



(51) International Patent Classification:

G06F 17/27 (2006.01)

G06N 3/04 (2006.01)

(21) International Application Number:

PCT/US2019/032207

(22) International Filing Date:

14 May 2019 (14.05.2019)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

62/673,606

18 May 2018 (18.05.2018)

US

16/006,691

12 June 2018 (12.06.2018)

US

(71) Applicant: **SALESFORCE.COM, INC.** [US/US]; The Landmark@ One Market St., Suite 300, San Francisco, California 94105 (US).

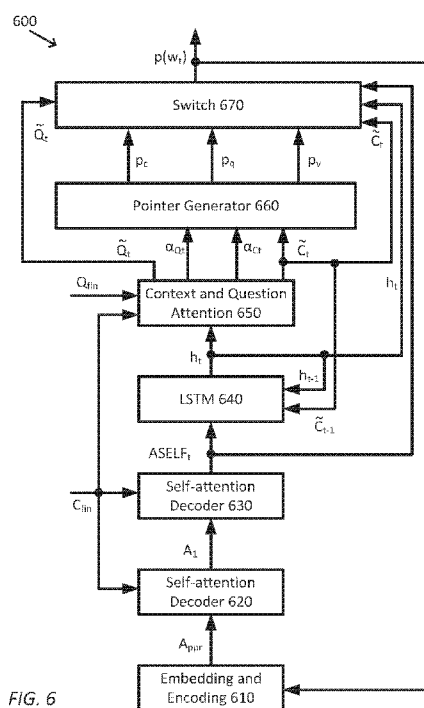
(72) Inventors: **MCCANN, Bryan**; The Landmark@ One Market St., Suite 300, San Francisco, California 94105 (US). **KESKAR, Nitish Shirish**; The Landmark@ One Market St., Suite 300, San Francisco, California 94105 (US). **XIONG, Caiming**; The Landmark@ One Market St., Suite 300, San Francisco, California 94105 (US). **SOCHER, Richard**; The Landmark@ One Market St., Suite 300, San Francisco, California 94105 (US).

(74) Agent: **WELCH, Henry** et al.; Haynes and Boone, LLP, 2323 Victory Avenue, Suite 700, Dallas, Texas 75219 (US).

(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, 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).

(54) Title: MULTITASK LEARNING AS QUESTION ANSWERING



(57) Abstract: Approaches for natural language processing include a multi-layer encoder for encoding words from a context and words from a question in parallel, a multi-layer decoder for decoding the encoded context and the encoded question, a pointer generator for generating distributions over the words from the context, the words from the question, and words in a vocabulary based on an output from the decoder, and a switch. The switch generates a weighting of the distributions over the words from the context, the words from the question, and the words in the vocabulary, generates a composite distribution based on the weighting of the distribution over the first words from the context, the distribution over the second words from the question, and the distribution over the words in the vocabulary, and selects words for inclusion in an answer using the composite distribution.

FIG. 6

Declarations under Rule 4.17:

- *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
- *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

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

MULTITASK LEARNING AS QUESTION ANSWERING

Inventors: Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher

RELATED APPLICATIONS

[0001] The present application claims priority to U.S. Provisional Patent Application No. 62/673,606, filed May 18, 2017, entitled “Multitask Learning As Question Answering,”; and U.S. Patent Application No. 16/006,691 filed June 12, 2018, entitled “Multitask Learning As Question Answering,” which are hereby incorporated by reference in their entirety.

TECHNICAL FIELD

[0002] The present disclosure relates generally to natural language processing and more specifically to answering natural language questions about a natural language context.

BACKGROUND

[0003] Natural language processing and the ability of a system to answer natural language questions about the content of a natural language sample is a benchmark to test for context-specific reasoning about information provided in natural language form. This can be a complex task because there are many different types of natural language questions that can be asked and whose answering may require different types of reasoning and/or different types of analysis.

[0004] Accordingly, it would be advantageous to have unified systems and methods for simultaneously being able to answer different kinds of natural language questions.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] Figure 1 is a simplified diagram of examples of natural language processing task types characterized as question answering according to some embodiments.

[0006] Figure 2 is a simplified diagram of a computing device according to some embodiments.

[0007] Figure 3 is a simplified diagram of an encoder according to some embodiments.

[0008] Figure 4 is a simplified diagram of an attention network according to some embodiments.

[0009] Figure 5 is a simplified diagram of a layer for an attention-based transformer network according to some embodiments.

[0010] Figure 6 is a simplified diagram of a decoder according to some embodiments.

[0011] Figure 7 is a simplified diagram of a method of multitask learning according to some embodiments.

[0012] Figure 8 is a simplified diagram of a summary of training sets according to some embodiments.

[0013] Figure 9 is a simplified diagram of results from single task and multitask learning according to some embodiments.

[0014] Figure 10 is a simplified diagram of results for different training strategies according to some embodiments.

[0015] Figure 11 is a simplified diagram of source for word selection for answers according to some embodiments.

[0016] Figures 12A and 12B are simplified diagrams of results from different training approaches according to some embodiments.

[0017] In the figures, elements having the same designations have the same or similar functions.

DETAILED DESCRIPTION

[0018] Context specific reasoning, including context specific reasoning regarding the content of natural language information, is an important problem in machine intelligence and learning applications. Context specific reasoning may provide valuable information for use in the interpretation of natural language text and can include different tasks, such as answering questions about the content of natural language text, language translation, semantic context

analysis, and/or the like. However, each of these different types of natural language processing tasks often involve different types of analysis and/or different types of expected responses.

[0019] Multitask learning in natural language processing has made progress when the task types are similar. However, when tackling different types of tasks, such as language translation, question answering and classification, parameter sharing is often limited to word vectors or subsets of parameters. The final architectures are typically highly optimized and engineered for each task type, limiting their ability to generalize across task types.

[0020] However, many of these task types can be handled by the same architecture and model when framed as a single type of task. For example, it is possible to treat many, if not all, natural language processing tasks as question answering tasks. For example, the task types of question answering, machine translation, document summarization, database query generation, sentiment analysis, natural language inference, semantic role labeling, relation extraction, goal oriented dialogue, and pronoun resolution may be framed as question answering tasks. Figure 1 is a simplified diagram of examples of natural language processing task types characterized as question answering according to some embodiments. Each of the examples in Figure 1 is shown in three column format with a question, a context, and a ground truth answer. Example 105 is an example of a question answering task where a question is asked about the content of a context. Example 110 is an example of a machine translation task from English to German where the question is posed in the form “what is the translation. . . .” Example 115 is an example of a document summarization task where the question is posed as “what is the summary?” Example 120 is an example of a natural language inference task where the question is posed using a hypothesis and a query as to whether the hypothesis entails, contradicts, or is neutral to the context. Example 125 is an example of a sentiment analysis task where the question is posed as to whether the context is positive or negative. Example 130 is an example of a semantic role labeling task where the question is posed as which entity plays an indicated role in the context. Example 135 is an example of a relation extraction task where the question is posed about the relationship of one of the entities mentioned in the context. Example 140 is an example of a goal oriented dialog task where the question is posed about knowledge that may be learned from the context. Example 145 is an example of a database query generation task where the question asks for a translation to a database query language (e.g., SQL). Example 150 is an example of a

pronoun resolution task where the question is directed to answering a question about a pronoun appearing in the context.

[0021] Figure 2 is a simplified diagram of a computing device 200 according to some embodiments. As shown in Figure 2, computing device 200 includes a processor 210 coupled to memory 220. Operation of computing device 200 is controlled by processor 210. And although computing device 200 is shown with only one processor 210, it is understood that processor 210 may be representative of one or more central processing units, multi-core processors, microprocessors, microcontrollers, digital signal processors, field programmable gate arrays (FPGAs), application specific integrated circuits (ASICs), graphics processing units (GPUs) and/or the like in computing device 200. Computing device 200 may be implemented as a stand-alone subsystem, as a board added to a computing device, and/or as a virtual machine.

[0022] Memory 220 may be used to store software executed by computing device 200 and/or one or more data structures used during operation of computing device 200. Memory 220 may include one or more types of machine readable media. Some common forms of machine readable media may include floppy disk, flexible disk, hard disk, magnetic tape, any other magnetic medium, CD-ROM, any other optical medium, punch cards, paper tape, any other physical medium with patterns of holes, RAM, PROM, EPROM, FLASH-EPROM, any other memory chip or cartridge, and/or any other medium from which a processor or computer is adapted to read.

[0023] Processor 210 and/or memory 220 may be arranged in any suitable physical arrangement. In some embodiments, processor 210 and/or memory 220 may be implemented on a same board, in a same package (e.g., system-in-package), on a same chip (e.g., system-on-chip), and/or the like. In some embodiments, processor 210 and/or memory 220 may include distributed, virtualized, and/or containerized computing resources. Consistent with such embodiments, processor 210 and/or memory 220 may be located in one or more data centers and/or cloud computing facilities.

[0024] As shown, memory 220 includes a question answering module 230 that may be used to implement and/or emulate the question answering systems and models described further herein and/or to implement any of the methods described further herein. In some examples,

question answering module 230 may be used to answer natural language questions about natural language contexts. In some examples, question answering module 230 may also handle the iterative training and/or evaluation of a question answering system or model used to answer natural language questions about natural language contexts. In some examples, memory 220 may include non-transitory, tangible, machine readable media that includes executable code that when run by one or more processors (e.g., processor 210) may cause the one or more processors to perform the counting methods described in further detail herein. In some examples, question answering module 230 may be implemented using hardware, software, and/or a combination of hardware and software. As shown, computing device 200 receives a natural language context 240 and a natural language question 250 about natural language context 240, which are provided to question answering module 230, question answering module 230 then generates a natural language answer 260 to natural language question 250 based on the content of natural language context 240.

[0025] Figure 3 is a simplified diagram of an encoder 300 according to some embodiments. Encoder 300 receives a natural language context c and a natural language question q , which each consist of a sequence of ordered words in a natural language, such as English, French, German, Spanish, and/or the like. Both the context c and the question q are encoded using respective encoding layers 310 and 315 into matrix form where the i th row of the matrix corresponds to a d_{emb} -dimensional embedding for the i th token or word in the sequence as indicated according to Equation 1, where l corresponds to the number of words or tokens in the context and m corresponds to the number of words or tokens in the question. In some examples, the encoding of each word is based on the GloVe encodings where each word is encoded as an element of $\mathbb{R}^{300}$. In some examples, the encoding of each word is based on the character n -gram encodings where each word is encoded as an element of $\mathbb{R}^{100}$. In some example, the encoding of each word is based on a concatenation of GloVe and character n -gram encodings. In some examples, when there is no GloVe and/or character n -gram encoding for a word (e.g., the word is not in English), a random encoding is selected from a Normal distribution having a mean and standard deviation that is the same as the GloVe encodings (e.g., mean of zero and standard deviation of 0.4), with the same random encoding used consistently for each occurrence of the respective word. In some examples, when there is no GloVe and/or character n -gram encoding for a word (e.g., the word is

not in English) a value of zero is used. In some examples, during training, the ground truth answer is encoded similarly as also shown in Equation 1, where n corresponds to the number of words or tokens in the ground truth answer.

$$\begin{aligned} C &\in \mathbb{R}^{l \times d_{\text{emb}}} \\ [0026] \quad Q &\in \mathbb{R}^{m \times d_{\text{emb}}} \\ A &\in \mathbb{R}^{n \times d_{\text{emb}}} \end{aligned} \quad \text{Equation 1}$$

[0027] Independent Representations In some examples, independent representations C_{proj} and Q_{proj} for the encoded context and question, respectively, are generated using respective linear networks 320 and 325 according to Equation 2. In some examples, the respective linear networks 320 and 325 may also reduce the dimensionality of the encoded context and question to d . In some examples, the dimensionality d is 200. As shown in Equation 2, both of linear networks 320 and 325 use a same weight matrix W_1 and a same bias b_1 so the independent representations C_{proj} and Q_{proj} are projected onto the d dimensional space consistently. In some examples, the bias b_1 is optional and may be omitted. In some examples, weight matrix W_1 and bias b_1 are trained along with the rest of encoder 300.

$$\begin{aligned} [0028] \quad CW_1 + b_1 &= C_{\text{proj}} \in \mathbb{R}^{l \times d} \\ QW_1 + b_1 &= Q_{\text{proj}} \in \mathbb{R}^{m \times d} \end{aligned} \quad \text{Equation 2}$$

[0029] The projected representations for the context, C_{proj} , and the question, Q_{proj} , are passed to a shared, one-layer bidirectional Long Short-Term Memory Network (BiLSTM) 330 to form C_{ind} and Q_{ind} according to Equation 3.

$$\begin{aligned} [0030] \quad \text{BiLSTM}(C_{\text{proj}}) &= C_{\text{ind}} \in \mathbb{R}^{l \times d} \\ \text{BiLSTM}(Q_{\text{proj}}) &= Q_{\text{ind}} \in \mathbb{R}^{m \times d} \end{aligned} \quad \text{Equation 3}$$

[0031] BiLSTM 330 generates an output at each time step i as h_i as the concatenation of $h_i^{\rightarrow}$ and $h_i^{\leftarrow}$ according to Equation 4, where x is the input to biLSTM 330 and LSTM corresponds to a long-term short-term memory network. In some examples, BiLSTM 330 uses a dropout of 0.2 on inputs.

$$\begin{aligned}
 [0032] \quad h_i^{\rightarrow} &= \text{LSTM}(x_i, h_{i-1}^{\rightarrow}) \\
 h_i^{\leftarrow} &= \text{LSTM}(x_i, h_{i+1}^{\leftarrow})
 \end{aligned}
 \tag{Equation 4}$$

[0033] **Alignment.** The outputs C_{ind} and Q_{ind} of biLSTM 330 are used to generate coattended representations C_{proj} , Q_{proj} , C_{sum} , Q_{sum} , C_{coa} , and Q_{coa} using a coattention network 340. Coattention network 340 first aligns the encoded representations of the context and question sequences. In some examples, because some tokens from the question or the context sequence might not align well with any tokens in the other of the question and context sequences, a separate trained, dummy embedding is added to each of C_{ind} and Q_{ind} , which are now in $\mathbb{R}^{(l+1) \times d}$ and $\mathbb{R}^{(m+1) \times d}$.

[0034] Coattention network 340 then generates normalized dot-product similarity scores between each of the encoded tokens for the question and the context according to Equation 5, where $\text{softmax}(X)$ denotes a column-wise softmax that normalizes each column of the matrix X to have entries that sum to 1.

$$\begin{aligned}
 [0035] \quad \text{softmax}(C_{\text{ind}} Q_{\text{ind}}^T) &= S_{\text{cq}} \in \mathbb{R}^{(l \times 1) \times (m+1)} \\
 \text{softmax}(Q_{\text{ind}} C_{\text{ind}}^T) &= S_{\text{qc}} \in \mathbb{R}^{(m+1) \times (l+1)}
 \end{aligned}
 \tag{Equation 5}$$

[0036] **Dual Coattention.** Coattention network 340 then uses the normalized dot-product similarity scores S_{cq} and S_{qc} as weights to determine summaries of the information from the context and question sequences that is relevant to each token in the other of the context and question sequences according to Equation 6.

$$\begin{aligned}
 [0037] \quad S_{\text{cq}}^T C_{\text{ind}} &= C_{\text{sum}} \in \mathbb{R}^{(m+1) \times d} \\
 S_{\text{qc}}^T Q_{\text{ind}} &= Q_{\text{sum}} \in \mathbb{R}^{(l+1) \times d}
 \end{aligned}
 \tag{Equation 6}$$

[0038] Coattention network 340 then uses the coattended representations C_{sum} and Q_{sum} along with the use the normalized dot-product similarity scores S_{cq} and S_{qc} to transfer information gained from the alignments back to the original sequences according to Equation 7.

$$\begin{aligned}
 [0039] \quad S_{\text{qc}}^T C_{\text{sum}} &= C_{\text{coa}} \in \mathbb{R}^{(l+1) \times d} \\
 S_{\text{cq}}^T Q_{\text{sum}} &= Q_{\text{coa}} \in \mathbb{R}^{(m+1) \times d}
 \end{aligned}
 \tag{Equation 7}$$

[0040] The first column of the coattended representations C_{coa} and Q_{coa} correspond to the dummy embeddings previously added. Because this information is not needed, coattention network 340 drops that column of the matrices to get $C_{coa} \in \mathbb{R}^{l \times d}$ and $Q_{coa} \in \mathbb{R}^{m \times d}$.

[0041] **Compression.** Two biLSTMs 350 and 355 are used to compress the outputs from coattention network 340 in two separate groups. The outputs from coattention network 340 are concatenated into two groups, with each group being processed by biLSTMs 350 and 355, respectively, according to Equations 8 and 9. In some examples, BiLSTMs 350 and 355 use a dropout of 0.2 on inputs.

$$[0042] \quad \text{BiLSTM}_{comC} ([C_{proj}; C_{ind}; Q_{sum}; C_{coa}]) = C_{com} \in \mathbb{R}^{l \times d} \quad \text{Equation 8}$$

$$[0043] \quad \text{BiLSTM}_{comQ} ([Q_{proj}; Q_{ind}; C_{sum}; Q_{coa}]) = Q_{com} \in \mathbb{R}^{m \times d} \quad \text{Equation 9}$$

[0044] **Self-Attention.** The outputs C_{com} and Q_{com} of biLSTMs 350 and 355 are then passed to encoder portions of respective multi-layer self-attention based transformers. More specifically, the C_{com} output of biLSTM 350 is passed to a self-attention encoder 360 and the output C_{self1} of self-attention encoder 360 is passed to a self-attention encoder 370 to generate an output C_{self2} . In parallel, the Q_{com} output of biLSTM 355 is passed to a self-attention encoder 365 and the output Q_{self1} of self-attention encoder 365 is passed to a self-attention encoder 375 to generate an output Q_{self2} . In some examples, the self-attention encoders 360 and 370 capture long distance dependencies within the context and self-attention encoders 365 and 375 capture long distance dependencies within the question. And although encoder 300 is shown with two self-attention encoders for each of the context and question, encoder 300 may include only a single self-attention encoder or three or more self-attention encoders for one or both of the context and question. Each self-attention encoder 360, 365, 370, and 375 includes multi-head self-attention mechanisms that are followed by a position-wise fully connected feed-forward network along with a residual connection and layer normalization as is described in further detail below with respect to Figures 4 and 5.

[0045] Figure 4 is a simplified diagram of an attention network 400 according to some embodiments. As shown in Figure 4, attention network 400 receives a query $q \in \mathbb{R}^{d_q}$, a key $k \in \mathbb{R}^{d_k}$, and a value $v \in \mathbb{R}^{d_v}$. Each of the q , k , and v are subject to respective weights W^Q 410, W^K

420, and W^V 430 according to Equation 10. The weights W^Q 410, W^K 420, and W^V 430 are altered during training using back propagation.

$$\begin{aligned} Q &= qW^Q \in \mathbb{R}^{d_q} \\ K &= kW^K \in \mathbb{R}^{d_k} \\ V &= vW^V \in \mathbb{R}^{d_v} \end{aligned} \quad \text{Equation 10}$$

[0047] The resulting Q, K, and V vectors are passed through an attention transfer function 440, which generates a dot product of Q and K, which is then applied to V according to Equation 11.

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V \in \mathbb{R}^{d_v} \quad \text{Equation 11}$$

[0049] An addition and normalization module 450 is then used to combine the query q with the output from the attention transfer function to provide a residual connection that improves the rate of learning by attention network 400. Addition and normalization module 450 implements Equation 12 where μ and σ are the mean and standard deviation, respectively, of the input vector and g_i is gain parameter for scaling the layer normalization. The output from addition and normalization module 450 is the output of attention network 400.

$$\begin{aligned} &\text{LayerNorm}(\text{Attention}(Q, K, V) + q) \\ \text{LayerNorm}(a_i) &= g_i \frac{a_i - \mu}{\sigma} \end{aligned} \quad \text{Equation 12}$$

[0051] Attention network 400 is often used in two variant forms. The first variant form is a multi-head attention network where multiple attention networks consistent with attention network 400 are implemented in parallel, which each of the “heads” in the multi-head attention network having its own weights W^Q 410, W^K 420, and W^V 430, which are initialized to different values and thus trained to learn different encodings. The outputs from each of the heads are then concatenated together to form the output of the multi-head attention network. The second variant form is a self-attention network that is a multi-head attention network where the q, k, and v inputs are the same for each head of the attention network.

[0052] Self-attention based layers are further described in Vaswani, et al., “Attention is All You Need,” arXiv preprint arXiv:1706.03762, submitted June 12, 2017, which is hereby incorporated by reference in its entirety.

[0053] Figure 5 is a simplified diagram of a layer 500 for an attention-based transformer network according to some embodiments. As shown in Figure 5, layer 500 includes an encoder 510 and a decoder 520.

[0054] Encoder 510 receives layer input (e.g., from an input network for a first layer in an encoding stack or from layer output of a next lowest layer for all other layers of the encoding stack) and provides it to all three (q, k, and v) inputs of a multi-head attention network 511, thus multi-head attention network 511 is configured as a self-attention network. Each head of multi-head attention network 511 is consistent with attention network 400. In some examples, multi-head attention network 511 includes three heads, however, other numbers of heads such as two or more than three are possible. In some examples, each attention network has a dimension of 200 and a hidden size of 128. The output of multi-head attention network 511 is provided to a feed forward network 512 with both the input and output of feed forward network 512 being provided to an addition and normalization module 513, which generates the layer output for encoder 510. In some examples, feed forward network 512 is a two-layer perceptron network with a rectified linear unit (ReLU) activation, which implements Equation 13 where γ is the input to feed forward network 512 and M_i and b_i are the weights and biases respectively of each of the layers in the perceptron network. In some examples, addition and normalization module 513 is substantially similar to addition and normalization module 450.

[0055]
$$FF(\gamma) = \max(0, \gamma M_1 + b_1) M_2 + b_2$$
 Equation 13

[0056] Decoder 520 receives layer input (e.g., from an input network for a first layer in a decoding stack or from layer output of a next lowest layer for all other layers of the decoding stack) and provides it to all three (q, k, and v) inputs of a multi-head attention network 521, thus multi-head attention network 521 is configured as a self-attention network. Each head of multi-head attention network 521 is consistent with attention network 400. In some examples, multi-head attention network 521 includes three heads, however, other numbers of heads such as two or more than three are possible. The output of multi-head attention network 511 is provided as

the q input to another multi-head attention network 522 and the k and v inputs of multi-head attention network 522 are provided with the output from the encoder. Each head of multi-head attention network 521 is consistent with attention network 400. In some examples, multi-head attention network 522 includes three heads, however, other numbers of heads such as two or more than three are possible. In some examples, each attention network has a dimension of 200 and a hidden size of 128. The output of multi-head attention network 522 is provided to a feed forward network 523 with both the input and output of feed forward network 523 being provided to an addition and normalization module 524, which generates the layer output for encoder 510. In some examples, feed forward network 523 and addition and normalization module 524 are substantially similar to feed forward network 512 and addition and normalization module 513, respectively.

[0057] Referring back to Figure 3, each of self-attention encoders 360, 365, 370, and/or 375 are consistent with encoder 510 and generate self-attention based encodings according to Equation 14, where SAEncode(X) corresponds to the encoding performed by encoder 510 and X corresponds to the input received by multi-head attention network 511.

$$\begin{aligned}
 & \text{SAEncode}(C_{\text{com}}) = C_{\text{self1}} \in \mathbb{R}^{l \times d} \\
 & \text{SAEncode}(C_{\text{self1}}) = C_{\text{self2}} \in \mathbb{R}^{l \times d} \\
 & \text{SAEncode}(Q_{\text{com}}) = Q_{\text{self1}} \in \mathbb{R}^{m \times d} \\
 & \text{SAEncode}(Q_{\text{self1}}) = Q_{\text{self2}} \in \mathbb{R}^{m \times d}
 \end{aligned}
 \tag{Equation 14}$$

[0059] **Final Encodings.** Final encodings for the context are then generated from the C_{self2} output from self-attention encoder 370 by integrating the context information over time using a BiLSTM 380 according to Equation 15. Similarly, final encodings for the question are then generated from the Q_{self2} output from self-attention encoder 375 by integrating the question information over time using a BiLSTM 385 according to Equation 15. In some examples, BiLSTMs 380 and 385 use a dropout of 0.2 on inputs.

$$\begin{aligned}
 & \text{BiLSTM}_{\text{finC}}(C_{\text{self2}}) = C_{\text{fin}} \in \mathbb{R}^{l \times d} \\
 & \text{BiLSTM}_{\text{finQ}}(Q_{\text{self2}}) = Q_{\text{fin}} \in \mathbb{R}^{m \times d}
 \end{aligned}
 \tag{Equation 15}$$

[0061] Figure 6 is a simplified diagram of a decoder according to some embodiments. Decoder 600 uses an iterative approach to generate an answer a single word or token at a time.

Figure 6 represents temporal/positional relationships using subscripts that indicate which iteration in the generation of the answer is being referred to. (E.g., X_t refers to the value for X from the t -th iteration and X_{t-1} refers to the value for X from the $(t-1)$ st iteration, that is the iteration previous to the t -th iteration.) As described above with respect to Equation 1, when decoder 600 is used during training, the ground truth answer is initially encoded using the same encoding used for the context and the question.

[0062] Answer Representations. An embedding and encoder 610 is used to project the partially formed answer onto a d -dimensional space similar to the embeddings for the context and question according to Equation 16.

$$\text{[0063]} \quad AW_2 + b_2 = A_{\text{proj}} \in \mathbb{R}^{n \times d} \quad \text{Equation 16}$$

[0064] Because the words or tokens in the answer lack both recurrence and convolution, positional encodings PE are added to A_{proj} according to Equation 17.

$$\text{[0065]} \quad \text{PE}[t, k] = \begin{cases} \sin(t/10000^{k/2d}) & k \text{ is even} \\ \cos(t/10000^{(k-1)/2d}) & k \text{ is odd} \end{cases} \quad \text{Equation 17}$$

$$A_{\text{proj}} + \text{PE} = A_{\text{ppr}} \in \mathbb{R}^{n \times d}$$

[0066] MultiHead Decoder Attention. The output of embedding and encoder 610 is then passed to a self-attention decoder 620 whose output is then passed to a self-attention decoder 630 to generate A_{SELF} according to Equation 18, where $\text{SADecode}(A, B)$ corresponds to the decoding performed by decoder 520, A corresponds to the layer input received by multi-head attention network 521 and B corresponds to the encoding input received by multi-head attention network 522. Self-attention is used so that decoder 600 is aware of previous outputs and attention over the context to prepare for the next word or token to be added to the answer. However, during training and because the decoder operates step by step, appropriate entries of the embedded and encoded ground truth answer are set to a large negative number to prevent decoder 600 from seeing future time-steps when applying Equation 18.

$$\text{[0067]} \quad \begin{aligned} \text{SADecode}(A_{\text{ppr}}, C_{\text{fin}}) &= A_1 \in \mathbb{R}^{l \times d} \\ \text{SADecode}(A_1, C_{\text{fin}}) &= A_{\text{SELF}} \in \mathbb{R}^{l \times d} \end{aligned} \quad \text{Equation 18}$$

[0068] **Intermediate Decoder State.** A LSTM 640 generates an intermediate state h_t for the current iteration using the previous answer word or token $ASELF_{t-1}$, the context-decoder state $\tilde{c}_{t-1}$ from the previous iteration, and the intermediate state h_{t-1} from the previous iteration according to Equation 19. In some examples, LSTM 640 uses a dropout of 0.2 on inputs.

$$[0069] \quad \text{LSTM}([ASELF_{t-1}; \tilde{c}_{t-1}], h_{t-1}) = h_t \in \mathbb{R}^d \quad \text{Equation 19}$$

[0070] **Context and Question Attention.** The intermediate state h_t from LSTM 640 is provided to a context and question attention network 650 that is first generates attention weights α_{C_t} and α_{Q_t} over the final encoding of the context C_{fin} and the final encoding of the question Q_{fin} , respectively, according to Equation 20, where W_2 and W_3 are trainable weight matrices. In some examples, context and question attention network 650 may optionally include trainable biases b_2 and b_3 within Equation 20. The attention weights allow decoder 600 to focus on encoded information relevant to each iteration t .

$$[0071] \quad \begin{aligned} \text{softmax}(C_{fin}(W_2 h_t)) &= \alpha_{C_t} \in \mathbb{R}^l \\ \text{softmax}(Q_{fin}(W_3 h_t)) &= \alpha_{Q_t} \in \mathbb{R}^m \end{aligned} \quad \text{Equation 20}$$

[0072] **Recurrent Context State.** Context and question attention network 650 then combines the final encoding of the context C_{fin} and the final encoding of the question Q_{fin} as weighted by the attention weights α_{C_t} and α_{Q_t} , respectively, with the hidden state h_t according to Equation 21 to generate context representation $\tilde{C}_t$ and question representation $\tilde{Q}_t$, where W_4 and W_5 are trainable weights and $\tanh$ is the hyperbolic tangent transfer function. In some examples, context and question attention network 650 may optionally include trainable biases b_4 and b_5 within Equation 21. In some examples, other non-linear transfer functions may be used with Equation 21.

$$[0073] \quad \begin{aligned} \tanh(W_4 C_{fin}^T \alpha_{C_t}; h_t) &= \tilde{C}_t \in \mathbb{R}^d \\ \tanh(W_5 Q_{fin}^T \alpha_{Q_t}; h_t) &= \tilde{Q}_t \in \mathbb{R}^d \end{aligned} \quad \text{Equation 21}$$

[0074] **Multi-Pointer-Generator** A pointer generator 660 is used to not only generate distributions over the words or tokens in the context and the words or tokens in the question, but also a distribution over words or tokens v from a generative vocabulary. The distributions assign likelihoods that each of the words or tokens from the context, question, and vocabulary are the

next word or token to be selected by decoder 600 in the current iteration t . The distributions are generated according to Equation 22, where W_v is a trainable weight matrix. In some examples, pointer generator 660 may optionally include a trainable bias b_v within Equation 22 to generate the distribution p_v over the generative vocabulary within Equation 22.

$$\begin{aligned}
 \sum_{i:c_i=w_t} (\alpha_{c_t})_i &= p_c(W_t) \in \mathbb{R}^n \\
 \sum_{i:q_i=w_t} (\alpha_{q_t})_i &= p_q(W_t) \in \mathbb{R}^m \\
 \text{softmax}(W_v \tilde{c}_t) &= p_v(W_t) \in \mathbb{R}^v
 \end{aligned}
 \tag{0075} \quad \text{Equation 22}$$

[0076] Pointer generator 660 then extends the distributions p_c , p_q , and p_v over the union of the tokens in the context, question, and generative vocabulary by setting missing entries in each distribution to 0 so that each distribution p_c , p_q , and p_v is in $\mathbb{R}^{l+m+v}$.

[0077] A switch 670 is then used to determine the importance of the context and the question in determining the next word or token for the current iteration according to Equation 23, where σ is a sigmoid transfer function, such as logsig , tansig , and/or the like.

$$\begin{aligned}
 \sigma(W_{pv}[\tilde{c}_t; h_t; \text{ASELF}_{t-1}]) &= \gamma \in [0,1] \\
 \sigma(W_{cq}[\tilde{q}_t; h_t; \text{ASELF}_{t-1}]) &= \lambda \in [0,1]
 \end{aligned}
 \tag{0078} \quad \text{Equation 23}$$

[0079] Switch 670 then generates a composite distribution according to Equation 24 and the next word or token for the current iteration is selected based on the word or token from the composite distribution having the greatest likelihood. The selected word or token is then appended to the answer and fed back to embedding and encoder 610 where another iteration of decoder 600 begins. Decoder 600 continues to iterate until a maximum number of words or tokens are selected and/or a sentinel word or token indicating the end of the answer is selected.

$$\gamma P_v(w_t) + (1 - \gamma)[\lambda p_c(w_t) + (1 - \lambda)p_q(w_t)] = p(w_t) \in \mathbb{R}^{l+m+v} \tag{0080} \quad \text{Equation 24}$$

[0081] According to some embodiments, during training, encoder 300 and decoder 600 are trained by back propagation using a token-level negative log-likelihood loss function $\mathcal{L}$ over each time-step according to Equation 25, where a_t corresponds to the t -th word or token in the answer.

$$\mathcal{L} = - \sum_t^T \log p(a_t) \tag{0082} \quad \text{Equation 25}$$

[0083] Because the networks of Figures 3-6 are used for multiple tasks (e.g., question answering, machine translation, document summarization, database query generation, sentiment analysis, natural language inference, semantic role labeling, relation extraction, goal oriented dialogue, and pronoun resolution) and shares its parameters for the various layers and networks across each of the task types, it may be susceptible to catastrophic forgetting if it is not trained carefully. To address this, in some embodiments, the networks of Figures 3-6 may be trained according to a modified joint strategy where the networks of Figures 3-6 are trained using an ordering where training samples are presented so as to train the networks of Figures 3-6 against a balanced mix of each of the task types concurrently. That is, the order in which training samples are presented to the networks of Figures 3-6 selects consecutive training samples or consecutive small groups (e.g., 2-10 or so) training samples from different task types. In some examples, the joint strategy includes selecting a training sample (context *c*, questions *q*, and ground truth answer *a*) from a different one of the task types with each iteration of the training. The goal of the joint strategy is to train against each of the task types concurrently without overly focusing on one task type over another.

[0084] Figure 7 is a simplified diagram of a method of multitask learning according to some embodiments. One or more of the processes 710-780 of method 700 may be implemented, at least in part, in the form of executable code stored on non-transitory, tangible, machine-readable media that when run by one or more processors may cause the one or more processors to perform one or more of the processes 710-780. In some embodiments, method 700 may be used as a hybrid training strategy for training the networks of Figures 3-6, however, method 700 may also be used to train other multitasking systems other than the networks of Figures 3-6. In some examples, method 700 may be consistent with a curriculum and/or an anti-curriculum strategy. In some embodiments, the task types trained by method 700 may include any of various natural language processing task types, such as question answering, machine translation, document summarization, database query generation, sentiment analysis, natural language inference, semantic role labeling, relation extraction, goal oriented dialogue, pronoun resolution, and/or the like.

[0085] At a process 710, a training sample is selected according to a first training strategy. In some embodiments, the first training strategy is a joint training strategy where training samples

are selected from a subset of task types for which the network is being trained. In some examples, the subset of task types may be selected from a full set of task types. In some examples, the full set of task types may include one or more task types selected from question answering, machine translation, document summarization, database query generation, sentiment analysis, natural language inference, semantic role labeling, relation extraction, goal oriented dialogue, pronoun resolution, and/or the like.

[0086] In some embodiments, the subset of task types may be selected according to a curriculum strategy where the training sample is selected from those task types whose training converges in a relatively small number of training iterations. In some examples, the subset of task types for the curriculum training may include one or more task types selected from database query generation, sentiment analysis, semantic role labeling, relation extraction, goal oriented dialogue, pronoun resolution, and/or the like.

[0087] In some embodiments, the subset of task types may be selected according to an anti-curriculum strategy where the training sample is selected from those task types which are characterized as being more difficult to learn, have longer answer sequences, and/or involve different types of decoding. In some examples, the subset of task types for the anti-curriculum training may include one or more task types selected from question answering, summarization, machine translation, and/or natural language inference.

[0088] In some examples, the selected training sample includes a natural language context and a natural language question corresponding to the context and the question, respectively, as well as a ground truth natural language answer.

[0089] At a process 720, the selected training sample is presented to a system. In some examples, the system includes encoder 300 and decoder 600. When the training sample is applied to the system it is fed forward through the various layers of the system according to the currently trained parameters (e.g., weights and biases) and an answer is generated. In some examples, the answer is a natural language phrase.

[0090] At a process 730, the system is adjusted based on error. The answer generated by the system during process 720 is compared to the ground truth answer for the selected training

sample and the error for the selected training sample is determined. The error may then be fed back to the system using back propagation to update the various parameters (e.g., weights and biases) of the layers. In some examples, the back propagation may be performed using the stochastic gradient descent (SGD) training algorithm, the adaptive moment estimation (ADAM) training algorithm, and/or the like. In some examples, the meta parameters of ADAM are set to $(\beta_1, \beta_2, \epsilon) = (0.9, 0.98, 10^{-9})$. In some examples, the learning rate is increased linearly from 0 to 2.5×10^{-3} for the first 800 iterations before being decayed at $k^{-1/2}$, where k is the iteration count. In some examples, the gradients used for the back propagation may be clipped to 1.0.

[0091] At a process 740, it is determined whether to switch from training based on the subset of task types to training based on the full set of task types. In some examples, the decision to switch to the full set of task types occurs after a predetermined number of training samples have been presented. In some examples, the predetermined number of training samples may be 300,000, however, other predetermined numbers of training samples are possible. In some examples, the decision to switch to the full set of task types occurs after each of the training samples for each of the task types in the subset of task types has been selected a predetermined number of times. In some examples, the predetermined number of times may be five, although any other number such as three, four, and/or six or more may also be used. In some examples, one or more other factors may be used to make the determination about when to switch to training using the full set of task types. In some examples, the one or other factors may include monitoring changes in performance metrics for each of the task types in the subset of task types with each pass through the training samples and making the switch when an improvement in each of the performance metrics after each pass improves by less than a threshold amount. When it is determined not to switch to training on the full set of task types, method 700 returns to process 710 where training samples continue to be selected from the subset of task types. When it is determined to switch to training from the full set of task types, selection of the training samples occurs using the full set of task types beginning with a process 750.

[0092] At the process 750, a training sample is selected from the full set of task types. In some examples, the full set of task types includes one or more tasks selected from question answering, machine translation, document summarization, database query generation, sentiment

analysis, natural language inference, semantic role labeling, relation extraction, goal oriented dialogue, pronoun resolution, and/or the like.

[0093] At a process 760, the selected training sample is presented to the system using substantially the same process as process 720.

[0094] At a process 770, the system is adjusted based on error using substantially the same process as process 730.

[0095] At a process 780, it is determined whether the training is complete. In some examples, training is complete after a predetermined number of training samples have been presented. In some examples, the training is complete after the training samples for each of the task types in the full set of task types has been presented to the system a predetermined number of times. In some examples, the predetermined number of times may be eight, although any other number such as two to seven and/or nine or more may also be used. In some examples, one or more other factors may be used to make the determination about when training is complete. In some examples, the one or other factors may include monitoring changes in performance metrics for each of the task types with each pass through the training samples and noting that training is complete when an improvement in each of the performance metrics after each pass improves by less than a threshold amount. When it is determined that training is not complete, method 700 returns to process 740 where training samples continue to be selected from the full set of task types. When it is determined that training is complete, method 700 ends and the trained system may now be used for any of the tasks for which it is trained.

[0096] After training is complete, the trained system may be used for any of the task types using a process substantially similar to process 720 and/or 760 where a context *c* and a question *q* may be presented to the system and fed forward through the various layers of the system according to the parameters (e.g., weights and biases) trained according to method 700. The generated answer then corresponds to the response to the presented context *c* and question *q*.

[0097] Figure 8 is a simplified diagram of a summary of training sets according to some embodiments. More specifically, Figure 8 shows each task type, the dataset used, the number of

training samples in the dataset, the number of samples in the development/validation samples in the dataset, the number of test samples in the dataset, and the performance metric used.

[0098] Training samples for the question answering task type are based on the Stanford Question Answering Dataset (SQuAD), which includes training samples based on questions related to paragraph samples from Wikipedia articles. The performance metric used for the question answering task type is the normalized F1 (nF1) score.

[0099] Training samples for the machine translation task type are based on the International Workshop on Spoken Language Translation English to German (IWSLT EN->DE) training set, which contains sentence pairs transcribed from TED talks. The performance metric used for the machine translation task type is the BLEU score.

[00100] Training samples for the summarization task type are selected from the CNN daily mail (CNN/DM) training set. The performance metric used for the summarization task type is the ROUGE score.

[00101] Training samples for the natural language inference task type are selected from the Multi-genre Natural Language Inference Corpus (MNLI) training set. The performance metric used for the natural language inference task type is the normalized F1 (nF1) score.

[00102] Training samples for the sentiment analysis task type are based on the Stanford Sentiment Treebank (SST). The SST includes training samples based on movie reviews and their sentiment. The performance metric used for the sentiment classification task type is the normalized F1 (nF1) score.

[00103] Training samples for the semantic role labeling task type are selected from the Question-Answer Semantic Role Labeling (QA-SRL) training set. The performance metric used for the semantic role labeling task type is the normalized F1 (nF1) score.

[00104] Training samples for the relationship extraction task type are selected from the Question-Answer Zero-shot Relation Extraction (QA-ZRE) training set. The performance metric used for the relationship extraction task type is the F1 score.

[00105] Training samples for the goal oriented dialog task type are selected from the Wizard of Oz (WOZ) training set. The performance metric used for the goal oriented dialog task type is the exact match (EM) score.

[00106] Training samples for the database query generation task type are selected from the WikiSQL training set. The performance metric used for the database query generation task type is the exact match (EM) score.

[00107] Training samples for the pronoun resolution task type are selected from the Modified Winograd Schema Challenge (MWSC) training set. The performance metric used for the pronoun resolution task type is the normalized F1 (nF1) score.

[00108] Figure 9 is a simplified diagram of results from single task and multitask learning according to some embodiments. More specifically, Figure 8 shows the results of training systems, such as encoder 300 and decoder 600, against various task types including question answering, machine translation, document summarization, database query generation, sentiment analysis, natural language inference, semantic role labeling, relation extraction, goal oriented dialogue, pronoun resolution.

[00109] Results of the training are further evaluated according to the composite Decathlon Score (decaScore). The decaScore includes an additive combination of the task-specific performance metrics. Because each of the individual performance metrics fall between 0 and 100, the decaScore falls between 0 and 1000 for a composite performance score across the ten task types.

[00110] The single-task training columns of Figure 9 show the baseline results for each of the task types against their respective performance metrics when trained in isolation using the pointer generator sequence-to-sequence model (S2S) of See, et al. "Get to the Point: Summarization with Pointer-generator Networks," arXiv preprint arXiv:1704.04368, 2017; a version of S2S where the lower layers of S2S are replaced with self-attentive encoder and decoder layers (w/SAtt) of Vaswani, et al., "Attention is All You Need," arXiv preprint arXiv:1706.03762, 2017; a version of S2S augmented with a coattention mechanism (+CAtt) that

builds representations of both the context and question separately and explicitly models how the two should interact; and a variation of +CAtt with a question pointer added (+QPtr).

[00111] The multitask training columns of Figure 9 further shows the results of training each of the models using a joint training strategy where training samples are selected from each of the task types throughout the training. A further result (+ACurr) of training of encoder 300 and decoder 600 using an anti-curriculum version of the training of method 700 where the initial subset of task types includes only training samples from the question answering (SQuAD) training set. As Figure 8 indicates, the anti-curriculum training of method 700 using encoder 300 and decoder 600 yields better overall composite results (as measured by the decaScore) for each of these models.

[00112] Figure 10 is a simplified diagram of results for different training strategies according to some embodiments. More specifically, Figure 10 shows the differences between training encoder 300 and decoder 600 using a fully joint (Fully Joint) strategy; using method 700 employing a curriculum based approach where the subset of task types are limited to database query generation, sentiment analysis, semantic role labeling, relation extraction, goal oriented dialogue, and pronoun resolution (Curriculum); and using method 700 employing an anti-curriculum based approach where the subset of task types are limited to question answering (SQuAD), question answering, machine translation, and summarization (+IWSLT+CNN/DM), and question answering, machine translation, summarization, and natural language inference (+MNL). As shown the curriculum-based variation of method 700 did not perform as well as the fully joint strategy. However, the use of method 700 with an anti-curriculum approach using only question answering training samples in the initial subset (SQuAD) showed an overall improvement over the fully joint strategy and an improvement or slight degradation against each of the task types. Overall improvements in performance did not occur when the subset of task types was extended to include training samples from other task types.

[00113] Figure 11 is a simplified diagram of source for word selection for answers according to some embodiments. More specifically, Figure 11 shows, for each of the task types, which of the context, question, or generative vocabulary is more likely to be the source of the words or tokens selected for the answer based on the γ and λ values selected by switch 670 and used in

Equations 23 and 24. As shown, the sentiment analysis (SST), natural language inference (MNLI), and pronoun resolution (MWSC) task types prefer words or tokens from the question because the question tends to include the classification options for the answer. The machine translation (IWSLT) and goal oriented dialog task types (WOZ) prefer words or tokens from the vocabulary because neither the context nor the question tend to include the desired words for the answer.

[00114] Figure 12A is a simplified diagram of results from different training approaches for a new machine translation task type according to some embodiments. More specifically, Figure 12A shows the results of training against a new machine translation task type using training samples from the IWSLT English to Czech (IWSLT En->Cs) training set using encoder 300 and decoder 600 pre-trained according to method 700 (curve 1210) in comparison to encoder 300 and decoder 600 trained from a random start against only the IWSLT English to Czech training samples (curve 1220).

[00115] Figure 12B is a simplified diagram of results from different training approaches for a new classification task according to some embodiments. More specifically, Figure 12B shows the results of training against a new named entity recognition (NER) task type training samples using encoder 300 and decoder 600 pre-trained according to method 700 (curve 1230) in comparison to encoder 300 and decoder 600 trained from a random start against only the NER training samples (curve 1240).

[00116] As shown, the pretraining using method 700 (curves 1210 and 1230) demonstrate better performance results than single task training (curves 1220 and 1240), thus demonstrating the advantages of using the multitask training of method 700 to pretrain for natural language processing tasks and additionally for adapting an already trained network based on encoder 300 and decoder 600 against new tasks.

[00117] Some examples of computing devices, such as computing device 200 may include non-transitory, tangible, machine readable media that include executable code that when run by one or more processors (e.g., processor 210) may cause the one or more processors to perform the processes of method 700 and/or emulate the structures of Figures 3-6. Some common forms of machine readable media that may include the processes of method 700 and the emulations of

Figures 3-6 are, for example, floppy disk, flexible disk, hard disk, magnetic tape, any other magnetic medium, CD-ROM, any other optical medium, punch cards, paper tape, any other physical medium with patterns of holes, RAM, PROM, EPROM, FLASH-EPROM, any other memory chip or cartridge, and/or any other medium from which a processor or computer is adapted to read.

[00118] This description and the accompanying drawings that illustrate inventive aspects, embodiments, implementations, or applications should not be taken as limiting. Various mechanical, compositional, structural, electrical, and operational changes may be made without departing from the spirit and scope of this description and the claims. In some instances, well-known circuits, structures, or techniques have not been shown or described in detail in order not to obscure the embodiments of this disclosure. Like numbers in two or more figures represent the same or similar elements.

[00119] In this description, specific details are set forth describing some embodiments consistent with the present disclosure. Numerous specific details are set forth in order to provide a thorough understanding of the embodiments. It will be apparent, however, to one skilled in the art that some embodiments may be practiced without some or all of these specific details. The specific embodiments disclosed herein are meant to be illustrative but not limiting. One skilled in the art may realize other elements that, although not specifically described here, are within the scope and the spirit of this disclosure. In addition, to avoid unnecessary repetition, one or more features shown and described in association with one embodiment may be incorporated into other embodiments unless specifically described otherwise or if the one or more features would make an embodiment non-functional.

[00120] Although illustrative embodiments have been shown and described, a wide range of modification, change and substitution is contemplated in the foregoing disclosure and in some instances, some features of the embodiments may be employed without a corresponding use of other features. One of ordinary skill in the art would recognize many variations, alternatives, and modifications. Thus, the scope of the invention should be limited only by the following claims, and it is appropriate that the claims be construed broadly and in a manner consistent with the scope of the embodiments disclosed herein.

WHAT IS CLAIMED IS:

1. A system for natural language processing, the system comprising:
 - a multi-layer encoder for encoding first words from a context and second words from a question in parallel;
 - a multi-layer decoder for decoding the encoded context and the encoded question;
 - a pointer generator for generating distributions over the first words from the context, the second words from the question, and third words in a vocabulary based on an output from the decoder; and
 - a switch for:
 - generating a weighting of the distribution over the first words from the context, the distribution over the second words from the question, and the distribution over the third words in the vocabulary;
 - generating a composite distribution based on the weighting of the distribution over the first words from the context, the distribution over the second words from the question, and the distribution over the third words in the vocabulary; and
 - selecting words for inclusion in an answer using the composite distribution.
2. The system of claim 1, wherein the context and the question correspond to a natural language processing task type selected from question answering, machine translation, document summarization, database query generation, sentiment analysis, natural language inference, semantic role labeling, relation extraction, goal oriented dialogue, and pronoun resolution.
3. The system of claim 1 or 2, wherein the multi-layer encoder comprises:
 - a coattention network for determining a coattention between the first words in the context and the second words in the question; and
 - parallel bi-directional long short term memories to compress outputs from the coattention network.
4. The system of any one of claims 1-3, wherein the multi-layer encoder comprises:
 - parallel self-attention encoders for generating an attention across the context and an attention across the question in parallel; and

parallel bi-directional long short term memories for generating final encodings of the context and the question in parallel based on the generated attention.

5. The system of any one of claims 1-4 wherein the multi-layer encoder comprises:

parallel encoding layers for encoding the first words from the context and the second words from the question in parallel;

parallel linear networks for projecting the encodings of the first words from the context and the second words from the question in parallel; and

a bidirectional long short term memory for further encoding the projections of the encodings.

6. The system of any one of claims 1-5, wherein the multi-layer decoder comprises:

an encoding and embedding layer for encoding and embedding an intermediate version of the answer;

a self-attention decoder for generating an attention between the encoded and embedded intermediate version of the answer and a final encoding of the context;

a long short term memory for generating an intermediate decoder state from outputs of the self-attention decoder; and

a context and question attention network for generating context and question decoder states based on a final encoding of the context, a final encoding of the question, and the intermediate decoder state.

7. The system of any one of claims 1-6, wherein the system is trained against a subset of task types before being trained against a full set of task types that the system is designed to process.

8. The system of claim 7, wherein the subset of task types are selected according to a curriculum strategy.

9. The system of claim 7, wherein the subset of task types are selected according to an anti-curriculum strategy.

10. A method for natural language processing, the method comprising:
 - encoding first words from a context and second words from a question in parallel using a multi-layer encoder;
 - decoding the encoded context and the encoded question using a multi-layer decoder;
 - generating distributions over the first words from the context, the second words from the question, and third words in a vocabulary based on an output from the decoder;
 - generating a weighting of the distribution over the first words from the context, the distribution over the second words from the question, and the distribution over the third words in the vocabulary;
 - generating a composite distribution based on the weighting of the distribution over the first words from the context, the distribution over the second words from the question, and the distribution over the third words in the vocabulary; and
 - selecting words for inclusion in an answer using the composite distribution.
11. The method of claim 10, wherein the context and the question correspond to a natural language processing task type selected from question answering, machine translation, document summarization, database query generation, sentiment analysis, natural language inference, semantic role labeling, relation extraction, goal oriented dialogue, and pronoun resolution.
12. The method of claim 10 or 11, further comprising:
 - determining a coattention between the first words in the context and the second words in the question.
13. The method of any one of claims 10-12, further comprising:
 - generating an attention across the context and an attention across the question in parallel;
 - and
 - generating final encodings of the context and the question in parallel based on the attention.
14. The method of any one of claims 10-13, further comprising:
 - encoding the first words from the context and the second words from the question in parallel;

projecting the encodings of the first words from the context and the second words from the question in parallel; and

further encoding the projections of the encodings.

15. The method of any one of claims 10-14, further comprising:

encoding and embedding an intermediate version of the answer;

generating an attention between the encoded and embedded intermediate version of the answer and a final encoding of the context;

generating an intermediate decoder state from the generated attention; and

generating context and question decoder states based on a final encoding of the context, a final encoding of the question, and the intermediate decoder state.

16. The method of any one of claims 10-15, further comprising training the multi-layer encoder and the multi-layer decoder against a subset of task types before being training the multi-layer decoder and the multi-layer encoder against a full set of task types.

17. The method of claim 16, wherein the subset of task types are selected according to a curriculum strategy.

18. The method of claim 16, wherein the subset of task types are selected according to an anti-curriculum strategy.

19. A non-transitory machine-readable medium comprising executable code which when executed by one or more processors associated with a computing device are adapted to cause the one or more processors to perform a method comprising:

encoding first words from a context and second words from a question in parallel using a multi-layer encoder;

decoding the encoded context and the encoded question using a multi-layer decoder;

generating distributions over the first words from the context, the second words from the question, and third words in a vocabulary based on an output from the decoder;

generating a weighting of the distribution over the first words from the context, the distribution over the second words from the question, and the distribution over the third words in the vocabulary;

generating a composite distribution based on the weighting of the distribution over the first words from the context, the distribution over the second words from the question, and the distribution over the third words in the vocabulary; and

selecting words for inclusion in an answer using the composite distribution.

20. The non-transitory machine-readable medium of claim 19, wherein the context and the question correspond to a natural language processing task type selected from question answering, machine translation, document summarization, database query generation, sentiment analysis, natural language inference, semantic role labeling, relation extraction, goal oriented dialogue, and pronoun resolution.

	<u>Question</u>	<u>Context</u>	<u>Answer</u>
105	To whom did the Virgin Mary appear?	...the Virgin Mary reputedly appeared to Saint Bernadette Soubirous...	Saint Bernadette Soubirous
110	What is the translation from English to German?	Most of the planet is ocean water.	Der Großteil der Erde ist Meerwasser
115	What is the summary?	Harry Potter star Daniel Radcliffe gains access to a reported £320 million fortune...	Harry Potter star Daniel Radcliffe gets £320 fortune...
120	Hypothesis: Product and geography are what make cream skimming work. Entailment, neutral, or contradiction?	Premise: Conceptually cream skimming has two basic dimensions — product and geography.	Entailment
125	Is this sentence positive or negative?	A stirring, funny and finally transporting re-imagining of Beauty and the Beast and 1930s horror film.	positive
	<u>Question</u>	<u>Context</u>	<u>Answer</u>
130	What rows?	There are four boat clubs that row on the River Dee:...	four boat clubs
135	What was the title that Timur held?	Both Timur and Demir are popular male names in Turkey today.	unanswerable
140	What information have you gathered about the user?	Are there any Eritrean restaurants in town?	food: Eritrean
145	What is the translation from English to SQL?	The table has column names... Tell me what the notes are for South Australia	SELECT note from table WHERE 'Current Slogan' = 'South Australia'
150	Who had given help? Susan or Joan?	Joan made sure to thank Susan for all the help she had given.	Susan

FIG. 1

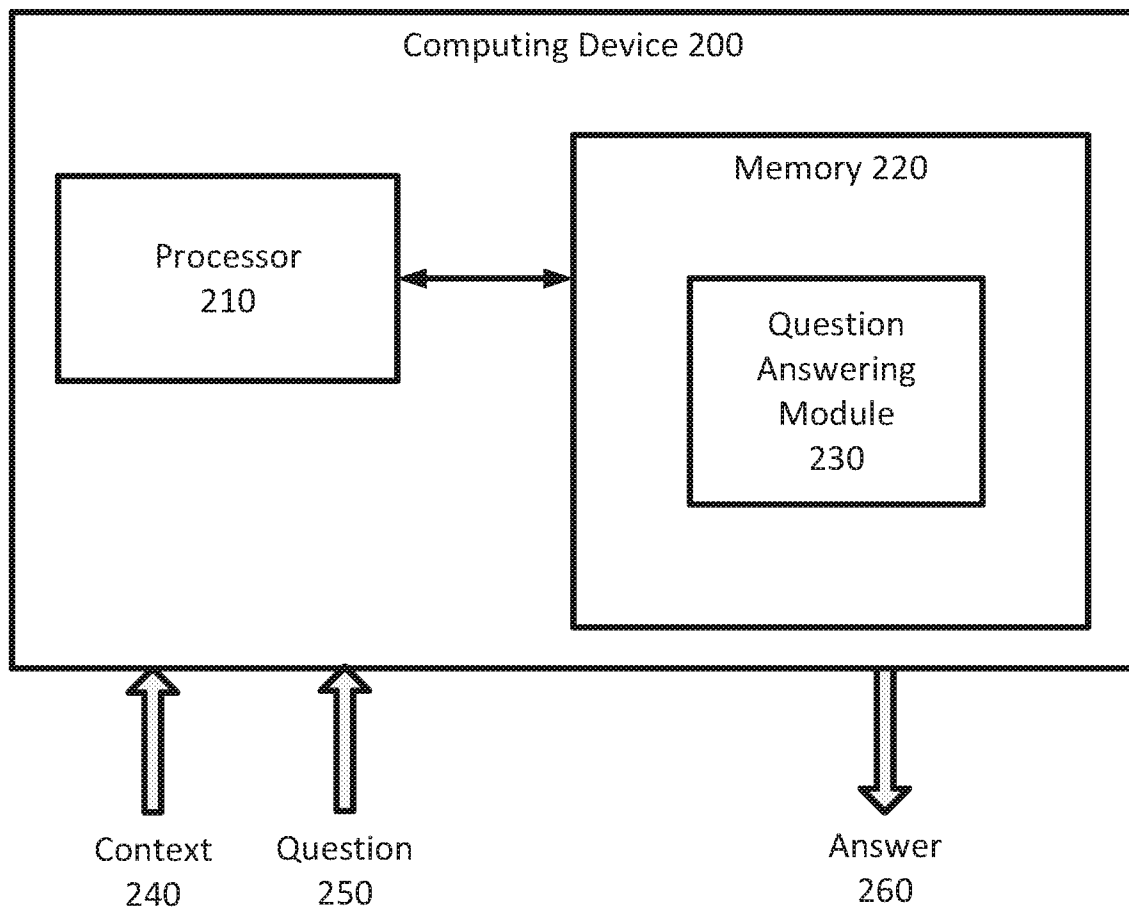


FIG. 2

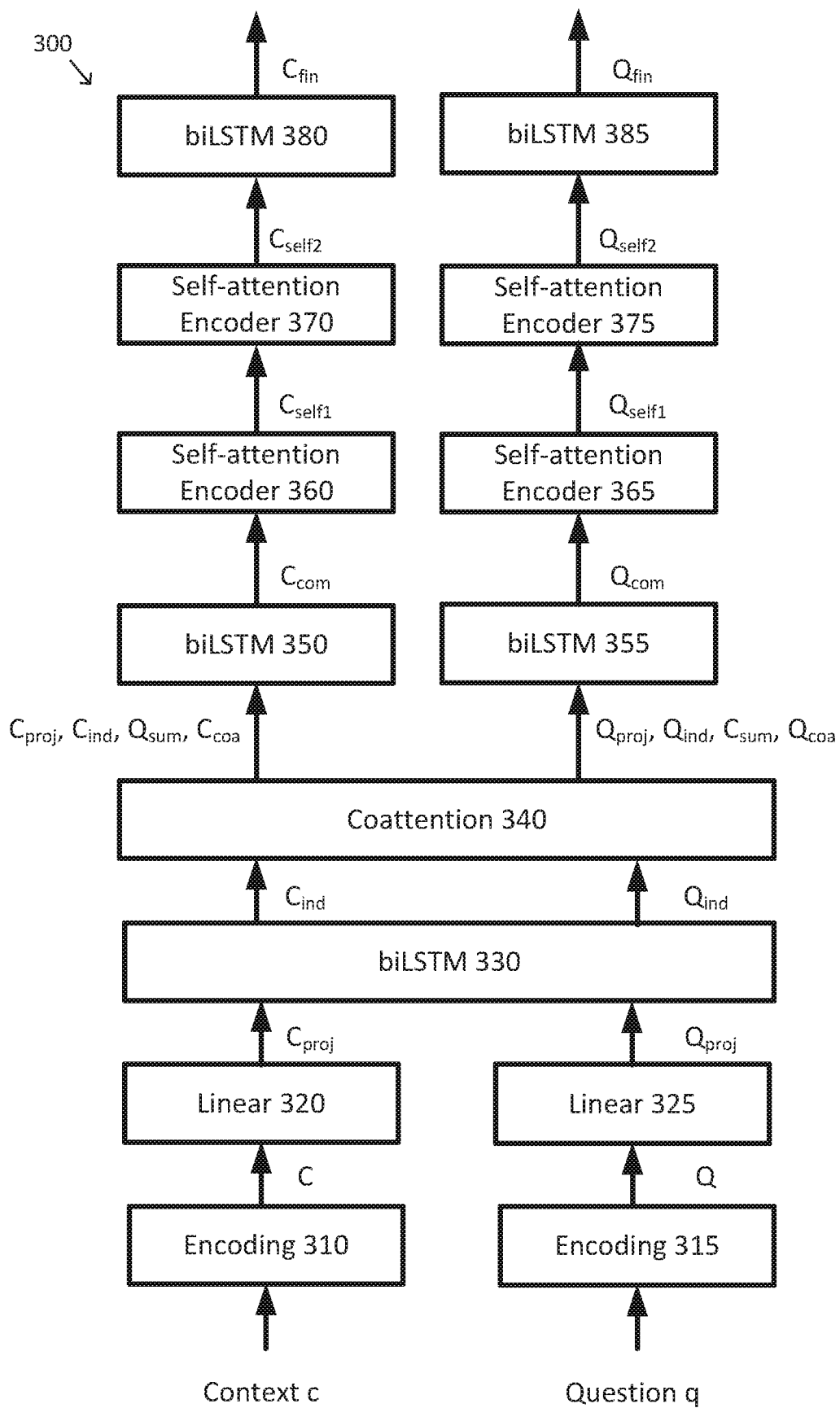


FIG. 3

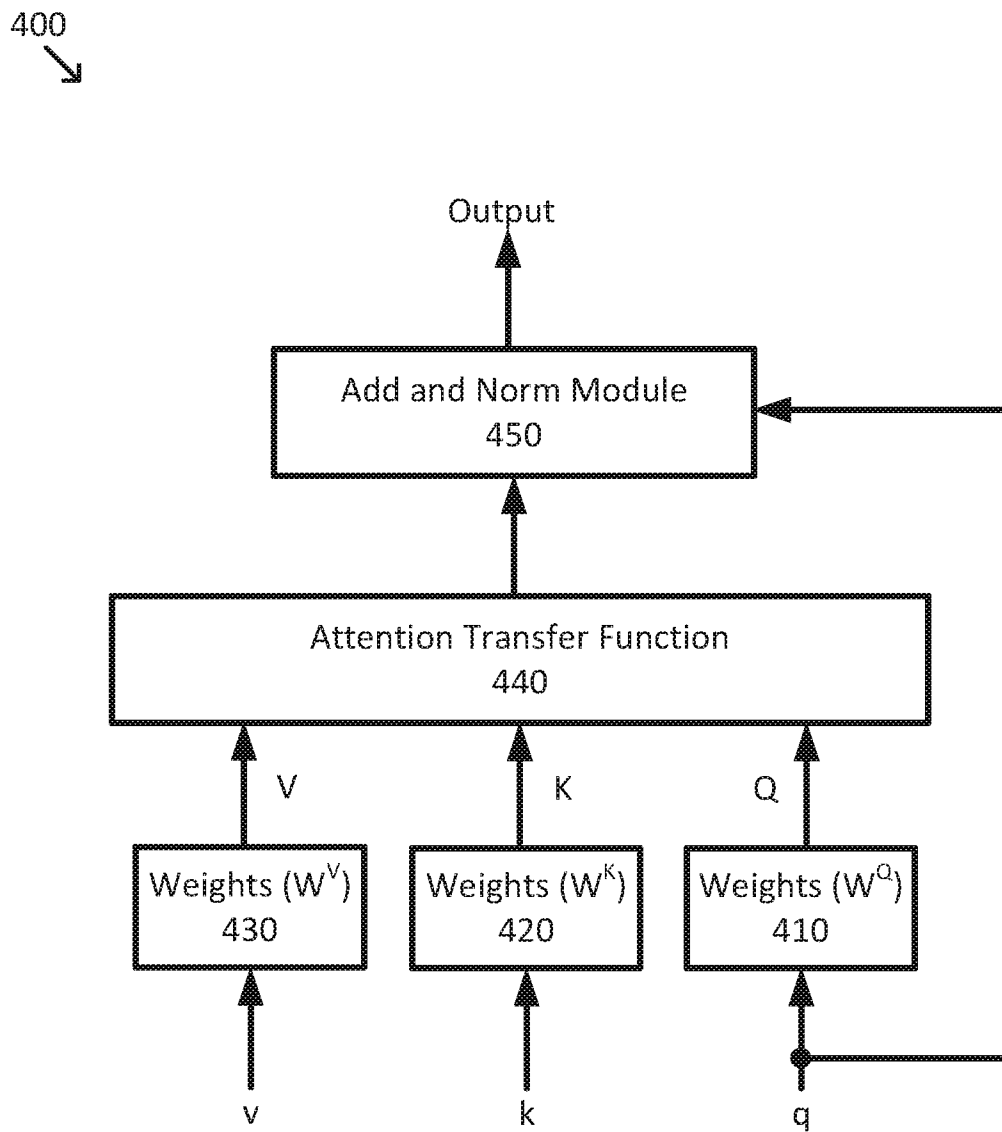


FIG. 4

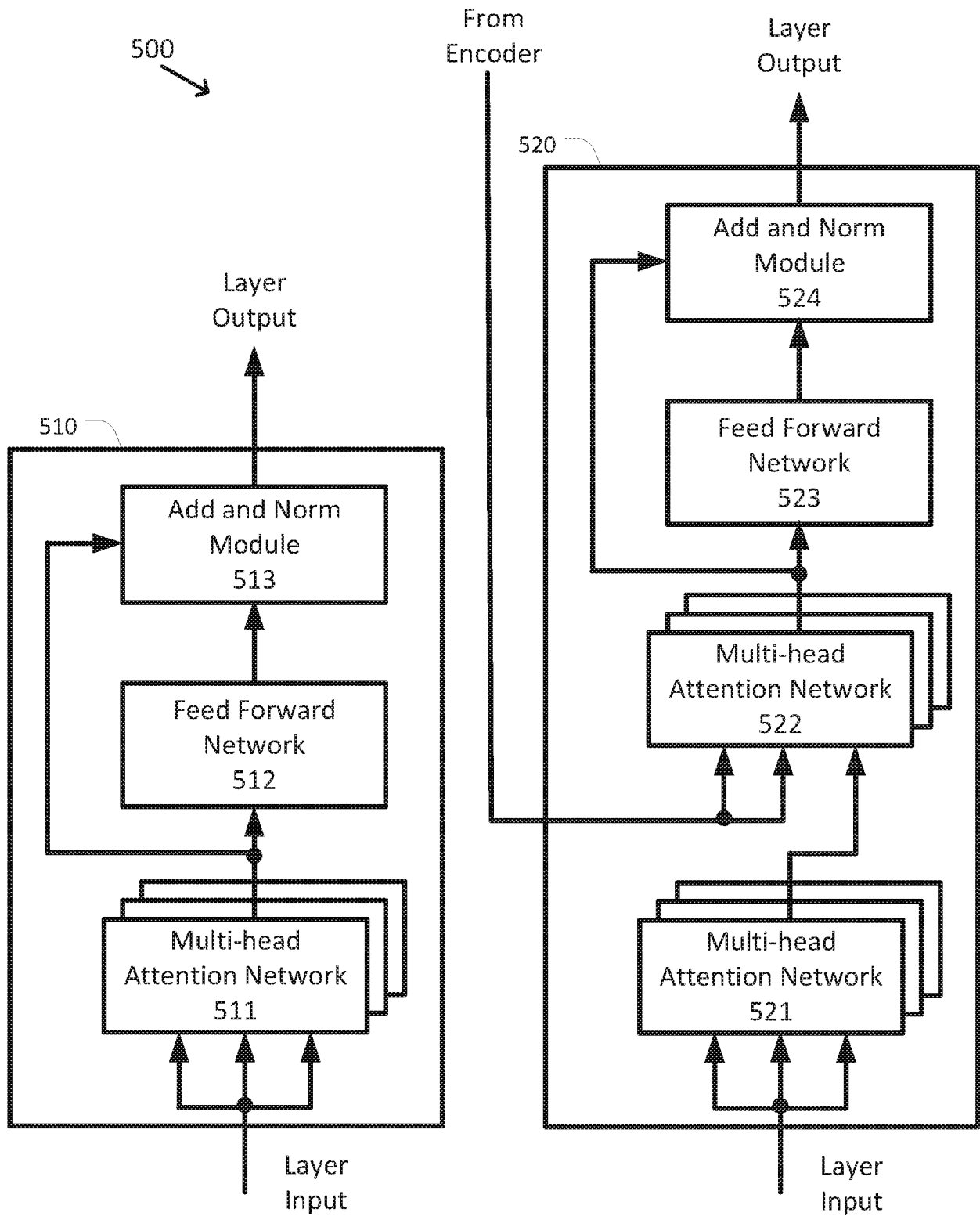


FIG. 5

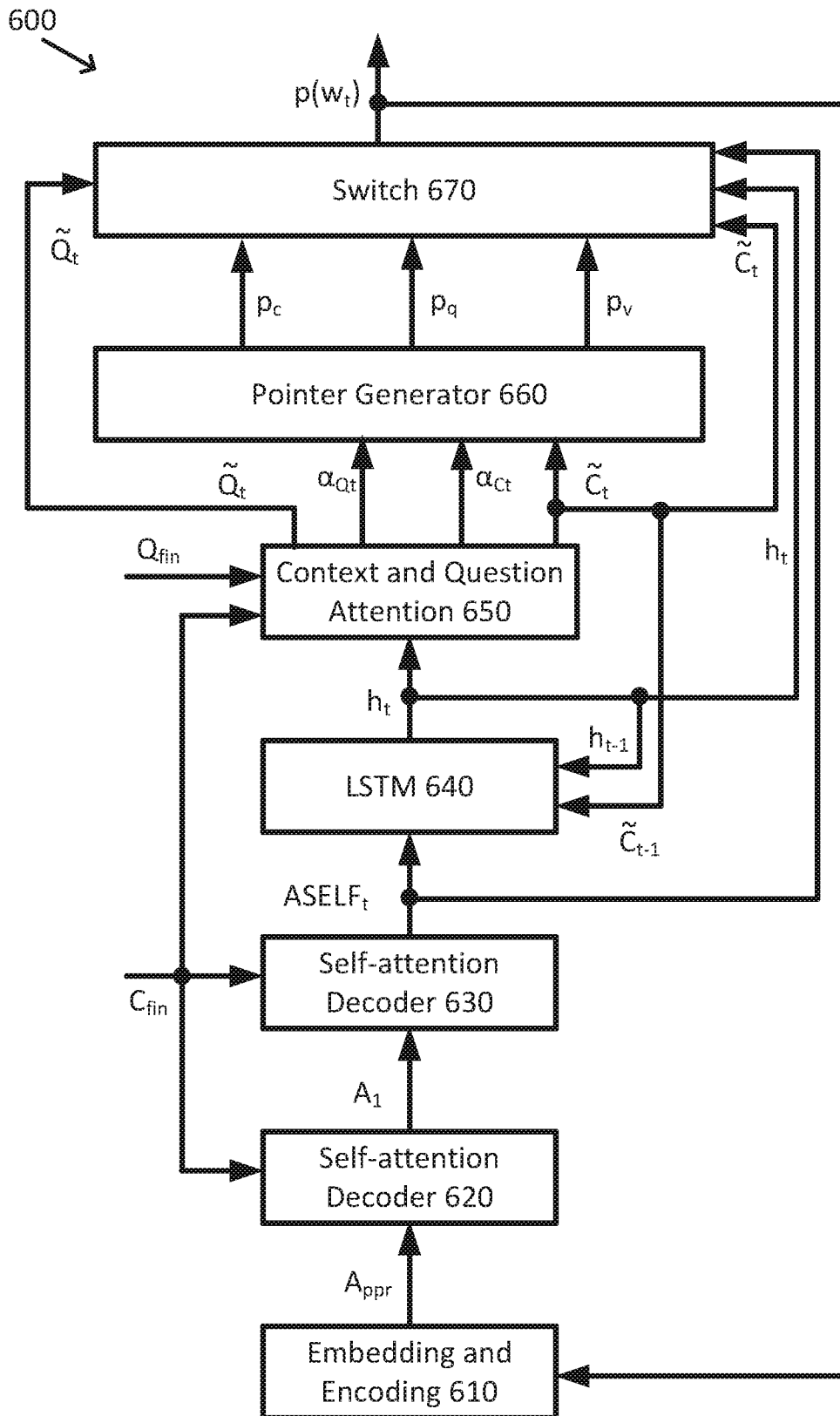


FIG. 6

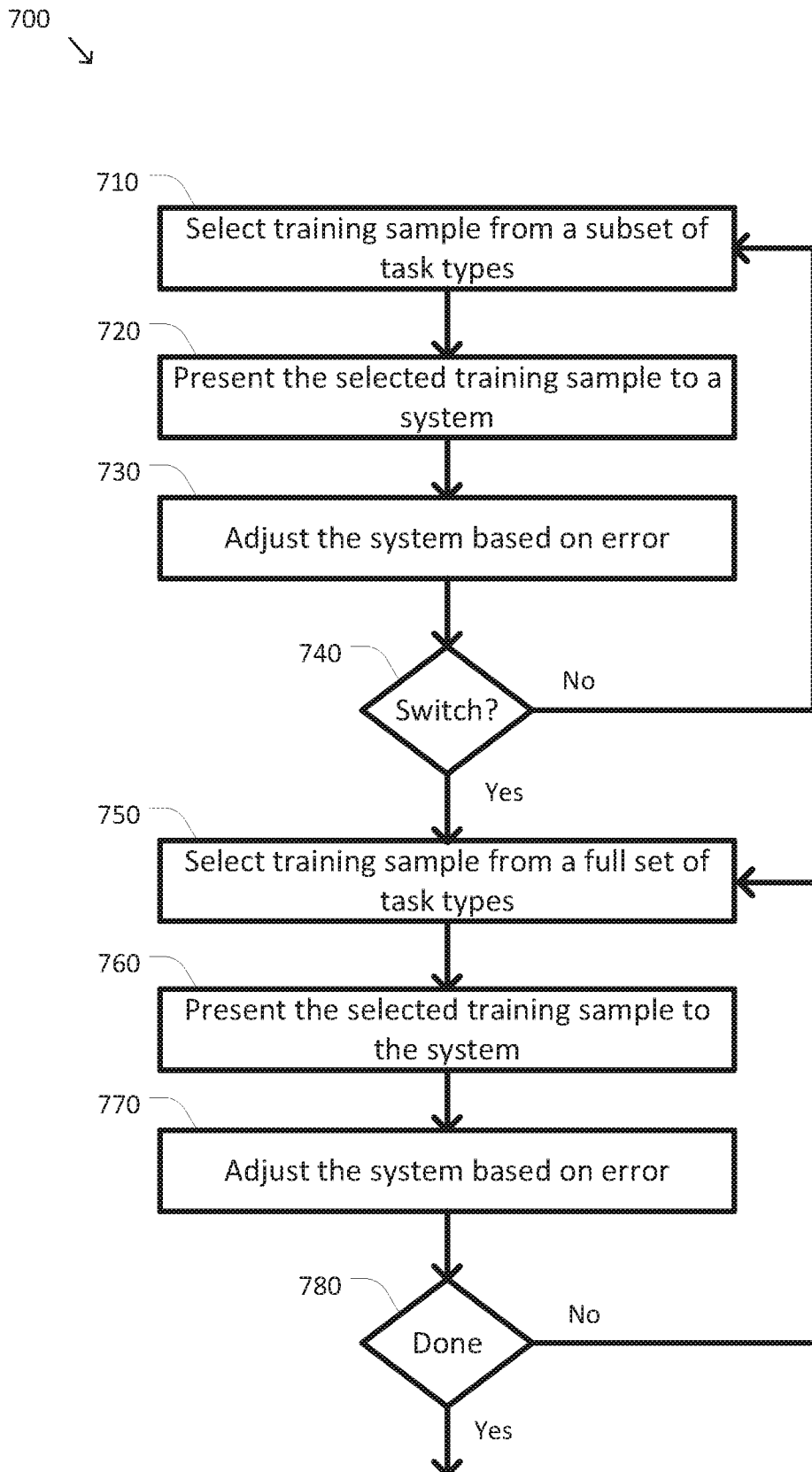


FIG. 7

Task	Dataset	# Train	# Dev	# Test	Metric
Question Answering	SQuAD	87599	10570	9616	nF1
Machine Translation	IWSLT	196884	993	1305	BLEU
Summarization	CNN/DM	287227	13368	11490	ROUGE
Natural Language Inference	MNLI	392702	20000	20000	nF1
Sentiment Analysis	SST	6920	872	1821	nF1
Semantic Role Labeling	QA-SRL	6414	2183	2201	nF1
Zero-Shot Relation Extraction	QA-ZRE	840000	600	12000	F1
Goal-Oriented Dialogue	WOZ	2536	830	1646	EM
Database Query Generation	WikiSQL	56355	8421	15878	EM
Pronoun Resolution	MWSC	80	82	100	nF1

FIG. 8

Dataset	Fully Joint	Curriculum	Anti-Curriculum		
			SQuAD	+IWSLT+CNN/DM	+MNLI
SQuAD	70.8	43.4	74.3	74.5	74.6
IWSLT	16.1	4.3	13.7	18.7	19.0
CNN/DM	23.9	21.3	24.6	20.8	21.6
MNLI	70.5	58.9	69.2	69.6	72.7
SST	86.2	84.5	86.4	83.6	86.8
QA-SRL	75.8	70.6	77.6	77.5	75.1
QA-ZRE	28.0	24.6	34.7	30.1	37.7
WOZ	80.6	81.9	84.1	81.7	85.6
WikiSQL	62.0	68.6	58.7	54.8	42.6
MWSC	48.8	41.5	48.4	34.9	41.5
decaScore	562.7	499.6	571.7	546.2	557.2

FIG. 10

Dataset	Single-task Training				Multitask Training			
	S2S	w/SAtt	+CAtt	+QPtr	S2S	w/SAtt	+CAtt	+QPtr
SQuAD	48.2	68.2	74.6	75.5	47.5	66.8	71.8	70.8
IWSLT	25.0	23.3	26.0	25.5	14.2	13.6	9.0	16.1
CNN/DM	19.0	20.0	25.1	24.0	25.7	14.0	15.7	23.9
MNLI	67.5	68.5	34.7	72.8	60.9	69.0	70.4	70.5
SST	86.4	86.8	86.2	88.1	85.9	84.7	86.5	86.2
QA-SRL	63.5	67.8	74.8	75.2	68.7	75.1	76.1	75.8
QA-ZRE	20.0	19.9	16.6	15.6	28.5	31.7	28.5	28.0
WOZ	85.3	86.0	86.5	84.4	84.0	82.8	75.1	80.6
WikiSQL	60.0	72.4	72.3	72.6	45.8	64.8	62.9	62.0
MWSC	43.9	46.3	40.4	52.4	52.4	43.9	37.8	48.8
decaScore	-	-	-	-	473.6	546.4	533.8	562.7
								571.7

FIG. 9

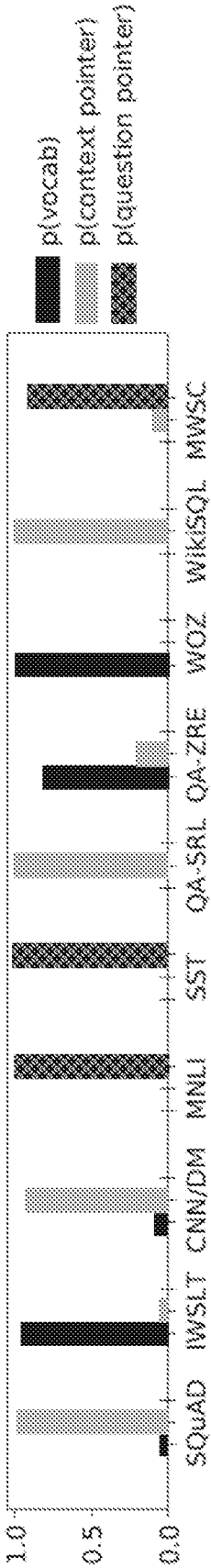


FIG. 11

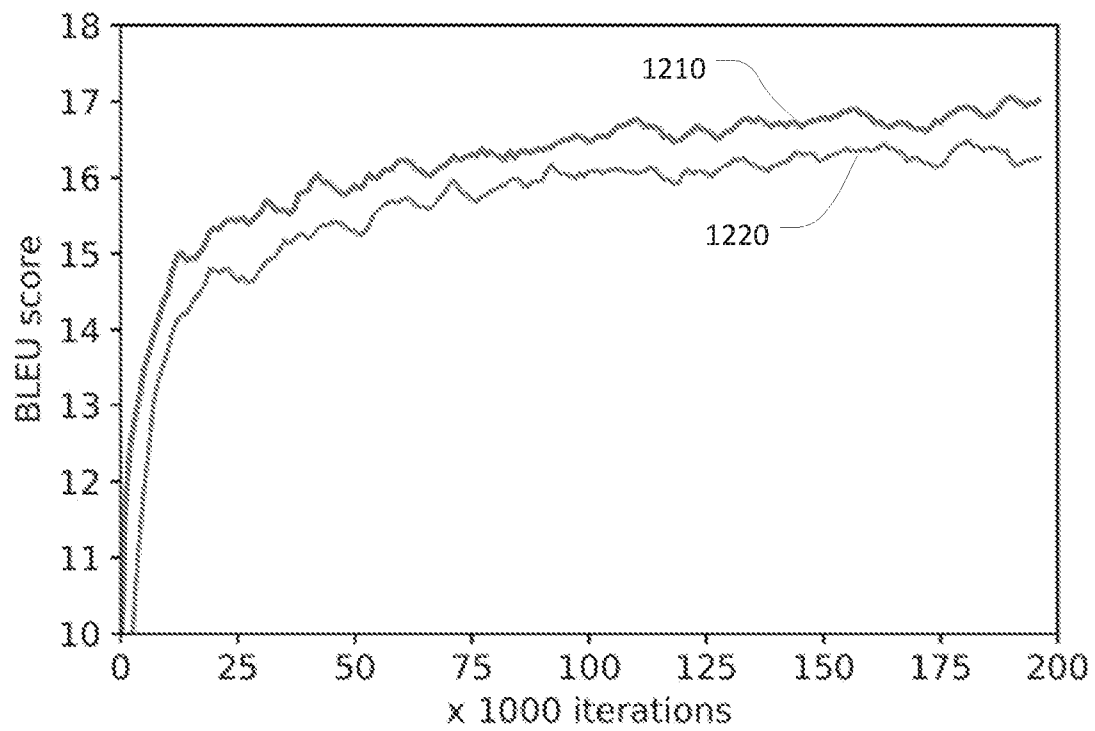


FIG. 12A

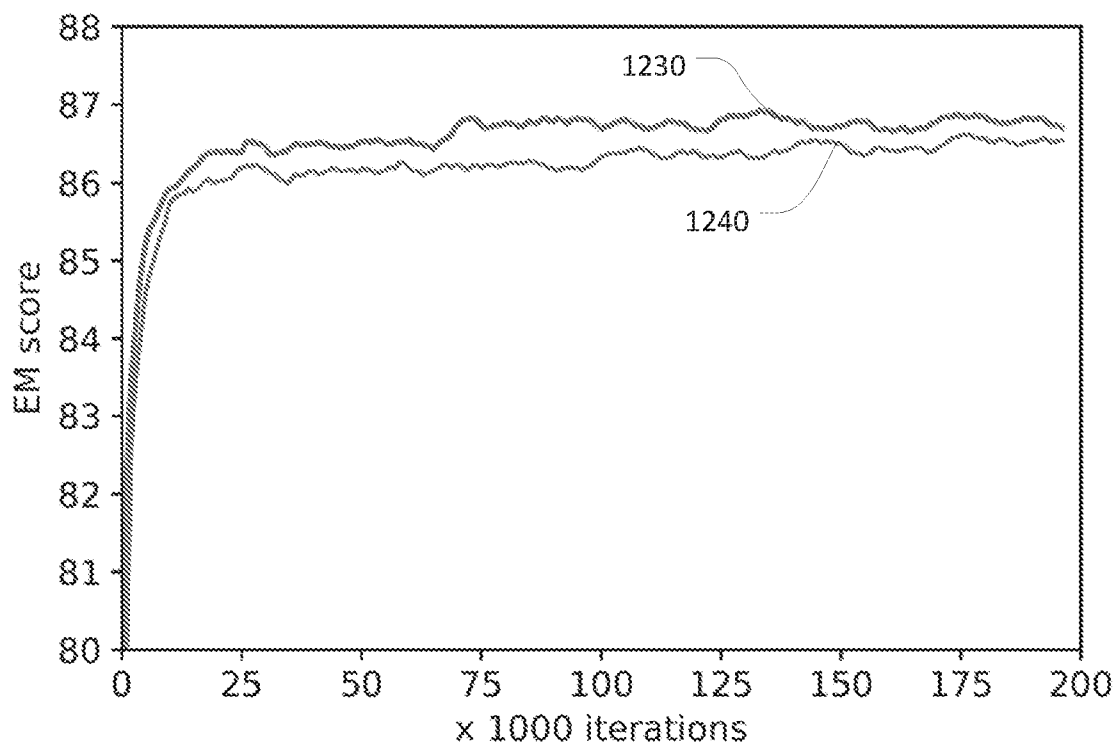


FIG. 12B

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2019/032207

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F17/27 G06N3/04
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 G06N

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, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	SHIZHU HE ET AL: "Generating Natural Answers by Incorporating Copying and Retrieving Mechanisms in Sequence-to-Sequence Learning", PROCEEDINGS OF THE 55TH ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS (VOLUME 1: LONG PAPERS), 30 July 2017 (2017-07-30), pages 199-208, XP055609663, Stroudsburg, PA, USA DOI: 10.18653/v1/P17-1019 sections 1-3; figures 1, 2 ----- -/--	1-20



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" document which may throw doubts on priority claim(s) or which 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

30 July 2019

Date of mailing of the international search report

05/08/2019

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Domingo Vecchioni, M

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2019/032207

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	Rajarshee Mitra: "An abstractive approach to question answering", arxiv.org, arXiv:1711.06238v1 [cs.CL] 16 Nov 2017, 16 November 2017 (2017-11-16), XP055609635, Retrieved from the Internet: URL:https://arxiv.org/pdf/1711.06238v1.pdf [retrieved on 2019-07-30] sections 1-3; figure 1	1-20
A	----- LINFENG SONG ET AL: "A Unified Query-based Generative Model for Question Generation and Question Answering", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 4 September 2017 (2017-09-04), XP080818294, section "Model", in particular "LSTM Decoder"; figure 1	1-20
A	----- Caiming Xiong ET AL: "DCN+: Mixed Objective and Deep Residual Coattention for Question Answering", arxiv.org, arXiv:1711.00106v2 [cs.CL] 10 Nov 2017 , 10 November 2017 (2017-11-10), XP055541783, Retrieved from the Internet: URL:https://arxiv.org/pdf/1711.00106v2.pdf [retrieved on 2019-01-14] sections 1, 2; figure 1	1-20
A	----- W0 2018/085728 A1 (SALESFORCE COM INC [US]) 11 May 2018 (2018-05-11) abstract; figure 1A	1-20
A	----- PENTINA ANASTASIA ET AL: "Curriculum learning of multiple tasks", 2015 IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR), IEEE, 7 June 2015 (2015-06-07), pages 5492-5500, XP032794019, DOI: 10.1109/CVPR.2015.7299188 [retrieved on 2015-10-14] abstract sections 1-3	7-9, 16-18
	----- -/--	

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2019/032207

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X,P	BRYAN MCCANN ET AL: "The Natural Language Decathlon: Multitask Learning as Question Answering", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 20 June 2018 (2018-06-20), XP080893587, the whole document -----	1-20
X,P	CAO LIU ET AL: "Curriculum Learning for Natural Answer Generation", PROCEEDINGS OF THE TWENTY-SEVENTH INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 13 July 2018 (2018-07-13), pages 4223-4229, XP055609675, California DOI: 10.24963/ijcai.2018/587 ISBN: 978-0-9992411-2-7 sections 1, 2 -----	1-20
A,P	FEI SUN ET AL: "Multi-Source Pointer Network for Product Title Summarization", INFORMATION AND KNOWLEDGE MANAGEMENT, ACM, 2 PENN PLAZA, SUITE 701NEW YORKNY10121-0701USA, 17 October 2018 (2018-10-17), pages 7-16, XP058420201, DOI: 10.1145/3269206.3271722 ISBN: 978-1-4503-6014-2 section 2.2 -----	1-20

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2019/032207

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2018085728	A1	11-05-2018	CA 3039386 A1 11-05-2018
			CA 3039517 A1 11-05-2018
			CA 3039551 A1 11-05-2018
			CN 109923557 A 21-06-2019
			US 2018121787 A1 03-05-2018
			US 2018121788 A1 03-05-2018
			US 2018121799 A1 03-05-2018
			WO 2018085728 A1 11-05-2018
			WO 2018085729 A1 11-05-2018
			WO 2018085730 A1 11-05-2018
