embodiments described herein may also use a convolutional neural network (CNN) with multiple window sizes to generate phrase embeddings based on, for example, the context span of the underlying sentence in the source paragraph.

[0033] Another novel component of various embodiments described herein is the use of a language model on top of a softmax layer. This enables the model to find an appropriate combination of words rather than just using the most probable word at each position. Accordingly, the language model can generate natural language questions.

[0034] Features of various embodiments described herein can be implemented in a variety of applications. For example, features of the systems and methods described herein can create question suggestions in search engines or conversational dialogue systems. They can also be used to generate clarification questions/decompositions from a complex question or scenario (e.g., to better understand patient complaints, etc.). They can further be used to generate question answering corpus and perform educational assessments (e.g., as part of a tutoring system).

[0035] Methods and systems of various embodiments described herein may receive, as an input, a 3-tuple comprising a source text (e.g., a paragraph of information), a focused factual statement describing the topic of the question to be generated, and an indication of the question type. In some embodiments, the method and system may then create embeddings of both the source text and the focused fact. These embeddings, along with the question type, are fed as inputs into a question generator module that includes a trained RNN that generates a sequence of word embeddings that represent the output question. A language module may also translate the generated embeddings into natural language.

[0036] FIG. 1 illustrates a system 100 for generating a question from text in accordance with one embodiment. As shown, the system 100 includes a processor 120, memory 130, a user interface 140, a network interface 150, and storage 160 interconnected via one or more system buses 110. It will be understood that FIG. 1 constitutes, in some respects, an abstraction and that

the actual organization of the system 100 and the components thereof may differ from what is illustrated.

[0037] The processor 120 may be any hardware device capable of executing instructions stored on memory 130 or storage 160 or otherwise capable of processing data. As such, the processor 120 may include a microprocessor, field programmable gate array (FPGA), application-specific integrated circuit (ASIC), or other similar devices.

[0038] The memory 130 may include various memories such as, for example LI, L2, or L3 cache or system memory. As such, the memory 130 may include static random access memory (SRAM), dynamic RAM (DRAM), flash memory, read only memory (ROM), or other similar memory devices.

[0039] The user interface 140 may include one or more devices for enabling communication with a user. For example, the user interface 140 may include a display, a mouse, and a keyboard for receiving user commands. In some embodiments, the user interface 140 may include a command line interface or graphical user interface that may be presented to a remote terminal via the network interface 150.

[0040] The user interface 140 may present an agent in the form of an avatar to communicate with a user. If the user is a child, for example, the avatar may be presented as a cartoon character to make the user feel more comfortable. The displayed agent may of course vary and depend on the application.

[0041] The network interface 150 may include one or more devices for enabling communication with other hardware devices. For example, the network interface 150 may include a network interface card (NIC) configured to communicate according to the Ethernet protocol. Additionally, the network interface 150 may implement a TCP/IP stack for communication according to the TCP/IP protocols. Various alternative or additional hardware or configurations for the network interface 150 will be apparent.

[0042] The storage 160 may include one or more machine-readable storage media such as read-only memory (ROM), random-access memory (RAM), magnetic disk storage media, optical storage media, flash-memory devices, or similar storage media. In various embodiments, the

storage 160 may store instructions for execution by the processor 120 or data upon with the processor 120 may operate.

[0043] For example, the storage 160 may include a question generator module 161. The question generator module 161 may include a fact embedder module 162, a paragraph embedder 163, a question generator 164, and a first gated recurrent unit 165. The question generator module 161 may further include an encoder 166 with a second bidirectional fact-based gated recurrent unit 167 and a decoder 168 with attention generator(s) 169, and a third gated recurrent unit 170. This illustration of the question generator module 161 is merely exemplary and it is contemplated that the system 100, as well as the question generator module 161, may include components in addition to or in lieu of those shown in FIG. 1.

[0044] The system 100 and, namely, the question generator module 161 may be trained on any appropriate corpus of language data. For example, one embodiment of the system 100 may use the Stanford Question Answering Dataset (SQuAD). SQuAD was originally used as a reading comprehension dataset consisting of over 100,000 question-answer pairs. The questions were derived from over 23,000 short paragraphs curated from a collection of over 5,000 Wikipedia articles, and the answer to each question is a segment of text from the corresponding paragraph.

[0045] The paragraphs may be used as the source, the answers may be used as the focused factual statements, and the corresponding question may be used as the target to train the question generator module 161. The SQuAD dataset also contains different question types (e.g., "what" questions, "when" questions, "how" questions, etc.) about domains such as math, sports, biographies, events, etc.

[0046] To train the question generator module 161, tuples of (paragraphs, questions, and answers) from SQuAD may be used. From the entire SQuAD dataset, randomly selected portions of the data may be selected to be used as the training set, development set, and testing set, respectively. It follows that the same paragraph may have different corresponding facts and target questions in the training and testing sets.

[0047] FIG. 2 illustrates the workflow 200 of the question generator module 161 of FIG. 1 in accordance with one embodiment. As mentioned previously, a dataset 202 or other input

mechanism may provide a tuple 204 comprising a source paragraph, a focused fact, and a question type.

[0048] The focused fact may be communicated to the fact embedder module 162 for embedding since the focused fact may be represented by more than word. The question type may indicate the type of question to be generated (e.g., a "what" question, a "where" question, a "when" question, etc.).

[0049] The paragraph (or other free text) may be communicated to the paragraph embedder 163 for embedding. The paragraph embedder 163 may generate a sequence of embeddings for words from the paragraph. The output of the paragraph embedder 163 may be a concatenation of each window's output.

[0050] The question generator 164 may include or otherwise be configured with the encoder 166 and decoder 168 that implement a bi-directional recurrent neural network (RNN). The output of the question generator 164 is the generated question 206 about the focused fact.

[0051] FIG. 3 depicts a more detailed workflow 300 of the question generator module 161 in accordance with one embodiment. As stated previously, the input into the question generator module 161 is a tuple including a paragraph 302 (i.e., free text), a focused-fact 304, and a question type 306.

[0052] The paragraph 302 may be fed into a convolutional neural network (CNN) 308. The CNN 308 may use windows of various sizes W_s to capture sequences of words from the paragraph 302. In other words, a CNN window W_s may capture all words of the source text by capturing smaller sequences of words individually. The embeddings may therefore be generated based on the context span of the underlying sentence. Accordingly, the CNN 308 provides a rich embedding solution.

[0053] The output of the CNN may be the concatenation of each CNN's window's output. This output may then be fed into a plurality of recurrent neural networks 310 implemented by the question generator 164.

[0054] The focused fact 304 may also be embedded in various embodiments (e.g., if the focused factual statement consists of more than one word). The embedded focused fact may be communicated to a squeezer module 312 that generates a vector representation from the

sequence of word embeddings. The squeezer module 312 may implement any one of various tools known in the art such as those listed in FIG. 3.

[0055] The embedded focused fact 304 and the question type 306 may be fed into a concatenator module 314. The concatenator module 314 may be configured to concenate the focused-fact 304 and the question type 306 and feed them into one or more layers of a bi-directional recurrent neural network (RNN) 310 implemented by the question generator 164.

[0056] FIG. 4 illustrates an RNN 310 implemented by the question generator 164 in accordance with one embodiment. In the context of FIG. 4, $In\text{-}W_t$, $F\text{-}W_J$, and $Q\text{-}WK$ stand for the $/^{h}$, J^{h}, and K^{th} word of the input paragraph (received from the paragraph embedder 163), the focused fact, and the generated question, respectively. Also, n, m, and q are the length of the paragraph, the focused fact, and the generated question, respectively.

[0057] It is noted that the RNN 310 shown in FIG. 4 is only one embodiment, which is referred to as a "deep curious" RNN model. There may be several other types of RNN models that consider different types of input. These different models are discussed below.

[0058] The gated recurrent units 165a and 165b receive the same input $(\{F\text{-}Wi, F\text{-}W_2,...,F\text{-}W_m\})$. The output of the second bidirectional fact-based GRU 167 may be the concatenation of forward and backward outputs of the GRU 165a and the output of the paragraph embedder 163.

[0059] That is, the bidirectional fact-based gated recurrent unit 167 receives the words of the input paragraph $In\text{-}Wi$, $In\text{-}W2$, $In\text{-}Wk$, $In\text{-}W_{n}.i\ldots,In\text{-}W_{n}$, and outputs the concatenation of the forward and backward representation of the input paragraph. The forward representation of an input word $In\text{-}W_k$ may refer to analyzing the word(s) in the supplied text by reading the words in order from left to right. The backward representation reverses the order and reads the word(s) backwards in order (right to left). Analyzing both representations recognizes the dependency between the words and allows for a better understanding of how words relate to one another. The second bidirectional fact-based GRU 167 then outputs sequences 402 for each input word $In\text{-}W_k$.

[0060] To compute the hidden state h_t, the encoder 166 performs the following calculations:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + F_r d_{enc}) \quad \text{(Eq. 1)}$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + F_z d_{enc}) \quad \text{(Eq. 2)}$$

$$\hat{h}_t = \tanh(Wx_t + U(r_t \odot h_{t-1})) \quad \text{(Eq. 3)}$$

$$h_t = (1 - z_t)h_{t-1} + z_t\hat{h}_t \quad \text{(Eq. 4)}$$

[0061] In the encoder-decoder model shown in FIG. 4, the encoder 166 reads the input sequence of word embeddings $X = (x_1, \ldots, x_T)$ obtained from the paragraph embedder 163 into to the sequence of embeddings $H = (h_1, h_2, \ldots, h_T)$, while the received focused fact received from the GRU 165a is used to compute the gating for all sequences.

[0062] Equations 1-4 may be referred to as the operational stages of the GRU 165a. W, U, F are weighting parameters, x_t is the input at time t, h_{t-1} is the state and the output of the second fact-based gated recurrent unit 167 at time t-1. d_{enc} is the time-independent embedding of the fact that is extracted from the GRU 165a and is the same for all time sequences. $\odot$ is the element-wise product operation, r_t and z_t are the reset gate and the update gate at time t, respectively. $\hat{h}$ is the new candidate state at time t, and h_t is the final state and output of the GRU 165a at time t.

[0063] Equation 1 calculates the reset gate and determines the importance of h_{t-1} (the state at t-1) in calculating the summarization $\hat{h}_t$. Equation 2 calculates the update signal and determines how much of h_{t-1} should be considered in the calculating the next state h_t at time t. For example, if z_t is approximately equal to 1, then h_{t-1} is almost entirely copied to h_t. On the other hand, if z_t is approximately equal to 0, then mostly the new memory $\hat{h}_t$ is forwarded to calculate the next hidden state.

[0064] Equation 3 calculates the new memory $\hat{h}_t$ which is the consolidation of a new input x_t with the past hidden state h_{t-1}. This equation essentially combines a newly observed word with a previous state h_{t-1} to summarize the new word in the context of the previous state. Finally, equation 4 calculates the final state h_t which is the output of the gated recurrent unit 165a.

[0065] The functions executed by the decoder 168 are similar to equations 1-4 above and are:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + F_r d_{dec}) \quad \text{(Eq. 5)}$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + F_z d_{dec}) \quad \text{(Eq. 6)}$$

$$\hat{h}_t = \tanh(Wx_t + U(r_t \odot h_{t-1})) \quad \text{(Eq. 7)}$$

$$h_t = (1 - z_t)h_{t-1} + z_t\hat{h}_t \quad \text{(Eq. 8)}$$

[0066] The decoder 168 utilizes the focused fact (received from GRU 165b) in a similar manner as the encoder 166. The equations 5-8 are similar to equations 1—4 except ***ddec*** stands for the independent embedding of the focused fact that is extracted from the GRU 165b.

[0067] The decoder 168 is trained to predict the next word y_t (referred to as $Q\text{-}W_t$ in FIG. 4) based on the context vector c from the encoder 166 and the previous predicted words $(y_t,...,y_{t-i})$. The decoder 168 may define a probability over the prediction sequence $Y = (yi,...,yr)$ by decomposing the joint probability that a particular sequence of words appears in the dataset into the order conditionals:

$$P(Y) = \prod_{t=1}^{T} P(y_t \mid \{y_1, - , y_{t-i}\}, c_t, d_{dec}) \quad \text{(Eq. 9)}$$

[0068] In RNNs, each conditional probability is modeled as follows:

$$p(y_t \mid \{yi, - , y_t\text{-}i\}, c_t, d_{dec}) = f(y_t\text{-}i, s_t, c_t, d_{dec}) \quad \text{(Eq. 10)}$$

[0069] where:

$$s_t = g(y_{t-i}, S_{t-i}, c_{t-ddec}), \quad \text{(Eq. 11)}$$

$$c_t = q(\{hi, ..., h_T\}, s_{t-1}) \quad \text{(Eq. 12)}$$

and f, g, and q, are nonlinear and potentially multilayered functions, s_t is the hidden state of the decoder 168 at time t, and c_t is the context vector from the encoder 166 at time t which is generated based on the function q. The context vector c_t is computed as a weighted sum of the outputs $H = (hi,..., h_T)$ from the encoder 166 using the equation:

$$c_t = \sum_{t=1}^{T} a_{ti} h_i \quad \text{(Eq. 13)}$$

[0070] where:

$$e_{ti} = \Phi(s_{t-1}, h_i), \quad \text{(Eq- 14)}$$

$$a_{ti} = \frac{\exp(e_{ti})}{\sum_{j=1}^{T} \exp(e_y)} \quad \text{(Eq- 15)}$$

and ϕ is a feedforward neural network that is jointly trained with other components of the model.

[0071] FIG. 5 illustrates the outputted concatenated sequences 402 of FIG. 4 being communicated to the attention generator 169 of FIG. 4. The attention generator(s) 169 may be

configured to compute normalized weights for each sequence (i.e., hidden representation) outputted by the encoder 166.

[0072] Specifically, in order to generate the i^{th} word of the question $(Q\text{-}W_t)$ the attention generator 169 may compute the weight and importance of each encoder output according to the previous decoder hidden state using equation 14. Referring back to FIG. 3, this step may be performed by the sofltmax layer 314. The sofltmax function mitigates the effect of extreme values or outliers in a dataset without entirely removing them from the dataset.

[0073] Then, the attention generator 169 may normalize the computed weights for all sequences and communicate the normalized weights to the third fact-based GRU 170. The third fact-based GRU 170 may use the weighted sum of the encoder's hidden representations to generate a possible word for the question.

[0074] Referring back to FIG. 3, the output of the decoder 168, as well as the sofltmax layer 314, may be communicated to a language model 316. The language model 316 may be configured to at least assess proposed questions.

[0075] For example, for each word position in a sequence, the top k probable words will be communicated to the language model 316. The language model 316 may execute a beam search and/or an n-gram based language model (as well as any other suitable type of language model). The score of each word would be the multiplication of the sofltmax value (as determined by equation 14) and the n_gram probabilities. Finally, the generated question may be supplied to a user via a user interface 140.

[0076] As mentioned previously, there may be several configurations of the recurrent neural networks. One implementation may be referred to as an "encoder-decoder" model. In this configuration, the encoder-decoder model may be built on a simple attention-based RNN encoder-decoder framework. The model does not use a factual statement and the RNN generates questions directly from the source text or paragraph.

[0077] Another implementation may be referred to as simply a "deep curious" model. The deep curious model may be a fact-based RNN that is built on an attention-based encoder-decoder framework. In this embodiment, the factual statement affects the gating of the encoder and the decoder RNN.

[0078] Another implementation may be referred to as an "augmented deep curious" model which is an augmented version of the deep curious model discussed above. In this type of model, the focused fact (i.e., the fact representation that is extracted for the decoder part, *ddec)* also affects the attention generator of the network. In other words, the attention generator considers the factual statement in addition to the previous hidden state of the decoder and the hidden states of the encoder in accordance with equation 16 below.

$$e_{ti} = \phi\,(s_{t-1}, h_i) \qquad \text{(Eq. 16)}$$

[0079] Another implementation may be referred to as a "simplified deep curious" model. This model is similar to the deep curious model discussed above, but uses one single fact representation for both the encoder and decoder.

[0080] Another implementation may be referred to as an "elementary deep curious" model. The elementary deep curious model is a fact-based RNN that is built on an attention-based encoder-decoder framework. However, the final output from the encoder that passes through the decoder is the element-wise product of the encoder output and the extracted fact representation from the encoder GRU *(d enc)-*

[0081] Another implementation may be referred to as a "separate deep curious" model. This model is similar to the elementary deep curious model. However, the separate deep curious model uses a separate GRU to extract a fact representation called *d eiem-* Then, the model uses *deiem* instead of *d enc* in the element-wise product.

[0082] Another implementation may be referred to as a "fact+ encoder-decoder" model. This model is similar to the encoder-decoder model discussed above, but its input is the concatenation of the factual statement and input paragraph.

[0083] Another implementation may be referred to as a "fact+ deep curious" model. This model is similar to the deep curious model, but its input is the concatenation of the factual statement and input paragraph.

[0084] Another implementation may be referred to as an "augmented+ deep curious" model. This model is similar to the augmented deep curious model, but its input is the concatenation of the factual statement and paragraph.

[0085] FIG. 6 depicts a flowchart of a method 600 of generating a question from text in accordance with one embodiment. Step 602 involves receiving textual content using an interface. The textual content may be in the form of a paragraph of free text, for example, and may be received through any suitable interface such as those discussed previously.

[0086] Step 604 involves receiving a factual statement associated with the textual content using the interface. The factual statement or "focused fact" indicates to what the generated question should be directed. Accordingly, the question generator module 161 may suggest more relevant questions based on the desires of the user(s).

[0087] Step 606 involves generating, using a processor executing instructions stored on a memory to provide a question generator module, a question from the textual content relating to the factual statement. The processor may rely on convolutional neural networks that feed into one or more recurrent neural networks to find the most optimal combination of words to generate the most relevant question. Then, a generated question may be outputted to a user.

[0088] FIG. 7 depicts a flowchart of a method 700 of generating a question from text in accordance with another embodiment. Steps 702 and 704 are similar to steps 602 and 604 of FIG. 6, respectively, and are not repeated here. Step 706 involves receiving a question type related to the textual content. The question type may refer to whether the generated question should be a "what" question, a "why" question, a "when" question, etc.

[0089] Step 708 involves mapping a sequence of the factual statement to word embeddings. In some embodiments the factual statement consists of a plurality of words. In these embodiments, the processor may execute instructions stored on a memory to provide a fact embedder module. The fact embedder module may be similarly configured to the fact embedder 162 of FIG. 2, for example, and may map the sequence of words of the factual statement to word embeddings.

[0090] Step 710 involves processing the word embeddings of the fact embedder module using a first gated recurrent unit to result in a set of computed weights. The first gated recurrent unit may be similar to the GRU 165a of FIG. 4, for example.

[0091] Step 712 involves providing the received textual content to a second bidirectional fact-based gated recurrent unit whose weighting is determined by the set of computed weights.

The second bidirectional fact-based gated recurrent unit may be similar to the bidirectional fact-based gated recurrent unit 167 of FIG. 4, for example.

[0092] Step 714 involves providing output of the second bidirectional fact-based gated recurrent unit to at least one attention generator, each attention generator computing normalized weights for all sequences of the second gated recurrent unit. The attention generators may be similar to the attention generators 169 of FIG. 4.

[0093] Step 716 involves using the computed normalized weights and a third unidirectional fact-based gated recurrent unit to generate a plurality of words forming the question. The third unidirectional fact-based gated recurrent unit may be similar to the third fact-based GRU 170 of FIG. 4.

[0094] Finally, the method 700 ends with step 718, which involves generating a question relating to the factual statement using the words generated in step 716. The question may then be outputted to a user.

[0095] The methods, systems, and devices discussed above are examples. Various configurations may omit, substitute, or add various procedures or components as appropriate. For instance, in alternative configurations, the methods may be performed in an order different from that described, and that various steps may be added, omitted, or combined. Also, features described with respect to certain configurations may be combined in various other configurations. Different aspects and elements of the configurations may be combined in a similar manner. Also, technology evolves and, thus, many of the elements are examples and do not limit the scope of the disclosure or claims.

[0096] Embodiments of the present disclosure, for example, are described above with reference to block diagrams and/or operational illustrations of methods, systems, and computer program products according to embodiments of the present disclosure. The functions/acts noted in the blocks may occur out of the order as shown in any flowchart. For example, two blocks shown in succession may in fact be executed substantially concurrent or the blocks may sometimes be executed in the reverse order, depending upon the functionality/acts involved. Additionally, or alternatively, not all of the blocks shown in any flowchart need to be performed and/or executed. For example, if a given flowchart has five blocks containing functions/acts, it

may be the case that only three of the five blocks are performed and/or executed. In this example, any of the three of the five blocks may be performed and/or executed.

[0097] A statement that a value exceeds (or is more than) a first threshold value is equivalent to a statement that the value meets or exceeds a second threshold value that is slightly greater than the first threshold value, e.g., the second threshold value being one value higher than the first threshold value in the resolution of a relevant system. A statement that a value is less than (or is within) a first threshold value is equivalent to a statement that the value is less than or equal to a second threshold value that is slightly lower than the first threshold value, e.g., the second threshold value being one value lower than the first threshold value in the resolution of the relevant system.

[0098] Specific details are given in the description to provide a thorough understanding of example configurations (including implementations). However, configurations may be practiced without these specific details. For example, well-known circuits, processes, algorithms, structures, and techniques have been shown without unnecessary detail in order to avoid obscuring the configurations. This description provides example configurations only, and does not limit the scope, applicability, or configurations of the claims. Rather, the preceding description of the configurations will provide those skilled in the art with an enabling description for implementing described techniques. Various changes may be made in the function and arrangement of elements without departing from the spirit or scope of the disclosure.

[0099] Having described several example configurations, various modifications, alternative constructions, and equivalents may be used without departing from the spirit of the disclosure. For example, the above elements may be components of a larger system, wherein other rules may take precedence over or otherwise modify the application of various implementations or techniques of the present disclosure. Also, a number of steps may be undertaken before, during, or after the above elements are considered.

[00100] Having been provided with the description and illustration of the present application, one skilled in the art may envision variations, modifications, and alternate embodiments falling within the general inventive concept discussed in this application that do not depart from the scope of the following claims.

CLAIMS

What is claimed is:

1. A method of generating a question from text, the method comprising:

receiving textual content using an interface;

receiving a factual statement associated with the textual content using the interface; and

generating, using a processor executing instructions stored on a memory to provide a question generator module, a question from the textual content relating to the factual statement.

2. The method of claim 1 further comprising receiving a question type related to the textual content using the interface.

3. The method of claim 1 wherein the factual statement consists of a plurality of words, and the method further comprises mapping, using a processor executing instructions stored on a memory to provide a fact embedder module, a sequence of the plurality of words to word embeddings.

4 The method of claim 3 further comprising processing the word embeddings of the fact embedder module using a first gated recurrent unit to result in a set of computed weights.

5. The method of claim 4 further comprising providing the received textual content to a second bidirectional fact-based gated recurrent unit whose weighting is determined by the set of computed weights.

6. The method of claim 5 further comprising providing output of the second bidirectional fact-based gated recurrent unit to at least one attention generator, each attention generator computing normalized weights for all sequences of the second gated recurrent unit.

7. The method of claim 6 further comprising using the computed normalized weights and a third unidirectional fact-based gated recurrent unit to generate a plurality of words forming the question.

8. The method of claim 6 wherein the at least one attention generator utilizes the set of computed weights in determining the normalized weights.

9. The method of claim 5 wherein the second bidirectional fact-based gated recurrent unit comprises a convolutional neural network feeding forward into a plurality of recurrent neural networks.

10. A system for generating a question from text, the system comprising:

an interface for receiving textual content and a factual statement associated with the textual content; and

a processor executing instructions stored on a memory to provide a question generator module configured to generate a question from the textual content relating to the factual statement.

11. The system of claim 10 wherein the interface is further configured to receive a question type related to the textual content.

12. The system of claim 10 wherein the factual statement consists of a plurality of words, and the system further includes a processor executing instructions stored on a memory to provide a fact embedder module configured to map a sequence of the plurality of words to word embeddings.

13. The system of claim 12 further comprising a first gated recurrent unit configured to process the word embeddings into a set of computed weights.

14. The system of claim 13 further comprising a second bidirectional fact-based gated recurrent unit configured to receive the textual content whose weighting is determined by the set of computed weights.

15. The system of claim 14 further comprising at least one attention generator configured to compute normalized weights for sequences outputted by the second bidirectional fact-based gated recurrent unit.

16. The system of claim 15 further comprising a third unidirectional fact-based gated recurrent unit configured to generate a plurality of words forming the question using the normalized weights.

17. The system of claim 15 wherein the at least one attention generator utilizes the set of computed weights in computing the normalized weights.

18. The system of claim 14 wherein the second bidirectional fact-based gated recurrent unit comprises a convolutional neural network feeding forward into a plurality of recurrent neural networks.

19. The system of claim 10 wherein the question generator module receives input that is a concatenation of the factual statement and a paragraph.

20. The system of claim 19 further comprising an attention generator that considers the factual statement and previous hidden states of the question generator module.

21. The system of claim 19 wherein the question generator module includes an encoder and a decoder, and the question generator module uses a single representation of the factual statement for the encoder and the decoder.

22. The system of claim 10 wherein the question generator module includes an encoder and a decoder, and an input to the decoder is the element-wise product of an encoder output and an extracted fact representation from the encoder.

23. A computer readable medium containing computer-executable instructions for performing a method of generating a question from text, the medium comprising:

computer-executable instructions for receiving textual content using an interface;

computer-executable instructions for receiving a factual statement associated with the textual content using the interface; and

computer-executable instructions for generating, using a processor executing instructions stored on a memory to provide a question generator module, a question from the textual content relating to the factual statement.

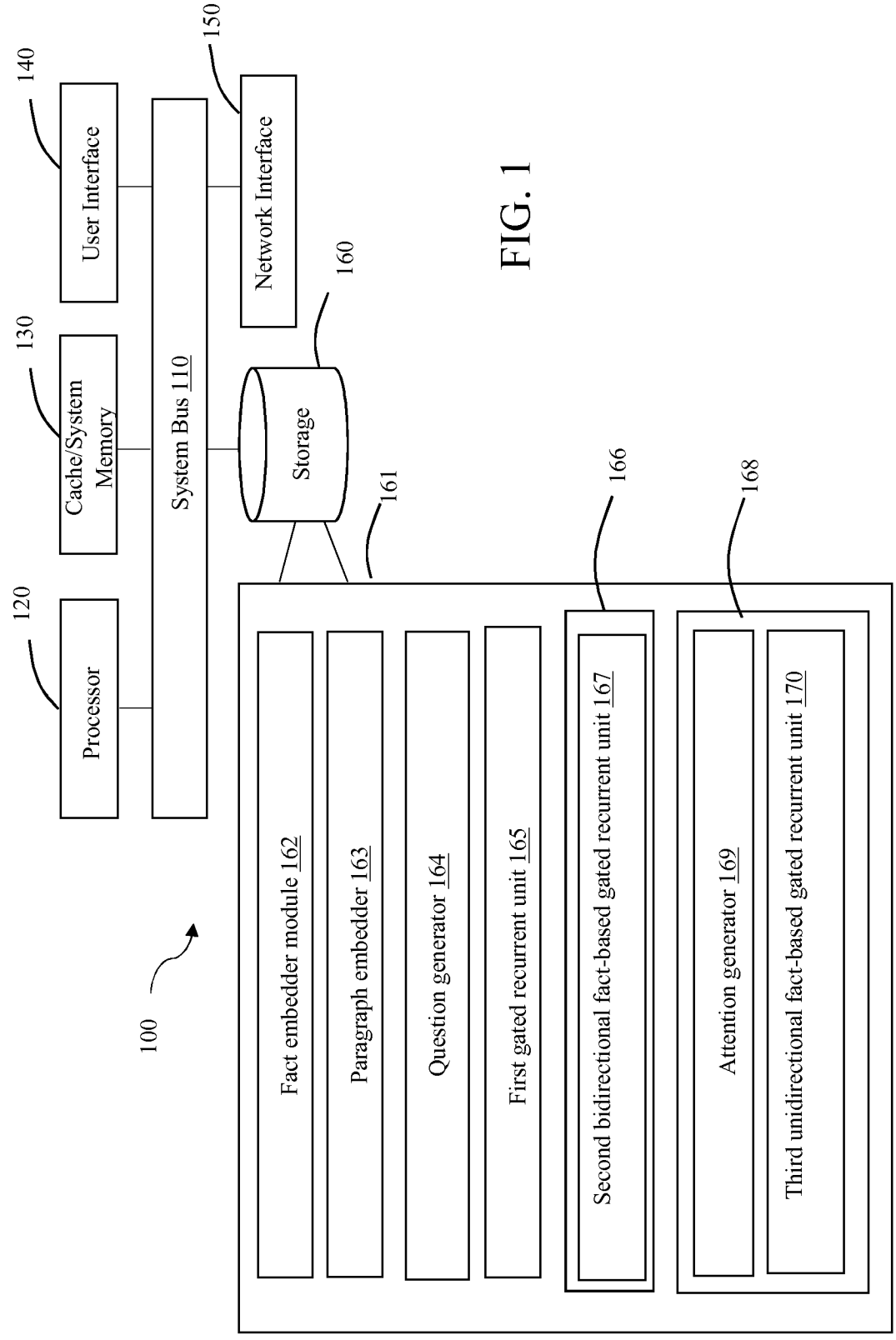
100
Processor
120
Cache/System Memory
130
User Interface
140
System Bus 110
Network Interface
150
Storage
160
161
Fact embedder module 162
Paragraph embedder 163
Question generator 164
First gated recurrent unit 165
Second bidirectional fact-based gated recurrent unit 167
166
Attention generator 169
Third unidirectional fact-based gated recurrent unit 170
168

FIG. 1

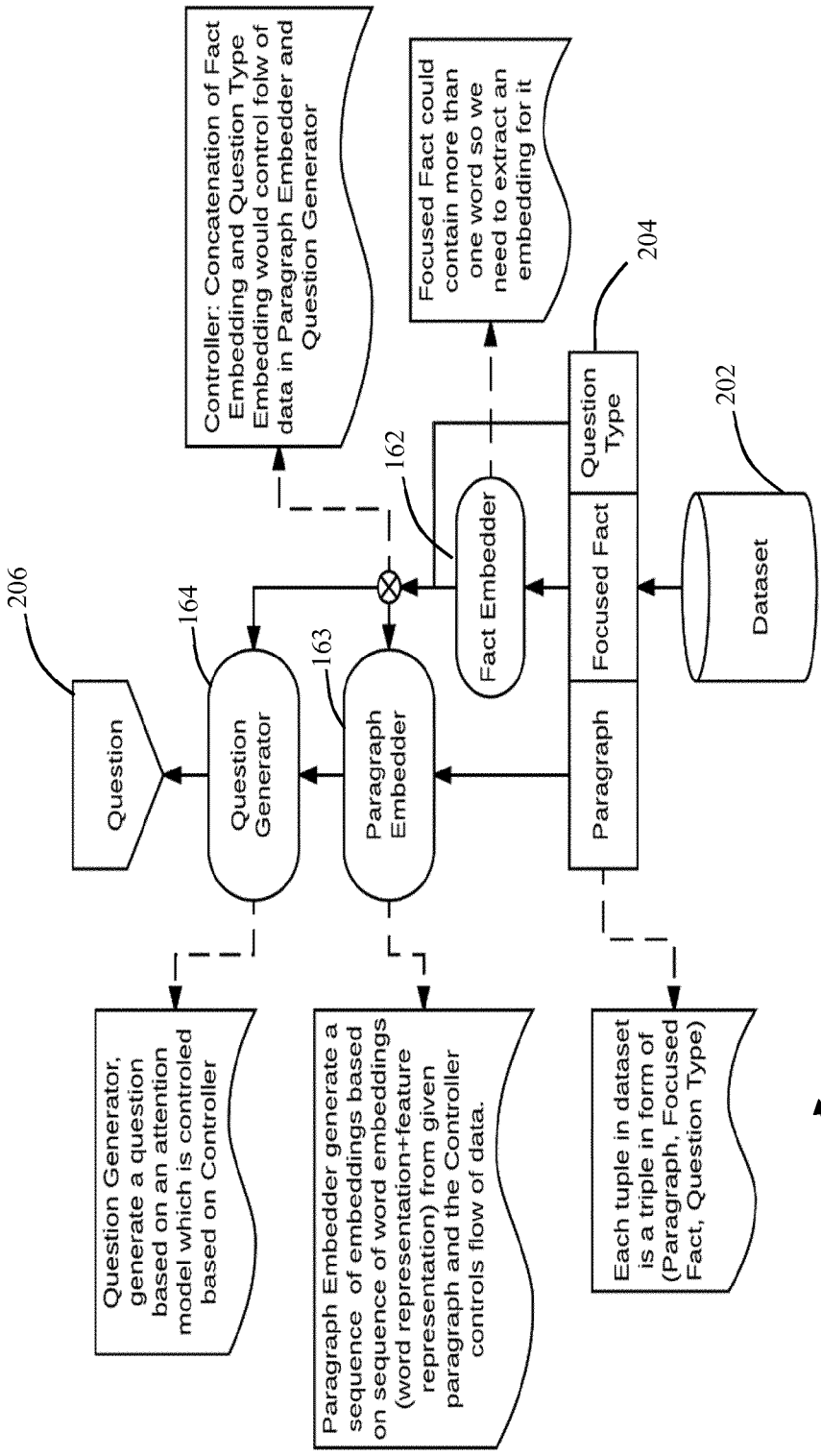
Controller: Concatenation of Fact Embedding and Question Type Embedding would control folw of data in Paragraph Embedder and Question Generator
Focused Fact could contain more than one word so we need to extract an embedding for it
162
Question Type
204
Fact Embedder
202
Dataset
206
164
163
Question
Question Generator
Paragraph Embedder
Paragraph
Focused Fact
Question Generator, generate a question based on an attention model which is controled based on Controller
Paragraph Embedder generate a sequence of embeddings based on sequence of word embeddings (word representation+feature representation) from given paragraph and the Controller controls flow of data.
Each tuple in dataset is a triple in form of (Paragraph, Focused Fact, Question Type)
200

FIG. 2

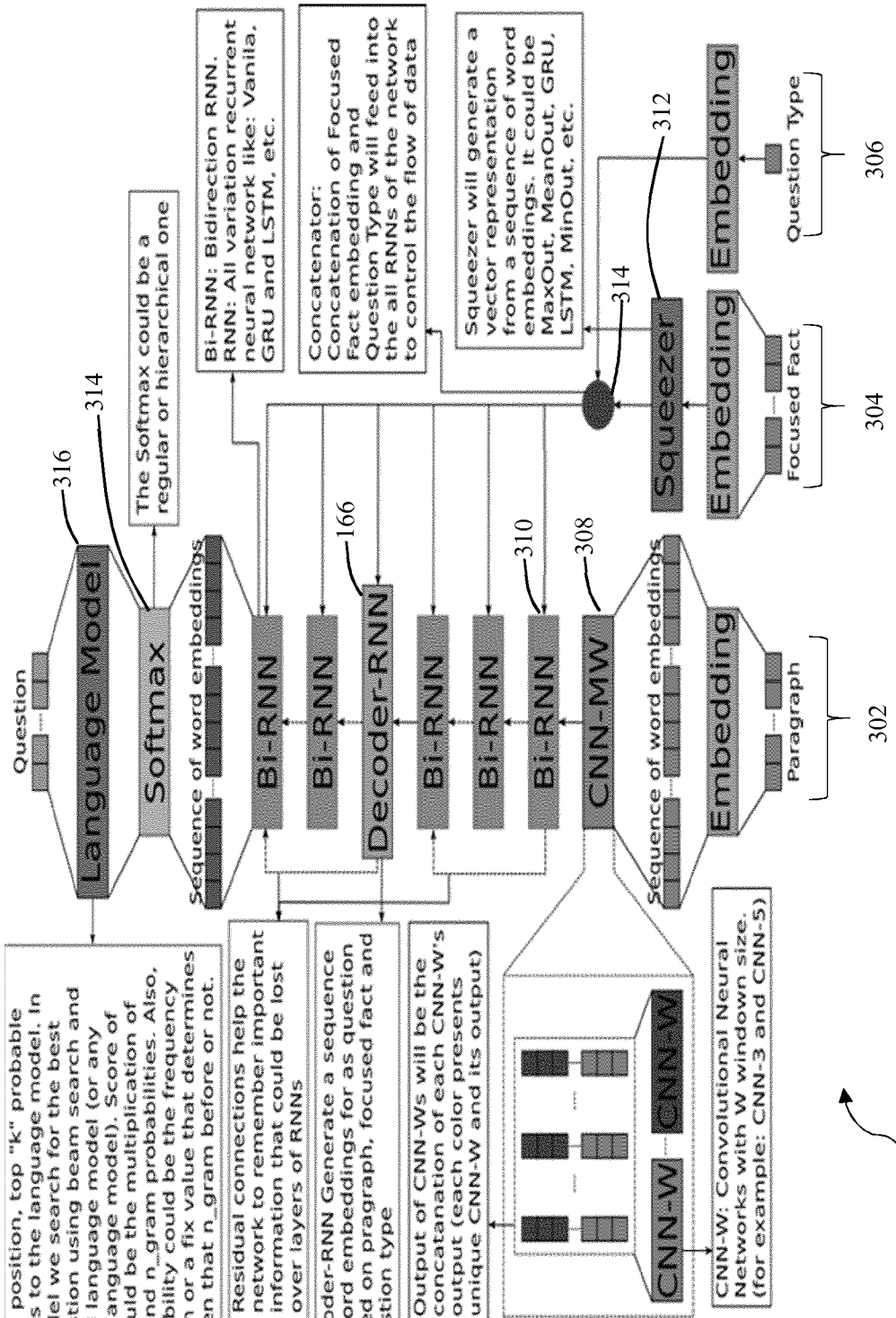
Question
Language Model
Softmax
Sequence of word embeddings
Bi-RNN
Bi-RNN
Decoder-RNN
Bi-RNN
Bi-RNN
Bi-RNN
CNN-MW
Sequence of word embeddings
Embedding
Paragraph
Squeezer
Embedding
Focused Fact
Embedding
Question Type
316
314
166
310
308
312
314
302
304
306
300
The Softmax could be a regular or hierarchical one
Bi-RNN: Bidirection RNN. RNN: All variation recurrent neural network like: Vanila, GRU and LSTM, etc.
Concatenator: Concatenation of Focused Fact embedding and Question Type will feed into the all RNNs of the network to control the flow of data
Squeezer will generate a vector representation from a sequence of word embeddings. It could be MaxOut, MeanOut, GRU, LSTM, MinOut, etc.
For each word position, top "k" probable words will pass to the language model. In language model we search for the best probable question using beam search and n_gram based language model (or any other type of language model). Score of each word would be the multiplication of the Softmax and n_gram probabilities. Also, n_gram probability could be the frequency of that n_gram or a fix value that determines wheter we seen that n_gram before or not.
Residual connections help the network to remember important information that could be lost over layers of RNNs
Decoder-RNN Generate a sequence of word embeddings for as question based on pragraph, focused fact and question type
Output of CNN-Ws will be the concatanation of each CNN-W's output (each color presents unique CNN-W and its output)
CNN-W
CNN-W
CNN-W: Convolutional Neural Networks with W windown size. (for example: CNN-3 and CNN-5)

FIG. 3

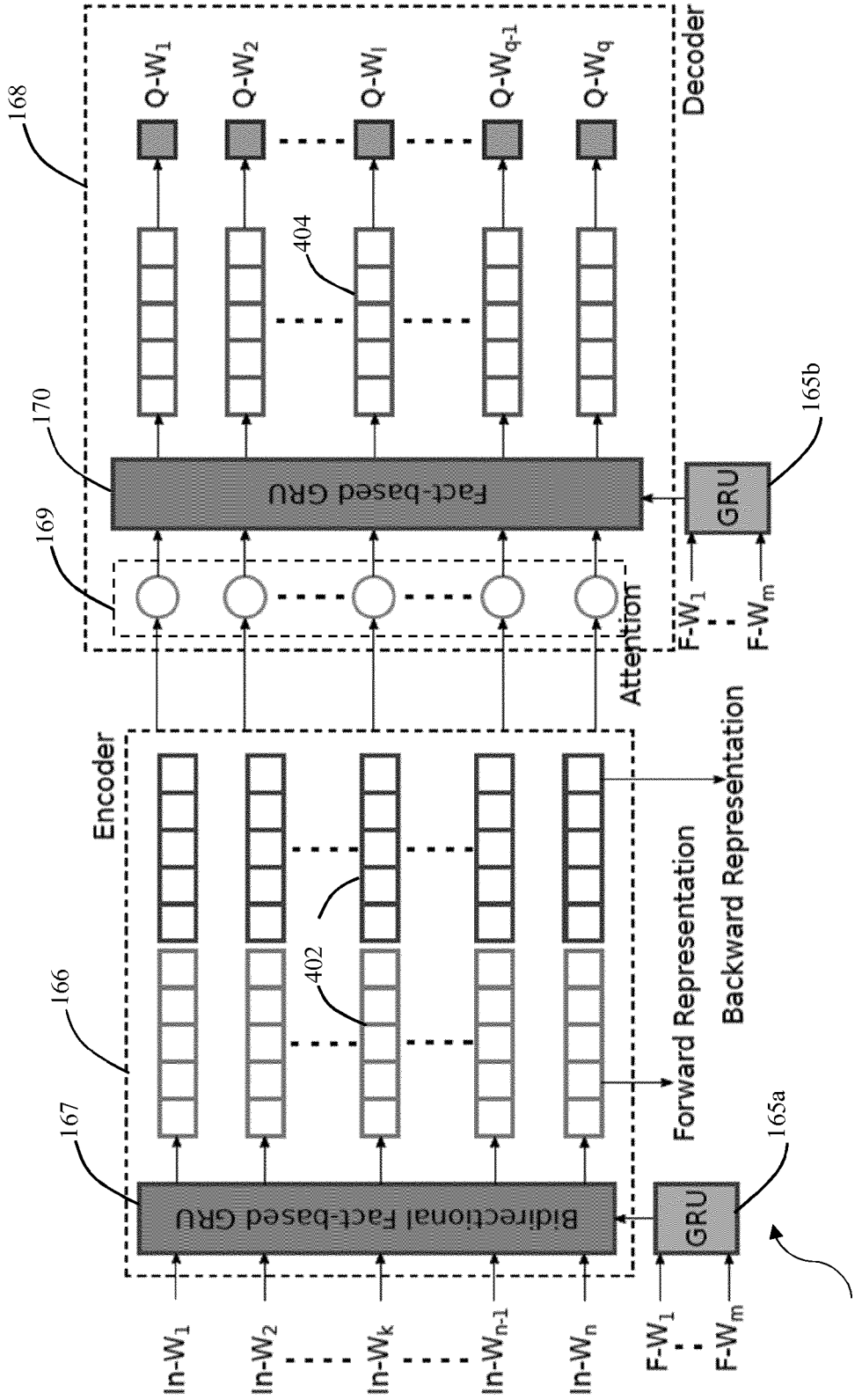
168
Decoder
Q-W1
Q-W2
Q-Wl
Q-Wq-1
Q-Wq
404
170
Fact-based GRU
GRU
165b
169
Attention
F-W1
Fm
Encoder
166
402
Forward Representation
Backward Representation
167
Bidirectional Fact-based GRU
GRU
165a
In-W1
In-W2
In-Wk
In-Wn-1
In-Wn
F-W1
Fm
310

FIG. 4

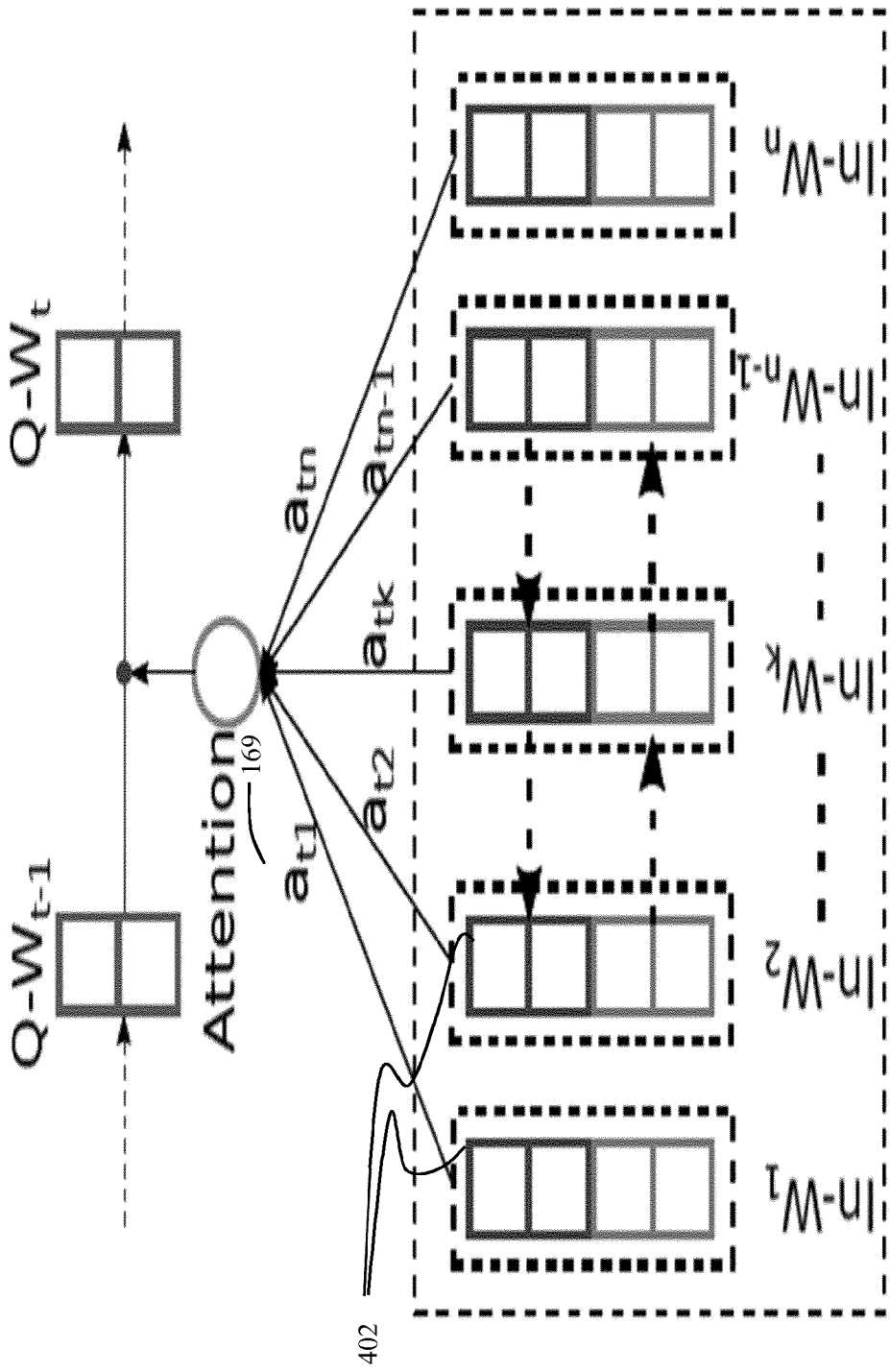
Q-W$_t$
Q-W$_{t-1}$
Attention
169
a_{tn}
a_{tn-1}
a_{tk}
a_{t2}
a_{t1}
In-W$_n$
In-W$_{n-1}$
In-W$_k$
In-W$_2$
In-W$_1$
402

FIG. 5

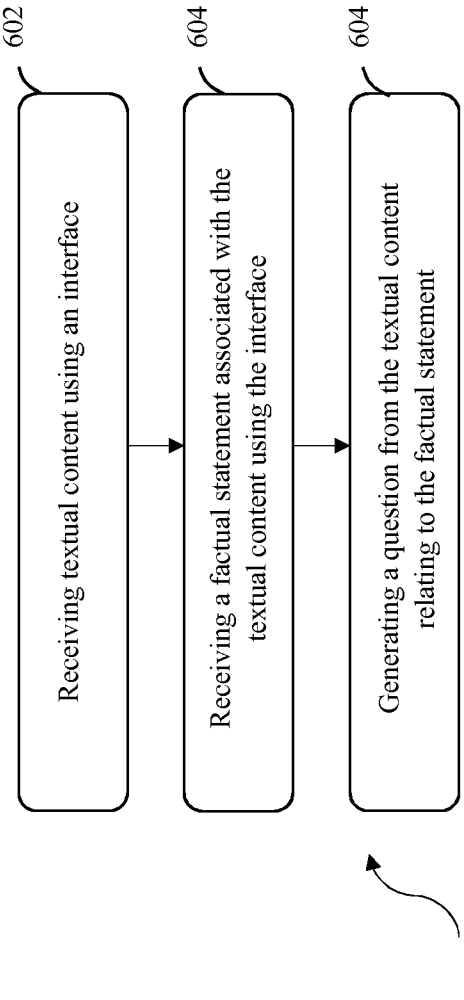
602
604
604
Receiving textual content using an interface
Receiving a factual statement associated with the textual content using the interface
Generating a question from the textual content relating to the factual statement
600

FIG. 6

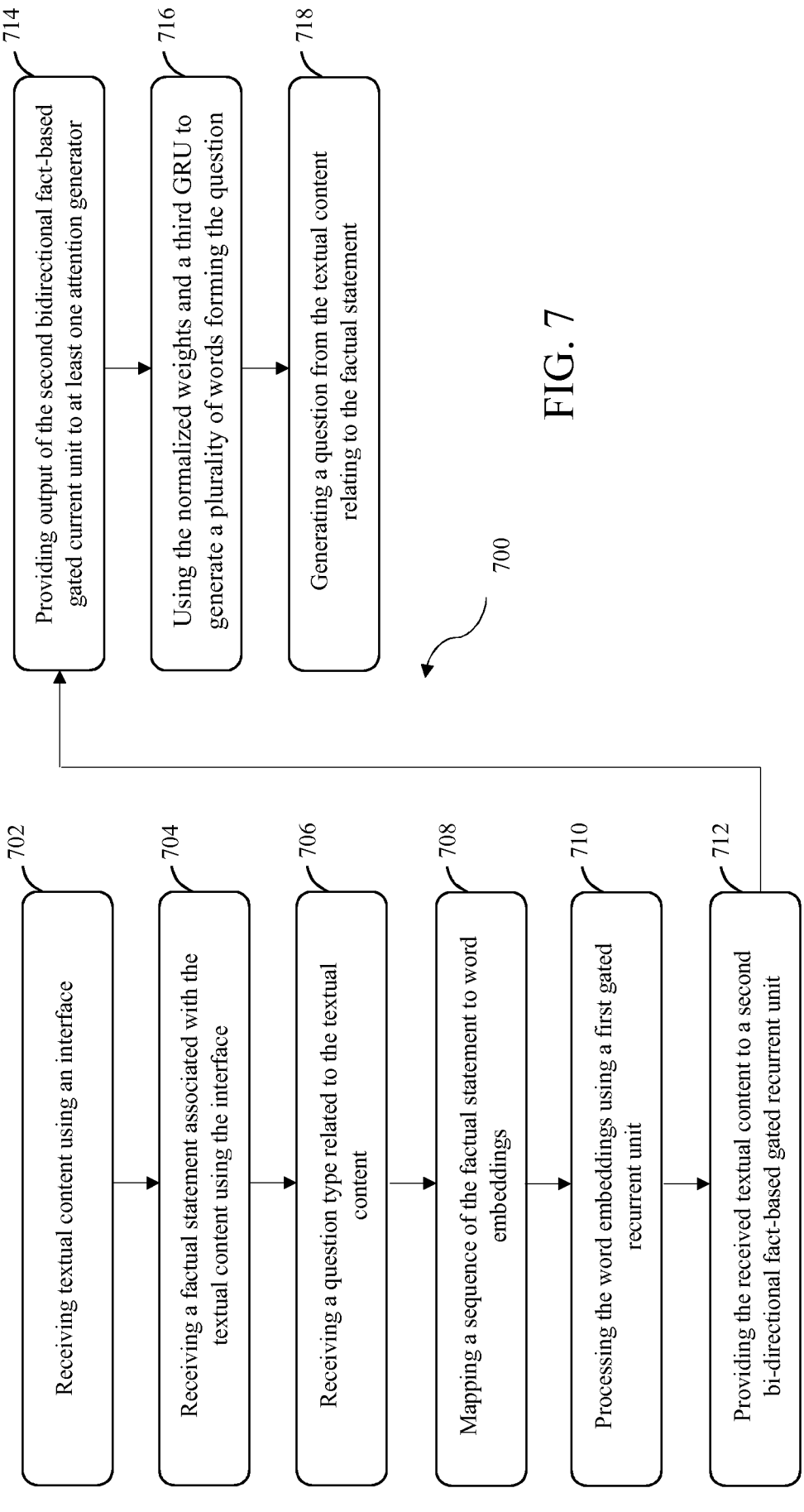
Receiving textual content using an interface
Receiving a factual statement associated with the textual content using the interface
Receiving a question type related to the textual content
Mapping a sequence of the factual statement to word embeddings
Processing the word embeddings using a first gated recurrent unit
Providing the received textual content to a second bi-directional fact-based gated recurrent unit
Providing output of the second bidirectional fact-based gated current unit to at least one attention generator
Using the normalized weights and a third GRU to generate a plurality of words forming the question
Generating a question from the textual content relating to the factual statement
700
702
704
706
708
710
712
714
716
718

FIG. 7

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2017/074818

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F17/30
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal , WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2016/032864 Al (MICROSOFT TECHNOLOGY LICENSING LLC [US]) 3 March 2016 (2016-03-03) page 1 - page 14	1-23
X	US 2016/117314 Al (KANT0R ARTHUR [CZ] ET AL) 28 Apri l 2016 (2016-04-28) page 1 - page 6	1-23

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" documentwhich may throw doubts on priority claim(s) orwhich is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
20 November 2017	28/11/2017

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Korkuzas , Val das

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2017/074818

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2016032864 A1	03-03-2016	CN 106796594 A	31-05-2017
		EP 3195148 A1	26-07-2017
		US 2016063879 A1	03-03-2016
		WO 2016032864 A1	03-03-2016
US 2016117314 A1	28-04-2016	GB 2531720 A	04-05-2016
		US 2016117314 A1	28-04-2016