



(51) International Patent Classification:

G06N 3/04 (2006.01) G06N 3/08 (2006.01)

(21) International Application Number:

PCT/US2021/032840

(22) International Filing Date:

18 May 2021 (18.05.2021)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

17/005,067 27 August 2020 (27.08.2020) US

(71) Applicant: MICROSOFT TECHNOLOGY LICENSING, LLC [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).

(72) Inventors: JIAO, Jian; MICROSOFT TECHNOLOGY LICENSING, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). GONG, Yeyun; MICROSOFT TECHNOLOGY LICENSING, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). DUAN, Nan; MICROSOFT TECHNOLOGY LICENSING, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). ZHANG, Ruofei; MICROSOFT TECHNOLOGY LICENSING, LLC, One Microsoft Way, Redmond, Wash-

ington 98052-6399 (US). ZHOU, Ming; MICROSOFT TECHNOLOGY LICENSING, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US).

(74) Agent: SWAIN, Cassandra T. et al.; MICROSOFT TECHNOLOGY LICENSING, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (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, IT, 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, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, 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,

(54) Title: TRANSFORMER-BASED NEURAL NETWORK INCLUDING A MASK ATTENTION NETWORK

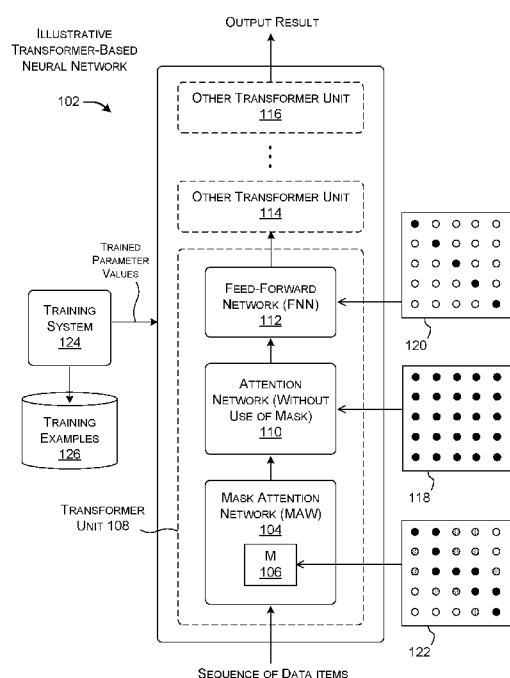


FIG. 1

(57) Abstract: A transformer-based neural network includes at least one mask attention network (MAN). The MAN computes an original attention data structure that expresses influence between pairs of data items in a sequence of data items. The MAN then modifies the original data structure by mask values in a mask data structure, to produce a modified attention data structure. Compared to the original attention data structure, the modified attention data structure better accounts for the influence of neighboring data items in the sequence of data items, given a particular data item under consideration. The mask data structure used by the MAN can have static and/or machine-trained mask values. In one implementation, the transformer-based neural network includes at least one MAN in combination with at least one other attention network that does not use a mask data structure, and at least one feed-forward neural network.

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))*
- *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))*

TRANSFORMER-BASED NEURAL NETWORK INCLUDING A MASK ATTENTION NETWORK

BACKGROUND

5 **[0001]** In recent years, both industry and academic environments have expressed significant interest in a type of neural network referred to as a transformer. A traditional transformer includes two types of neural networks operating in series: a self-attention network (SAN) followed by a feed-forward neural network (FNN). The SAN generates an output result that depends on the influence between pairs of data items in an input
10 sequence of data items. The FNN is a fully connected neural network that operates on an output result provided by the SAN. Unlike the SAN, the FNN does not incorporate a self-attention mechanism that takes into account the relations among data items.

[0002] The academic literature has proposed numerous refinements to the above-described traditional transformer architecture, e.g., by proposing modified SANs which
15 are designed to replace the traditional SAN in the transformer architecture. These solutions may increase the accuracy of the transformer, but they may do so by increasing its complexity. These improvements may therefore negatively affect the efficiency of the transformer, both with respect to its training and application.

SUMMARY

20 **[0003]** Described herein is a transformer-based neural network that includes at least one mask attention network (MAN). The MAN is configured to compute an original attention data structure that expresses influence between pairs of data items in a sequence of data items that is fed to the MAN. The MAN then modifies the original data structure by mask values in a mask data structure, to produce a modified attention data structure.
25 Compared to the original attention data structure, the modified attention data structure better accounts for the influence of neighboring data items in the sequence of data items, with respect to a particular data item under consideration. For example, the MAN can boost the relevance of data items within a local neighborhood of data items, with respect to the particular data item under consideration.

30 **[0004]** According to another illustrative aspect, the transformer-based neural network includes the MAN in combination with at least one attention network that does not use a mask data structure (referred to herein as a “mask-less attention network”), and at least one feed-forward neural network. For instance, the transformer-based neural network can provide a pipeline of data processing mechanisms that includes the MAN followed by a

mask-less attention network, followed by a feed-forward neural network.

[0005] According to another illustrative aspect, the transformer-based neural network can include a MAN that employs a mask data structure having static mask values.

5 [0006] According to another illustrative aspect, the transformer-based neural network can include a MAN that uses a mask data structure having mask values that are produced based on machine-trained parameter values. The parameter values used by the MAN can also vary depending on the particular level at which the MAN is deployed in the transformer-based neural network, and the particular attention head under consideration (to be described in greater detail below).

10 [0007] According to another illustrative aspect, the mask-less attention network and the feed-forward neural network can themselves be considered as two species of MANs having static mask data structures. That is, the mask-less attention network can be conceptualized as a MAN having a matrix of mask values that are all set to “1.” The feed-forward neural network can be conceptualized as a MAN having an identity matrix of
15 mask values (in which the diagonal of the matrix includes mask values set to “1,” and the remainder of the mask values set to zero).

[0008] According to one technical advantage, the transformer-based neural network provides more accurate results compared to a traditional transformer-based neural network. It accomplishes this objective by more effectively determining the influence
20 between pairs of data items within local neighborhoods of data items. Further, the transformer-based neural network achieves its increased accuracy without markedly increasing the number of machine-trained parameter values used by the transformer-based neural network. This characteristic allows the transformer-based neural network to be trained and applied in a resource-efficient manner.

25 [0009] The above-summarized technique can be manifested in various types of systems, devices, components, methods, computer-readable storage media, data structures, graphical user interface presentations, articles of manufacture, and so on.

[0010] This Summary is provided to introduce a selection of concepts in a simplified form; these concepts are further described below in the Detailed Description. This
30 Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used to limit the scope of the claimed subject matter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] Fig. 1 shows an overview of a transformer-based neural network that includes a

mask attention network (MAN).

[0012] Fig. 2 shows a sequence of data items (here, tokens) that is used to describe the operation of the transformer-based neural network of Fig. 1.

5 [0013] Fig. 3 shows computing equipment that can be used to implement the transformer-based neural network of Fig. 1

[0014] Fig. 4 shows one implementation of the transformer-based neural network of Fig. 1 that includes an encoder.

[0015] Fig. 5 shows one implementation of the transformer-based neural network of Fig. 1 that includes an encoder and a decoder.

10 [0016] Fig. 6 shows one implementation of part of the MAN of Fig. 1.

[0017] Figs. 7-9 show additional functionality that may be incorporated into the transformer-based neural network of Fig. 1.

[0018] Fig. 10 shows a process that provides an overview of one manner of operation of the transformer-based neural network of Fig. 1.

15 [0019] Fig. 11 shows a process for building a mask data structure for use by the MAN of Fig. 1.

[0020] Fig. 12 shows a process that represents the operation of a particular type of transformer-based neural network that includes a particular pipeline of processing mechanisms.

20 [0021] Fig. 13 shows an illustrative type of computing device that can be used to implement any aspect of the features shown in the foregoing drawings.

[0022] The same numbers are used throughout the disclosure and figures to reference like components and features. Series 100 numbers refer to features originally found in Fig. 1, series 200 numbers refer to features originally found in Fig. 2, series 300 numbers refer to features originally found in Fig. 3, and so on.

DETAILED DESCRIPTION

[0023] This disclosure is organized as follows. Section A describes an improved transformer-based neural network. Section B sets forth illustrative methods which explain the operation of the transformer-based neural network of Section A. And Section C describes illustrative computing functionality that can be used to implement any aspect of the features described in Sections A and B.

30 [0024] As a preliminary matter, the term “hardware logic circuitry” corresponds to technology that includes one or more hardware processors (e.g., CPUs, GPUs, etc.) that execute machine-readable instructions stored in a memory, and/or one or more other

hardware logic units (e.g., FPGAs) that perform operations using a task-specific collection of fixed and/or programmable logic gates. Section C provides additional information regarding one implementation of the hardware logic circuitry. In some contexts, each of the terms “component,” “module,” “engine,” “system,” and “tool” refers to a part of the hardware logic circuitry that performs a particular function or combination of functions.

[0025] In one case, the illustrated separation of various parts in the figures into distinct units may reflect the use of corresponding distinct physical and tangible parts in an actual implementation. Alternatively, or in addition, any single part illustrated in the figures may be implemented by plural actual physical parts. Alternatively, or in addition, the depiction of any two or more separate parts in the figures may reflect different functions performed by a single actual physical part.

[0026] Other figures describe the concepts in flowchart form. In this form, certain operations are described as constituting distinct blocks performed in a certain order. Such implementations are illustrative and non-limiting. Certain blocks described herein can be grouped together and performed in a single operation, certain blocks can be broken apart into plural component blocks, and certain blocks can be performed in an order that differs from that which is illustrated herein (including a parallel manner of performing the blocks). In one implementation, the blocks shown in the flowcharts that pertain to processing-related functions can be implemented by the hardware logic circuitry described in Section C, which, in turn, can be implemented by one or more hardware processors and/or other logic units that include a task-specific collection of logic gates.

[0027] As to terminology, the phrase “configured to” encompasses various physical and tangible mechanisms for performing an identified operation. The mechanisms can be configured to perform an operation using the hardware logic circuitry of Section C. The term “logic” likewise encompasses various physical and tangible mechanisms for performing a task. For instance, each processing-related operation illustrated in the flowcharts corresponds to a logic component for performing that operation. A logic component can perform its operation using the hardware logic circuitry of Section C. When implemented by computing equipment, a logic component represents an electrical element that is a physical part of the computing system, in whatever manner implemented.

[0028] Any of the storage resources described herein, or any combination of the storage resources, may be regarded as a computer-readable medium. In many cases, a computer-readable medium represents some form of physical and tangible entity. The term computer-readable medium also encompasses propagated signals, e.g., transmitted or

received via a physical conduit and/or air or other wireless medium, etc. However, the specific term “computer-readable storage medium” expressly excludes propagated signals per se in transit, while including all other forms of computer-readable media.

[0029] The following explanation may identify one or more features as “optional.”

5 This type of statement is not to be interpreted as an exhaustive indication of features that may be considered optional; that is, other features can be considered as optional, although not explicitly identified in the text. Further, any description of a single entity is not intended to preclude the use of plural such entities; similarly, a description of plural entities is not intended to preclude the use of a single entity. Further, while the description
10 may explain certain features as alternative ways of carrying out identified functions or implementing identified mechanisms, the features can also be combined together in any combination. Further, the term “plurality” refers to two or more items, and does not necessarily imply “all” items of a particular kind, unless otherwise explicitly specified. Further still, the descriptors “first,” “second,” “third,” etc. are used to distinguish among
15 different items, and do not imply an ordering among items, unless otherwise noted. Finally, the terms “exemplary” or “illustrative” refer to one implementation among potentially many implementations.

A. Illustrative Computing Systems

[0030] Fig. 1 shows an overview of a transformer-based neural network 102 that
20 includes a mask attention network (MAN) 104. The transformer-based neural network 102 transforms a sequence of data items into an output result. In most of the examples presented herein, the sequence of input data items is a sequence of linguistic tokens, such as a sequence of words or n-grams that make up one or more input sentences. But the transformer-based neural network 102 can be applied to other kinds of input information.
25 For example, the transformer-based neural network 102 can operate on a sequence of audio features extracted from audio content, or a sequence of image features extracted from image or video content, and so on.

[0031] The transformer-based neural network 102 produces different kinds of output results, depending on the application context in which it is applied. In one case, an
30 application relies on an encoder (not shown) to map a sequence of input items into a latent semantic vector and/or classification result. That encoder, in turn, incorporates at least one instance of the transformer-based neural network 102 shown in Fig. 1. The latent semantic vector and/or classification result expresses at least one characteristic of the sequence of input items. The application can use the semantic vector or classification

result in different ways. To cite one example, a search application provided by a search engine can use the encoder to map a query submitted by a user into a first semantic vector. The search application can use the same encoder to map a candidate linguistic item (such as a document, ad keyword, etc.) to a second latent semantic vector. The search application can then determine the relevance of the query to the candidate linguistic item by computing the distance between the first and second latent semantic vectors, e.g., using cosine similarity or some other distance metric. After repeating this task for plural candidate linguistic items, the search application can notify the user of the most relevant candidate item(s), or perform some other task based on the most relevant candidate item(s).

[0032] In another case, an application can rely on both an encoder and a decoder (not shown) to map the sequence of input items from an input form into an output form. Each of the encoder and the decoder, in turn, incorporates at least one instance of the transformer-based neural network 102 shown in Fig. 1. For instance, a machine-translation application can use the encoder to map a sentence in a first natural language (such as English) into an intermediary form, and then use the decoder to map the intermediary form to a sentence in a second natural language (such as French). A document summarization application can use the encoder and decoder to map an original document into a summary document which summarizes the otherwise rephrases the original document, and so on.

[0033] In another case, a training system can use an encoder and/or a decoder to produce a pre-trained model. A downstream training system can then use the pre-trained model as a starting point in generating an application-specific model. The encoder and/or the decoder can each incorporate at least one instance of the transformer-based neural network 102 shown in Fig. 1.

[0034] The above applications are mentioned here by way of example, not limitation. Other application environments can use the transformer-based neural network 102 for other respective purposes.

[0035] From a high-level standpoint, the MAN 104 is configured to generate an original attention data structure that identifies the influence between pairs of data items in the sequence of data items. The MAN then uses a mask data structure 106 to modify the original attention data structure, to produce a modified attention data structure. The modified attention data structure adjusts the original attention data structure to specifically take account for the influence among neighboring data items in the sequence of data items,

with respect to a particular data item of interest.

[0036] Jumping ahead momentarily in the figures, Fig. 2 provides an example of the above-described concepts. Assume that the sequence of input items expresses an input sentence 202 that reads, “I will be driving to Billings, Montana in my new Ford Ranger with my wife and youngest son.” The MAN 104 generates an original attention data structure that describes the influence between different pairs of words in that sentence. For example, consider the word “Billings” as a focus of interest, which is intended to refer to a city in the U.S. State of Montana. The original attention data structure includes a first original attention value that describes how much the word “driving” contributes to understanding the meaning of the word “Billings,” a second original attention value that describes how much the word “Montana” contributes to understanding the meaning of the word “Billings,” and so on. In this regard, the original attention data structure provides a global expression of the interrelations of words in the input sentence 202. But the original attention data structure may also capture attention information that is not particularly relevant and potentially misleading. For example, the original attention data structure includes an original attention data value that reflects how much the word “son” contributes to understanding the word “Billings.” The word “son,” however, can be expected to contribute very little to understanding the meaning of “Billings,” as it may not be common to use both the words “Billings” and “son” in a single sentence.

[0037] The MAN 104 addresses the above issue by using the mask data structure 106 to effectively diminish the importance of some pairings of words in the input sentence 202, and to bolster other pairings of words. For instance, again consider the case in which the word “Billings” is the focal point of interest at the moment. The mask data structure 106 defines a mask value for each pairing in the sentence 202, including a first mask value for the pair of “Billings” and “driving,” a second mask value for the pair of “Billings” and “Montana,” etc. The MAN 104 can then multiply each such mask value by a corresponding entry in the original attention data structure. For example, the MAN 104 can multiply the mask value for the combination of “Billings” and “driving” by the original attention value for that pairing of words in the original attention data structure, to produce a modified attention value. In many cases, the mask values can be selected in such a manner to boost words in the sentence 202 that are relatively near the word “Billings,” and diminish the influence of words that are relatively far from the word “Billings.” For example, the mask values can be chosen to enhance the influence of words in a neighborhood 204 of words that is centered on the word “Billings,” and diminish the

influence of words lying outside the neighborhood 204. In this case, the MAN 104 can apply a mask value to the combination of “Billings” and “son” which will block the contribution of this pairing, or otherwise diminish its importance.

[0038] As will be clarified in greater detail below, different implementations can rely on different kinds of mask data structures. In a first case, the mask data structure includes a static set of mask values. For example, again consider the case in which the word “Billings” is the focal point of interest. The mask data structure 206 can assign a mask value having the value “1” to any pairing of words that both appear in the neighborhood 204. The mask data structure 206 can assign the mask value “0” to any pairing that includes a word outside the neighborhood 204 (in combination with “Billings” as a first member of the pair). In a second case, the mask data structure 106 contains mask values that are produced based on machine-trained parameter values.

[0039] Mathematically expressed, the MAN 104 can produce its output result based on the following illustrative and non-limiting equation:

$$Attention(Q, K, V) = \frac{M_{i,j} \exp(Q_i K_j^T / \sqrt{d_k})}{\sum_k M_{i,k} \exp(Q_i K_k^T / \sqrt{d_k})} V \quad (1).$$

The symbols Q , K , and V refer respectively to a query matrix, key matrix, and value matrix that are fed to the MAN 104. For an encoder (for example), the transformer-based neural network 102 produces these input matrices by formulating a matrix X that expresses the sequence of input items as different vectors within the matrix X . It then linearly projects this matrix X into three different matrices (Q , K , V) using three respective machine trained matrices. That is, the transformer-based neural network 102 multiplies the matrix X by a first machine-trained matrix (W_Q) to produce the query matrix Q , multiplies the matrix X by a second machine-trained matrix (W_K) to produce the key matrix K , and multiplies the matrix X by a third machine-trained matrix (W_V) to produce the value matrix V . The symbol d_k refers to the dimensionality of the key matrix K . The symbol M refers to the mask data structure 106. (Note that the matrices Q , K , V , can have other interpretations in other application contexts, compared to the interpretation provided above for an encoder.)

[0040] Using the terminology set forth above, the matrix $\exp(QK^T / \sqrt{d_k})$ is one example of the original attention data structure, where QK^T refers to the dot product of the matrix Q by the transpose of the matrix K . That is, since the vectors in Q and K originate from individual words in the input sequence of data items, the matrix product QK^T

includes original attribute values that express the influence between pairs of data items in the sequence of data items. The pointwise multiplication $M \exp(QK^T / \sqrt{d_k})$ is one example of the modified attention data structure. This pointwise product is produced by multiplying the mask data structure M by the original attention data structure in an element-by-element manner.

[0041] Returning to Fig. 1, the transformer-based neural network 102 can optionally include a pipeline of data processing mechanisms that includes the MAN 104 as one member. For instance, the transformer-based neural network 102 can include at least one transformer unit 108 that includes a pipeline of data processing mechanisms that includes the MAN 104, another attention network 110 that does not make use of a mask data structure, and a feed-forward neural network (FNN) 112. Assume that the MAN 104 maps the sequence of input data items into a first output result. The other attention network 110 transforms the first output result into a second output result. And the FNN 112 maps the second output result into a third output result.

[0042] The other attention network 110 operates in the same manner as the MAN 104, with the exception that it does not build and apply a mask data structure. For this reason, the other attention network 110 is referred to herein as a “mask-less attention network.” This simply means that the other attention network 110 does not apply a mask data structure having the same constitution and purpose as the MAN 104 (and is not meant to exclude the possibility that the other attention network 110 may apply a mask data structure of some other kind in some other context).

[0043] More specifically, the mask-less attention network 110 operates using Equation (1), with the exception that the mask data structure M is removed. More succinctly stated, the mask-less attention network 110 operates by applying the equation:

$$Attention(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2).$$

[0044] The FNN 112 uses a fully-connected neural network. In one implementation, the FNN 112 transforms hidden state information (h_i) for each data item that is fed to it using two successive stages, as defined by the following illustrative and non-limiting equation:

$$FNN(h_i) = \text{ReLU}(h_i W_1 + b_1) W_2 + b_2 \quad (3).$$

Here, W_1 and W_2 are matrices of machine-trained parameter values, and b_1 and b_2 are optional machine-trained bias values. ReLU refers to a rectified linear unit activation

function.

[0045] The above-described arrangement of the MAN 104, mask-less attention network 110, and the FNN 112 is merely illustrative. In other implementations, a transformation unit can arrange these three processing mechanisms in a different order than described above. In addition, or alternatively, another implementation of a transformation unit can include two or more of any of the above-described types of processing mechanisms, e.g., by including two MANs. In addition, or alternatively, another implementation of a transformation unit can omit one or more of the above-described types of processing mechanisms, e.g., by omitting the mask-less attention network 110. In addition, or alternatively, another implementation of a transformation unit can add one or more other types of processing mechanisms, that is, besides the three types of processing mechanisms described above.

[0046] Further note that the transformer-based neural network 102 as a whole can include any number of transformer units stacked together, such as transformer unit 114 and transformer unit 116. Each such transformer unit occurs at a particular level (L) of in the transformer-based neural network 102. Each such transformer unit can include the same three processing mechanisms as the transformer unit 108, or represent a variation thereof. For example, each transformer unit in the transformer-based neural network 102 can include its own MAN as a lead component in its pipeline. However, as will be clarified below, each MAN at each level of the transformer-based neural network 102 can use level-specific parameter values, and can include a level-specific mask data structure.

[0047] On another level of abstraction, the MAN 104, mask-less attention network 110, and FNN 112 can be mathematically reformulated as three respective variants of a generically-expressed mask attention network. Each version of the generically-expressed mask attention network includes a different mask data structure M and applies a different activation function $\mathcal{F}$ to the results generated by Equation (1), a point which can be mathematically expressed as $\mathcal{F}(\text{Attention}(Q, K, V))$.

[0048] For example, the mask-less attention network 110 applies a mask data structure M having mask values that are all set to “1.” A mask data structure 118 represents this type of mask information, which includes black-centered dots that represent mask values set to the value “1.” The mask-less attention network 110 applies an activation function $\mathcal{F}$ that implements an identity function. With these settings, the generically-expressed mask attention network degenerates into Equation (2). Also note that the absence of a mask data

structure enables the mask-less attention network 110 to globally consider the impact of each data item on each other data item in the sequence of data items.

[0049] The FNN 112 applies a mask data structure M that adopts the identity matrix. A mask data structure 120 represents this type of mask information, which includes black-centered dots along its diagonal that represent mask values that are set to the value of “1,” and white-centered dots that represent mask values that are set to the value of “0.” The activation function $\mathcal{F}$ in this case is ReLU. With these settings, the generically-expressed mask attention network degenerates into Equation (3). Note that the FNN 112 refines the sequence of input items that are fed to it, but, unlike the mask-less attention network 110, does not consider the impact of any data item on any other data item.

[0050] Finally, in one implementation, the MAN 104 applies a mask data structure that includes machine-trained mask values. A mask data structure 122 represents this type of mask information. The dots having centers with differing degrees of shading represent mask values having differing machine-trained values, ranging from 0 to 1.0. The MAN 104 can be viewed as sharing some characteristics with both the mask-less attention network 110 and the FNN 112. It is like the mask-less attention network 110 in that it considers the impact between at least some pairs of data items in the sequence of data items. It is like the FNN 112 in that it also restricts the interaction among pairs of data items, at least to some extent. In other words, the MAN 104 can be viewed as operationally lying between the MAN 104 and the FNN 112.

[0051] A training system 124 produces parameter values used by the transformer-based neural network 102, including the various weighting matrices, bias values, and mask values specified in the above-described equations. The training system 124 performs this task by operating on a data store 126 of training examples. In one merely illustrative case, the training examples can include pairs of sentences. The first member of each pair expresses content in a first natural language, and the second member of the pair expresses the same content in a second natural language. The training system 124 can operate on this training set to minimize or maximize some training objective expressed by a cost function, e.g., by maximizing the production of correct machine translations. The training system 124 can use any training technique to iteratively achieve the above result, such as stochastic gradient descent.

[0052] According to one technical advantage, the transformer-based neural network 102 provides more accurate results compared to a traditional transformer-based neural network. It accomplishes this objective by more effectively determining the influence

between pairs of data items within local neighborhoods of data items. This can be seen in the example of Fig. 2, where the use of the mask data structure 106 may diminish significance of far-flung pairings of words in a sentence, such as “Billings” and “son.” Further, the transformer-based neural network 102 achieves its increased accuracy without markedly increasing the number of machine-trained parameter values used by the transformer-based neural network 102. For instance, Equation (1) incorporates the use of a new mask data structure 106, M , compared to Equation (2). The new parameter values required to build M grows with the number of words in an input sequence, the number of levels to be considered in the transformer-based neural network, and the number of heads to be considered in each MAN. This increase in the number of parameters is relatively modest compared to other solutions that require modifications to the manner in which the original attention data structure is computed. This characteristic allows the transformer-based neural network 102 to be trained and applied in a resource-efficient manner compared to other networks that adopt solutions that require an exponential increase in the number of parameter values.

[0053] Fig. 3 shows computing equipment that can be used to implement any aspects of the transformer-based neural network 102 of Fig. 1. The computing equipment includes a set of user computing devices 302 coupled to a set of servers 304 via a computer network 306. Each user computing device can correspond to any device that performs a computing function, including a desktop computing device, a laptop computing device, a handheld computing device of any type (e.g., a smartphone, a tablet-type computing device, etc.), a mixed reality device, a wearable computing device, an Internet-of-Things (IoT) device, a gaming system, and so on. The computer network 306 can be implemented as a local area network, a wide area network (e.g., the Internet), one or more point-to-point links, or any combination thereof.

[0054] Fig. 3 also indicates that one or more applications 308 can be spread across the user computing devices 302 and/or the servers 304 in any manner. Each such application can include at least one transformer-based neural network 102. In one case, an application is entirely implemented by one or more of the servers 304. Here, a user can interact with the application via a user computing device, e.g., using a browser application that runs on the user computing device. In another case, an application is entirely implemented by a user computing device in local fashion, in which case no interaction with the servers 304 is necessary. In another case, the functionality associated with an application is distributed between the servers 304 and each user computing device.

[0055] Fig. 4 shows an encoder 402 that relies on at least one transformer-based neural network (TNN) 404 of the type described in Fig. 1 to map a sequence of input items into a latent semantic vector and/or a classification result.

[0056] Fig. 5 shows an encoder 502 and a decoder 504 that transform a sequence of input values from one form to another. The encoder 502 includes at least one TNN 506, while the decoder 504 uses at least one TNN 508. In some implementations, the decoder 504 can provide at least one encoder-decoder attention network (not shown) that receives its K and V matrices from an output of the encoder 502, and its Q matrix from an underlying attention network.

[0057] Fig. 6 shows one implementation of the MAN 104 of Fig. 1. The MAN 104 includes a mechanism 602 for generating an original attention data structure 604. In the context of Equation (1) above, the mechanism 602 performs this task, in part, by multiplying the Q matrix by the transpose of the K matrix. In the case of an encoder, the vectors in the Q and K matrices represent different encodings of the sequence of input items (e.g., linguistic tokens).

[0058] A mask-building mechanism 606 dynamically generates the mask data structure 106. In one implementation, the mask-building mechanism 606 generates each mask value in the mask data structure for a given pair of data items (w_t, w_s), at positions t and s in the sequence, respectively, based on the following equation:

$$M_{t,s,head,level} = \sigma(h_t^{level} W^{level} + P_{t-s}^{level} + U_{head}^{level}) \quad (4).$$

[0059] For example, assume that the word w_t in the example of Fig. 2 is “Billings,” and the word w_s is “driving.” The symbols t and s refer to the positions of these two words in the sentence 202. The symbol $level$ in Equation (4) refers to a particular level in the transformer-based neural network 102 in which the mask data structure M is used. As will be clarified below, the MAN 104 can also apply two or more heads which perform the same calculation described in Equation (1), but with respect to different head-specific machine-trained parameter values; in that context, the symbol $head$ refers to a particular head in the MAN 104. The symbol h_t^{level} refers to a hidden state that represents the word w_t in the sequence of data items at the particular $level$. For example, h_t^{level} may correspond to an encoding, expressed as a vector, that represents the word “Billings” in Fig. 2. W^{level} represents a machine-trained matrix to be multiplied by h_t^{level} . The difference $t - s$ represents a separation between the words w_t and w_s in the sequence of input items. For instance, if w_t and w_s are separated by three words, then $t - s$ will be 3.

The symbol P_{t-s}^{level} refers to a machine-trained parameter value associated with the difference $t - s$ and the particular *level*. And U_{head}^{level} refers to a machine-trained parameter value associated with the particular *head* and the particular *level* under consideration. σ refers to an activation function applied to the sum shown in Equation (4),
 5 in this case, the sigmoid function.

[0060] Fig. 6 shows that the mask-building mechanism 606 draws machine-trained parameters values from one or more weight matrices 608. For example, a first weighting matrix (or a set of plural matrices) can store a set of machine-trained values for P_{t-s}^{level} , for different values of *level*, and for different values of $t - s$, such as
 10 $\{-3, -2, -1, 0, +1, +2, +3\}$. Similarly, a second weight matrix (or a set of plural matrices) can store a set of machine-trained values for U_{head}^{level} for different values of *level*, and for different values of *head*. The mask-building mechanism 606 selects values from these matrices depending on the particular values of *level*, $t - s$, and *head* under consideration. The mask-building mechanism 606 can also optionally draw from a weight
 15 matrix (or a set of plural matrices) that includes precomputed values of $h_t^{level} W^{level}$. This is possible for the case in which there is a predetermined vocabulary of words w_t (and associated hidden states h_t^{level}), and a predetermined set of matrices for W^{level} . Note that the mask value given by Equation (2) depends on the position t of w_t relative to the position s of w_s , and the hidden state of w_t (which is h_t), but does not depend on the
 20 hidden state of w_s . This means that the same mask value can be used for a word w_s that occurs n words before w_t , regardless of what that word may be.

[0061] Note that the mask-building mechanism 606 can update the map data structure 106 based on the receipt of a new sequence of input items having new data items (e.g., new tokens). This is because the first term of Equation (4) depends on the encoding h_t .
 25 And for this reason, the MAN 104 may be considered dynamic in nature. As also explained above, the MAN 104 provides different map data structures for different levels and attention heads.

[0062] A modifying mechanism 610 next produces a modified attention data structure 612 based on the mask data structure 106. It does this by pointwise (element-by-element)
 30 multiplying the original attention data structure 604 by the mask data structure 106, e.g., as expressed by the numerator of Equation (1). Although not represented in Fig. 6, the MAN 104 also multiplies the modified attention data structure 612 by the value matrix V , as specified by Equation (1).

[0063] In another implementation, the mask-building mechanism 606 produces a mask data structure that includes static mask values, without relying on any machine-trained parameter values. For example, the mask-building mechanism 606 can set an original attention value to zero if a pair of data items (t, s) under consideration are separated by more than a prescribed number of data items (such as five data items in one merely illustrative case). It can otherwise set an original attention value to “1.” In another implementation, the mask-building mechanism 606 produces a mask data structure that includes a combination of static mask values and mask values produced based on machine-trained parameter values.

[0064] Finally, Fig. 6 shows another arrow pointing to the modifying mechanism 610 that is given the label “other sources.” This label indicates that the modifying mechanism 610 can receive additional information from external sources, and leverage those sources to further modify the original attention data structure and/or to supplement the processing performed by the transformer-based neural network 102 in other ways. The “other sources” can include one or more knowledgebases that provide knowledge regarding the relationship among terms. For example, assume that a particular input sequence includes two words that are related, but the input sequence itself does not reveal that relationship. The modifying mechanism 610 can gain this insight by consulting a knowledgebase. The modifying mechanism 610 can then enhance or diminish an original attention value based on its determination of whether or not the two words are related. For example, assume that a sentence included the words “Billings” and “Yellowjacket.” A knowledgebase may indicate that Yellowjacket is the name of the mascot of a local university in Billings, MT, and that these two words therefore have a semantic relation.

[0065] Figs. 7-9 show additional functionality that may be incorporated in the transformer-based neural network 102 of Fig. 1. Starting with Fig. 7, this figure shows preliminary processing that can be applied to a sequence of data items prior to feeding those data items into the MAN 104. First, an embedding mechanism 702 transforms the data items (e.g., linguistic tokens) into a set of input embeddings, also referred to herein as input vectors. The embedding mechanism 702 can use different techniques to perform this task. In one approach, the embedding mechanism 702 can convert each input item in an input sequence into a vector representation using a lookup table, neural network, etc.

[0066] Next, an adding mechanism 704 adds position information to each input embedding. The position information describes the position of a data item (associated with a particular input embedding) in the sequence of data items. For example, assume

that an input sequence includes words in a phrase that reads “hybrid Ford hatchback 2020.” The adding mechanism 704 will add position information to the input embedding associated with “hybrid” that indicates that the word “hybrid” is the first word in the phrase. Overall, the transformer-based neural network 102 adds position information to the input embeddings to inform its attention networks of the positional context of each data item under consideration within the sequence of data items. The adding mechanism 704 can encode position information in different ways, such as by using one or more sinusoidal functions to map an index of a data item into position information, or by using a machine-trained function to map the index of the data item into position information, etc. Overall, the adding mechanism 704 produces position-modified embeddings for application to the MAN 104 of Fig. 1.

[0067] Fig. 8 shows an add-and-normalize mechanism 802 that can be placed between the mask-less attention network 110 and the FNN 112. Although not shown, another add-and-normalize mechanism may be provided after the FNN 112. The particular add-and-normalize mechanism 802 shown in Fig. 8 adds the input that is fed to the mask-less attention network 110 to its output, to produce a sum, and then performs layer normalization on that sum. That is, if the sum is expressed as a vector v , layer normalization involves subtracting the mean μ of the elements in the vector v from the vector v , and dividing that result by the standard deviation σ of the elements in the vector v .

[0068] Fig. 9 shows that the MAN 104 of Fig. 1 can include at least three heads (902, 904, 906) that perform the same calculation shown in Equation (1), but using different machine-trained parameter values. The different heads effectively capture different insights from the sequence of input items using different representation subspaces. Fig. 1 also shows that the heads (902, 904, 906) apply different respective mask data structures (908, 910, 912). A combination mechanism 914 concatenates the output results produced by the different heads (902, 904, 906), and then multiplies this concatenation result by another machine-trained matrix W_o . The mask-less attention network 110 also uses multiple heads, but without masks.

B. Illustrative Processes

[0069] Figs. 10-12 show processes that explain the operation of the transformer-based neural network 102 of Section A in flowchart form. Since the principles underlying the operation of the transformer-based neural network 102 have already been described in

Section A, certain operations will be addressed in summary fashion in this section. As noted in the prefatory part of the Detailed Description, each flowchart is expressed as a series of operations performed in a particular order. But the order of these operations is merely representative, and can be varied in any manner.

5 [0070] To begin with, Fig. 10 shows a process 1002 that provides an overview of the operation of a transformer-based neural network 102 of Fig. 1, according to one implementation. In block 1004, an application receives a sequence of data items, based on input provided by a user computing device. The application uses the transformer-based neural network 102, which, in turn, provides a pipeline of processing mechanisms. In
10 block 1006, the transformer-based neural network 102 processes the sequence of data items to produce an output result. In block 1008, the application provides information to the user that is based on the output result.

[0071] At least one processing mechanism in the pipeline of processing mechanisms provided by the transformer-based neural network 102 includes a mask attention network
15 (MAN) 104. The MAN 104 includes a mask data structure 106. Fig. 10 also summarizes operations performed by the MAN 104. More specifically, in block 1010, the MAN 104 computes an original attention data structure 604 that expresses influence between pairs of data items in the sequence of data items. In block 1012, the MAN 104 computes a modified attention data structure 612 by modifying the original attention data structure
20 604 by mask values provided by the mask data structure 106. The modified attention data structure 612 accounts for influence of neighboring data items in the sequence of data items, with respect to a particular data item under consideration, beyond that provided by the original attention data structure 604.

[0072] Fig. 11 shows a process 1102 for building the mask data structure 106. The
25 process 1102 is specifically framed in the context of a particular pair of data items in the sequence of data items that is made up of a first data item (w_t) at a position t and a second data item (w_s) at a position s . In block 1104, the MAN 104 modifies a hidden state h_t associated with the first data item by a machine-trained weight matrix, to produce a modified state. In block 1106, the MAN 104 determines a distance $(t - s)$ between the
30 first data item w_t and the second data item w_s within the sequence of data items. In block 1108, the MAN 104 selects a machine-trained distance-related parameter value (P_{t-s}) from a set of machine-trained distance-related parameter values, based on the distance that is determined in block 1106. In block 1110, the MAN 104 selects a machine-trained head-

related parameter value (U) from a set of machine-trained head-related parameter values, based on a particular head of the MAN 104 under consideration. Although not shown, blocks 1108 and 1110 can also make their selections based on the level at which the MAN 104 exists in the transformer-based neural network 102. In block 1112, the MAN 104 generates a mask value for the particular pair of data items based at least on the modified state, the distance-related parameter value that is selected in block 1108, and the head-related parameter value that is selected in block 1110.

[0073] Fig. 12 shows a process 1202 that represents an overview of the operation of the transformer-based neural network, for the particular configuration of the transformer-based neural network 102 shown in Fig. 1. In block 1204, the transformer-based neural network 102 processes a sequence of data items using a mask attention network (MAN) 104 to produce a first output result. More specifically, in block 1206, the MAN 104 computes an original attention data structure 604 that expresses influence between pairs of data items in the sequence of data items. In block 1208, the MAN 104 computes a modified attention data structure 612 by modifying the original attention data structure 604 by mask values provided by a mask data structure 106. In block 1210, the transformer-based neural network 102 processes the first output result using another attention network 110 that does not use a mask data structure, to provide a second output result. In block 1212, the transformer-based neural network 102 processes the second output result by a feed-forward neural network (FNN) 112, to produce a third output result.

C. Representative Computing Functionality

[0074] Fig. 13 shows a computing device 1302 that can be used to implement any aspect of the mechanisms set forth in the above-described figures. For instance, with reference to Fig. 3, the type of computing device 1302 shown in Fig. 13 can be used to implement any user computing device or any server. In all cases, the computing device 1302 represents a physical and tangible processing mechanism.

[0075] The computing device 1302 can include one or more hardware processors 1304. The hardware processor(s) 1304 can include, without limitation, one or more Central Processing Units (CPUs), and/or one or more Graphics Processing Units (GPUs), and/or one or more Application Specific Integrated Circuits (ASICs), etc. More generally, any hardware processor can correspond to a general-purpose processing unit or an application-specific processor unit.

[0076] The computing device 1302 can also include computer-readable storage media

1306, corresponding to one or more computer-readable media hardware units. The computer-readable storage media 1306 retains any kind of information 1308, such as machine-readable instructions, settings, data, etc. Without limitation, for instance, the computer-readable storage media 1306 may include one or more solid-state devices, one or more magnetic hard disks, one or more optical disks, magnetic tape, and so on. Any instance of the computer-readable storage media 1306 can use any technology for storing and retrieving information. Further, any instance of the computer-readable storage media 1306 may represent a fixed or removable unit of the computing device 1302. Further, any instance of the computer-readable storage media 1306 may provide volatile or non-volatile retention of information.

[0077] The computing device 1302 can utilize any instance of the computer-readable storage media 1306 in different ways. For example, any instance of the computer-readable storage media 1306 may represent a hardware memory unit (such as Random Access Memory (RAM)) for storing transient information during execution of a program by the computing device 1302, and/or a hardware storage unit (such as a hard disk) for retaining/archiving information on a more permanent basis. In the latter case, the computing device 1302 also includes one or more drive mechanisms 1310 (such as a hard drive mechanism) for storing and retrieving information from an instance of the computer-readable storage media 1306.

[0078] The computing device 1302 may perform any of the functions described above when the hardware processor(s) 1304 carry out computer-readable instructions stored in any instance of the computer-readable storage media 1306. For instance, the computing device 1302 may carry out computer-readable instructions to perform each block of the processes described in Section B.

[0079] Alternatively, or in addition, the computing device 1302 may rely on one or more other hardware logic units 1312 to perform operations using a task-specific collection of logic gates. For instance, the hardware logic unit(s) 1312 may include a fixed configuration of hardware logic gates, e.g., that are created and set at the time of manufacture, and thereafter unalterable. Alternatively, or in addition, the other hardware logic unit(s) 1312 may include a collection of programmable hardware logic gates that can be set to perform different application-specific tasks. The latter category of devices includes, but is not limited to Programmable Array Logic Devices (PALs), Generic Array Logic Devices (GALs), Complex Programmable Logic Devices (CPLDs), Field-Programmable Gate Arrays (FPGAs), etc.

[0080] Fig. 13 generally indicates that hardware logic circuitry 1314 includes any combination of the hardware processor(s) 1304, the computer-readable storage media 1306, and/or the other hardware logic unit(s) 1312. That is, the computing device 1302 can employ any combination of the hardware processor(s) 1304 that execute machine-readable instructions provided in the computer-readable storage media 1306, and/or one or more other hardware logic unit(s) 1312 that perform operations using a fixed and/or programmable collection of hardware logic gates. More generally stated, the hardware logic circuitry 1314 corresponds to one or more hardware logic units of any type(s) that perform operations based on logic stored in and/or otherwise embodied in the hardware logic unit(s).

[0081] In some cases (e.g., in the case in which the computing device 1302 represents a user computing device), the computing device 1302 also includes an input/output interface 1316 for receiving various inputs (via input devices 1318), and for providing various outputs (via output devices 1320). Illustrative input devices include a keyboard device, a mouse input device, a touchscreen input device, a digitizing pad, one or more static image cameras, one or more video cameras, one or more depth camera systems, one or more microphones, a voice recognition mechanism, any movement detection mechanisms (e.g., accelerometers, gyroscopes, etc.), and so on. One particular output mechanism may include a display device 1322 and an associated graphical user interface presentation (GUI) 1324. The display device 1322 may correspond to a liquid crystal display device, a light-emitting diode display (LED) device, a cathode ray tube device, a projection mechanism, etc. Other output devices include a printer, one or more speakers, a haptic output mechanism, an archival mechanism (for storing output information), and so on. The computing device 1302 can also include one or more network interfaces 1326 for exchanging data with other devices via one or more communication conduits 1328. One or more communication buses 1330 communicatively couple the above-described units together.

[0082] The communication conduit(s) 1328 can be implemented in any manner, e.g., by a local area computer network, a wide area computer network (e.g., the Internet), point-to-point connections, etc., or any combination thereof. The communication conduit(s) 1328 can include any combination of hardwired links, wireless links, routers, gateway functionality, name servers, etc., governed by any protocol or combination of protocols.

[0083] Fig. 13 shows the computing device 1302 as being composed of a discrete collection of separate units. In some cases, the collection of units may correspond to

discrete hardware units provided in a computing device chassis having any form factor. Fig. 13 shows illustrative form factors in its bottom portion. In other cases, the computing device 1302 can include a hardware logic unit that integrates the functions of two or more of the units shown in Fig. 1. For instance, the computing device 1302 can include a system on a chip (SoC or SOC), corresponding to an integrated circuit that combines the functions of two or more of the units shown in Fig. 13.

[0084] The following summary provides a non-exhaustive set of illustrative examples of the technology set forth herein.

[0085] According to a first example, one or more computing devices for processing a sequence of data items is described. The computing device(s) include a transformer-based neural network, which, in turn, includes neurons implemented by hardware logic circuitry. The transformer-based neural network has a pipeline of processing mechanisms. At least one processing mechanism in the pipeline of processing mechanisms includes a mask attention network. The mask attention network includes a mask data structure. The mask attention network is configured to perform operations of: computing an original attention data structure that expresses influence between pairs of data items in the sequence of data items; and computing a modified attention data structure by modifying the original attention data structure by mask values provided by the mask data structure. The modified attention data structure accounts for influence of neighboring data items in the sequence of data items, with respect to a particular data item under consideration, beyond that provided by the original attention data structure. The transformer-based neural network is configured to generate an output result based at least in part on the modified attention data structure. The computing device(s) also includes an application implemented by the hardware logic circuitry that uses the transformer-based neural network to provide a service to at least one user.

[0086] According to a second example, the pipeline of processing mechanisms includes, in addition to the mask attention network, another attention network that does not use the mask data structure, and a feed-forward neural network.

[0087] According to a third example, relating to the second example, the masked attention network occurs prior to the other attention network in the pipeline of processing mechanisms, and wherein the other attention network occurs prior to the feed-forward neural network in the pipeline of processing mechanisms.

[0088] According to a fourth example, the transformer-based neural network includes an encoder, and wherein the mask attention network is a part of the encoder.

[0089] According to a fifth example, the transformer-based neural network includes both an encoder and a decoder, the encoder feeding an encoder output result as input into the decoder. The encoder includes at least one mask attention network and the decoder includes at least one mask attention network.

5 [0090] According to a sixth example, the mask data structure includes a set of binary-valued static parameter values.

[0091] According to a seventh example, the mask data structure is built based on parameter values produced by a machine training system.

[0092] According to an eighth example, the mask attention network is configured to
10 build the mask data structure. The building the mask attention data structure includes, for a particular pair of data items in the sequence of data items that is made up of a first data item and a second data item: modifying a hidden state associated with the first data item by a machine-trained weight matrix, to produce a modified state; determining a distance between the first data item and the second data item within the sequence of data items;
15 selecting a machine-trained distance-related parameter value from a set of machine-trained distance-related parameter values, based on the distance that is determined; and generating a mask value for the particular pair of data items based at least on the modified state and the distance-related parameter value that is selected.

[0093] According to a ninth example, relating to the eighth example, the operation of
20 generating a mask value includes summing at least the modified state and the distance-related parameter value that is selected to produce a sum, and then modifying the sum by an activation function.

[0094] According to a tenth example, relating to the eighth example, the mask
25 attention network corresponds to a particular head of a multi-headed network. The mask attention network is further configured to select a machine-trained head-related parameter value from a set of machine-trained head-related parameter values, based on the particular head. Further, the operation of generating the mask value for the particular pair is also based on the head-related parameter value that is selected.

[0095] According to an eleventh example, relating to the eighth example, the mask
30 attention network exists within a particular level of the transformer-based neural network. The distance-related parameter value that is selected is also selected based on the particular level.

[0096] According to a twelfth example, the transformer-based neural network includes an encoder and a decoder that cooperatively transform the sequence of data items from a

first form to a second form. The output result includes the data items in the second form.

[0097] According to a thirteenth example, the transformer-based neural network includes an encoder that transforms the sequence of data items into a semantic vector and/or a classification result. The output result includes the semantic vector and/or the classification result.

[0098] According to a fourteenth example, a computer-implemented method for processing a sequence of data items is described. The method includes: receiving, by an application, the sequence of data items based on input provided by a user via a user computing device, the application providing a transformer-based neural network having a pipeline of processing mechanisms; processing the sequence of data items using the transformer-based neural network to produce an output result; and providing information to the user, by the application, that is based on the output result. At least one processing mechanism in the pipeline of processing mechanisms includes a mask attention network. The mask attention network includes a mask data structure. The mask attention network performs operations of: computing an original attention data structure that expresses influence between pairs of data items in the sequence of data items; and computing a modified attention data structure by modifying the original attention data structure by mask values provided by the mask data structure. The modified attention data structure accounts for influence of neighboring data items in the sequence of data items, with respect to a particular data item under consideration, beyond that provided by the original attention data structure.

[0099] According to a fifteenth example, relating to the fourteenth example, the pipeline of processing mechanisms includes, in addition to the mask attention network, another attention network that does not use the mask data structure, and a feed-forward neural network.

[00100] According to a sixteenth example, relating to the fifteenth example, the masked attention network occurs prior to the other attention network in the pipeline of processing mechanisms, and wherein the other attention network occurs prior to the feed-forward neural network in the pipeline of processing mechanisms.

[00101] According to a seventeenth example, relating to the fourteenth example, the method further includes building the mask data structure. The operation of building of the mask data structure includes, for a particular pair of data items in the sequence of data items that is made up of a first data item and a second data item: modifying a hidden state associated with the first data item by a machine-trained weight matrix, to produce a

modified state; determining a distance between the first data item and the second data item within the sequence of data items; selecting a machine-trained distance-related parameter value from a set of machine-trained distance-related parameter values, based on the distance that is determined and based on a particular level in which the mask attention network exists in the transformer-based neural network; and generating a mask value for the particular pair of data items based at least on the modified state and the distance-related parameter value that is selected.

[00102] According to an eighteenth aspect, relating to the seventeenth aspect, the mask attention network corresponds to a particular head of a multi-headed network. The mask attention network is further configured to select a machine-trained head-related parameter value from a set of machine-trained head-related parameter values, based on the particular head and based on the particular level. Further, the operation of generating the mask value for the particular pair is also based on the head-related parameter value that is selected.

[00103] According to a nineteenth aspect, a computer-readable storage medium for storing computer-readable instructions is described. The computer-readable instructions, when executed by one or more hardware processors, perform a method that includes processing a sequence of data items using a mask attention network to produce a first output result. More specifically, the mask attention network performs operations of: computing an original attention data structure that expresses influence between pairs of data items in the sequence of data items; and computing a modified attention data structure by modifying the original attention data structure by mask values provided by a mask data structure. The method also includes: processing the first output result using another attention network that does not use a mask data structure, to provide a second output result; and processing the second output result by a feed-forward neural network to produce a third output result. The mask attention network, the other attention network, and the feed-forward neural network are also implemented by the computer-readable instructions provided by the computer-readable storage medium.

[00104] According to a twentieth aspect, relating to the nineteenth aspect, the method further includes building the mask data structure, wherein the operation of building of the mask data structure includes, for a particular pair of data items in the sequence of data items that is made up of a first data item and a second data item: modifying a hidden state associated with the first data item by a machine-trained weight matrix, to produce a modified state; determining a distance between the first data item and the second data item within the sequence of data items; selecting a machine-trained distance-related parameter

value from a set of machine-trained distance-related parameter values, based on the distance that is determined; and generating a mask value for the particular pair of data items based at least on the modified state and the distance-related parameter value that is selected.

5 **[00105]** A twenty-first aspect corresponds to any combination (e.g., any logically consistent permutation or subset) of the above-referenced first through twentieth examples.

10 **[00106]** A twenty-second aspect corresponds to any method counterpart, device counterpart, system counterpart, means-plus-function counterpart, computer-readable storage medium counterpart, data structure counterpart, article of manufacture counterpart, graphical user interface presentation counterpart, etc. associated with the first through twenty-first examples.

15 **[00107]** In closing, the description may have set forth various concepts in the context of illustrative challenges or problems. This manner of explanation is not intended to suggest that others have appreciated and/or articulated the challenges or problems in the manner specified herein. Further, this manner of explanation is not intended to suggest that the subject matter recited in the claims is limited to solving the identified challenges or problems; that is, the subject matter in the claims may be applied in the context of challenges or problems other than those described herein.

20 **[00108]** Although the subject matter has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the specific features or acts described above. Rather, the specific features and acts described above are disclosed as example forms of implementing the claims.

CLAIMS

1. One or more computing devices for processing a sequence of data items, comprising:

a transformer-based neural network including neurons implemented by hardware logic circuitry,

the transformer-based neural network having a pipeline of processing mechanisms, at least one processing mechanism in the pipeline of processing mechanisms including a mask attention network,

the mask attention network including a mask data structure,

the mask attention network being configured to perform operations of:

computing an original attention data structure that expresses influence between pairs of data items in the sequence of data items; and

computing a modified attention data structure by modifying the original attention data structure by mask values provided by the mask data structure,

the modified attention data structure accounting for influence of neighboring data items in the sequence of data items, with respect to a particular data item under consideration, beyond that provided by the original attention data structure,

the transformer-based neural network being configured to generate an output result based at least in part on the modified attention data structure,

said one or more computing devices also including an application implemented by the hardware logic circuitry that uses the transformer-based neural network to provide a service to at least one user.

2. The one or more computing devices of claim 1, wherein the pipeline of processing mechanisms includes, in addition to the mask attention network, another attention network that does not use the mask data structure, and a feed-forward neural network.

3. The one or more computing devices of claim 2, wherein the masked attention network occurs prior to the other attention network in the pipeline of processing mechanisms, and wherein the other attention network occurs prior to the feed-forward neural network in the pipeline of processing mechanisms.

4. The one or more computing devices of claim 1, wherein the mask data structure includes a set of binary-valued static parameter values.

5. The one or more computing devices of claim 1, wherein the mask data structure is built based on parameter values produced by a machine training system.

6. The one or more computing devices of claim 1,
wherein the mask attention network is configured to build the mask data structure,
wherein building the mask attention data structure includes, for a particular pair of data items in the sequence of data items that is made up of a first data item and a second data item:

modifying a hidden state associated with the first data item by a machine-trained weight matrix, to produce a modified state;

determining a distance between the first data item and the second data item within the sequence of data items;

selecting a machine-trained distance-related parameter value from a set of machine-trained distance-related parameter values, based on the distance that is determined; and

generating a mask value for the particular pair of data items based at least on the modified state and the distance-related parameter value that is selected.

7. The one or more computing devices of claim 6, wherein said generating a mask value includes summing at least the modified state and the distance-related value that is selected to produce a sum, and then modifying the sum by an activation function.

8. The one or computing device of claim 6,
wherein the mask attention network corresponds to a particular head of a multi-headed network,

wherein the mask attention network is further configured to select a machine-trained head-related parameter value from a set of machine-trained head-related parameter values, based on the particular head, and

wherein said generating the mask value for the particular pair is also based on the head-related parameter value that is selected.

9. A computer-implemented method for processing a sequence of data items, comprising:

receiving, by an application, the sequence of data items based on input provided by a user via a user computing device, the application providing a transformer-based neural network having a pipeline of processing mechanisms;

processing the sequence of data items using the transformer-based neural network

to produce an output result; and

providing information to the user, by the application, that is based on the output result,

at least one processing mechanism in the pipeline of processing mechanisms including a mask attention network,

the mask attention network including a mask data structure,

the mask attention network performing operations of:

computing an original attention data structure that expresses influence between pairs of data items in the sequence of data items; and

computing a modified attention data structure by modifying the original attention data structure by mask values provided by the mask data structure,

the modified attention data structure accounting for influence of neighboring data items in the sequence of data items, with respect to a particular data item under consideration, beyond that provided by the original attention data structure.

10. A computer-readable storage medium for storing computer-readable instructions, the computer-readable instructions, when executed by one or more hardware processors, performing a method that comprises:

processing a sequence of data items using a mask attention network to produce a first output result,

the mask attention network performing operations of:

computing an original attention data structure that expresses influence between pairs of data items in the sequence of data items; and

computing a modified attention data structure by modifying the original attention data structure by mask values provided by a mask data structure;

processing the first output result using another attention network that does not use a mask data structure, to provide a second output result; and

processing the second output result by a feed-forward neural network to produce a third output result,

the mask attention network, the other attention network, and the feed-forward neural network also being implemented by the computer-readable instructions provided by the computer-readable storage medium.

11. The one or more computing devices of claim 1, wherein the transformer-based neural network includes an encoder, and wherein the mask attention network is a part of the encoder.

12. The one or more computing devices of claim 1,
wherein the transformer-based neural network includes both an encoder and a decoder, the encoder feeding an encoder output result as input into the decoder, and
wherein the encoder includes at least one mask attention network and the decoder includes at least one mask attention network.

13. The one or more computing devices of claim 6,
wherein the mask attention network exists within a particular level of the transformer-based neural network, and
wherein the distance-related parameter value that is selected is also selected based on the particular level.

14. The one or more computing devices of claim 1,
wherein the transformer-based neural network includes an encoder and a decoder that cooperatively transform the sequence of data items from a first form to a second form, and
wherein the output result includes the data items in the second form.

15. The one or more computing devices of claim 1,
wherein the transformer-based neural network includes an encoder that transforms the sequence of data items into a semantic vector and/or a classification result, and
wherein the output result includes the semantic vector and/or the classification result.

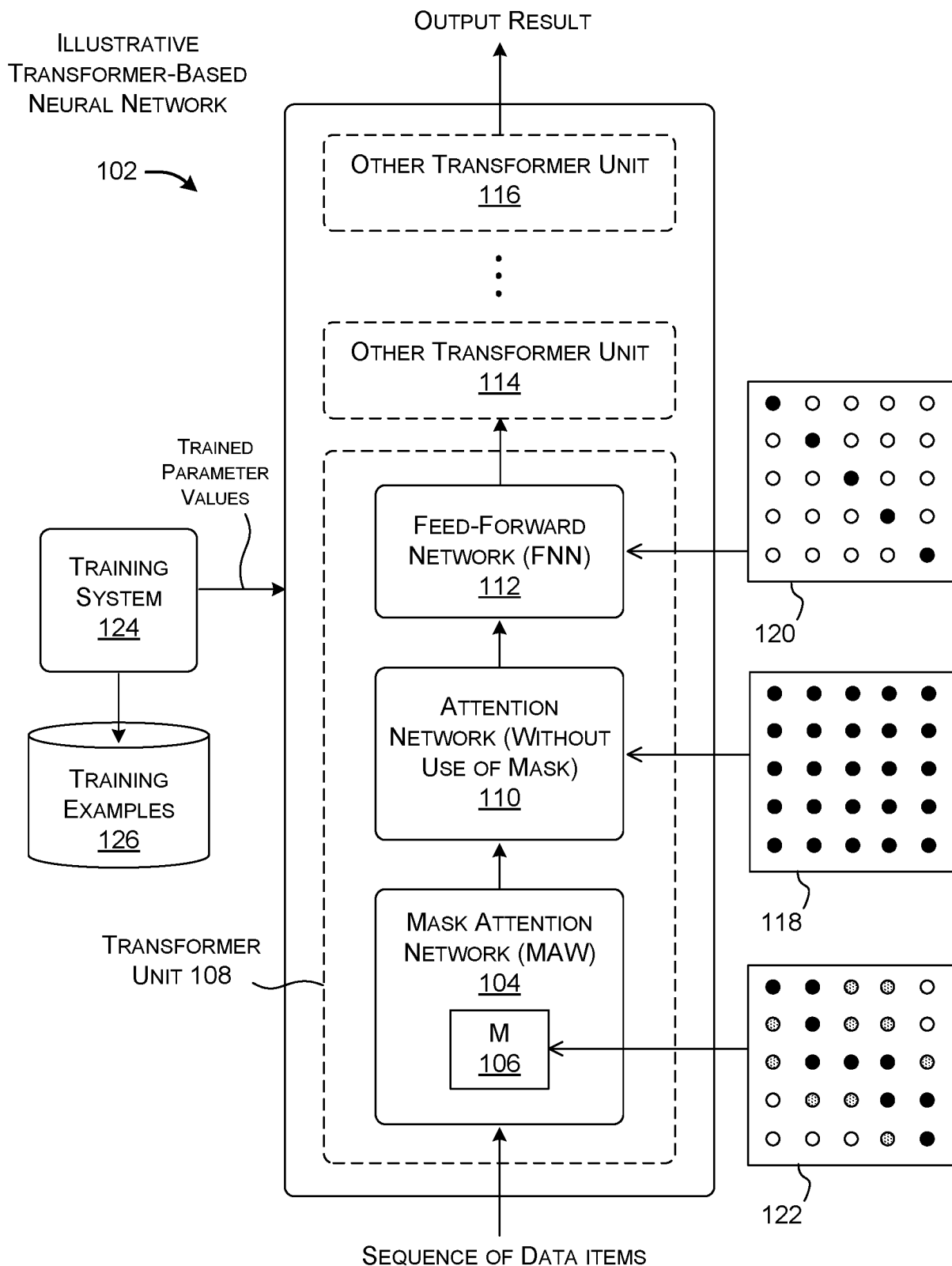
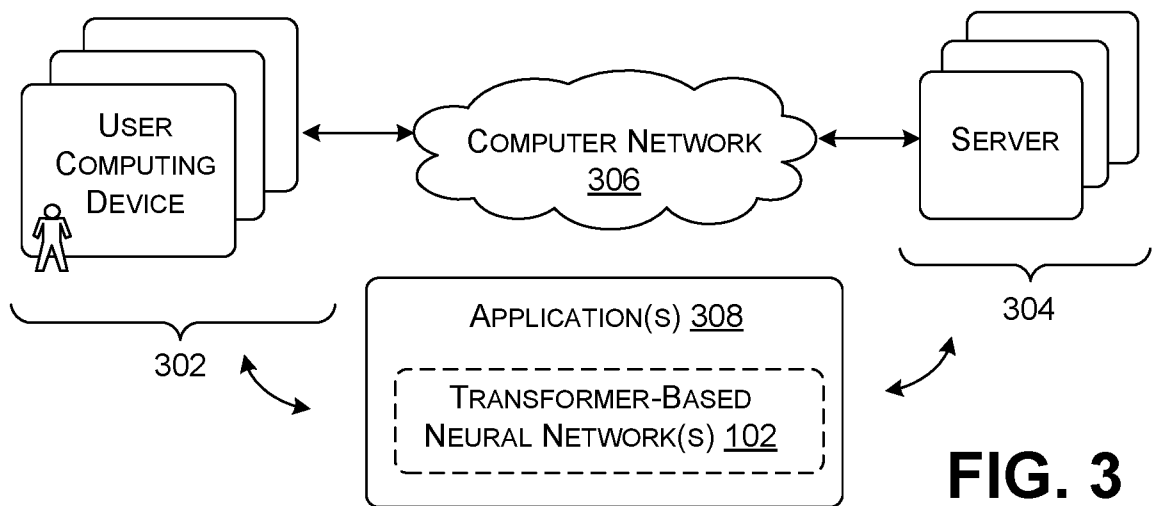
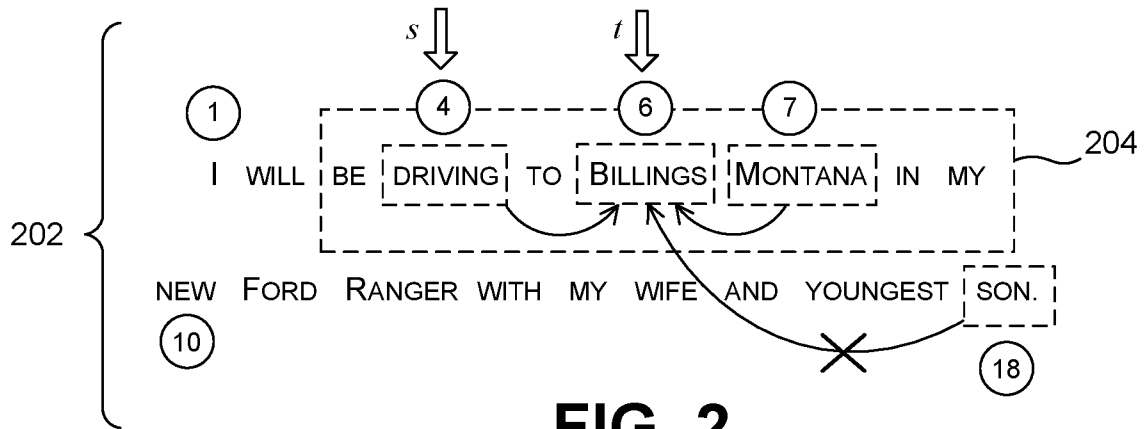
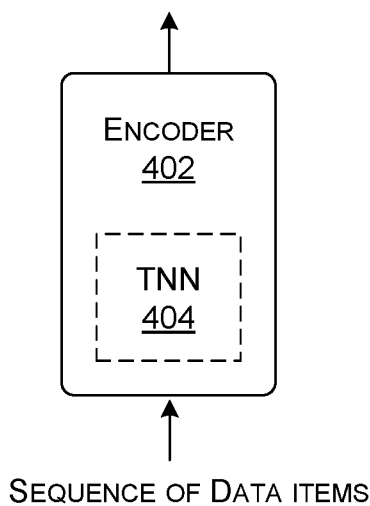


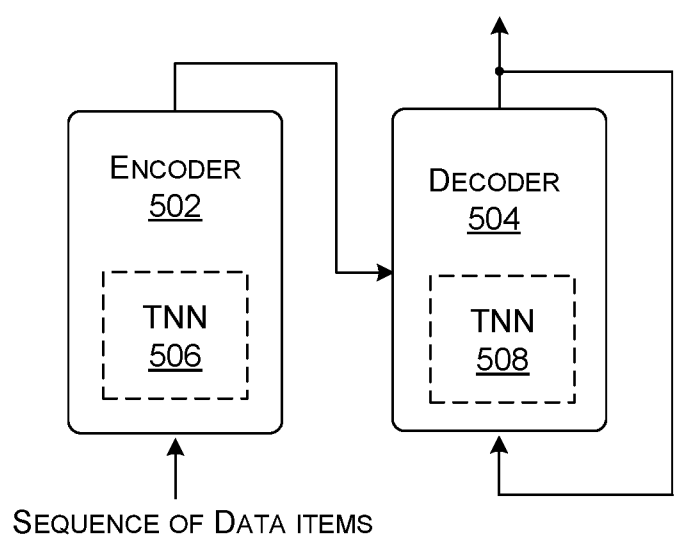
FIG. 1



SEMANTIC VECTOR AND/OR
CLASSIFICATION RESULT



TRANSFORMED SEQUENCE



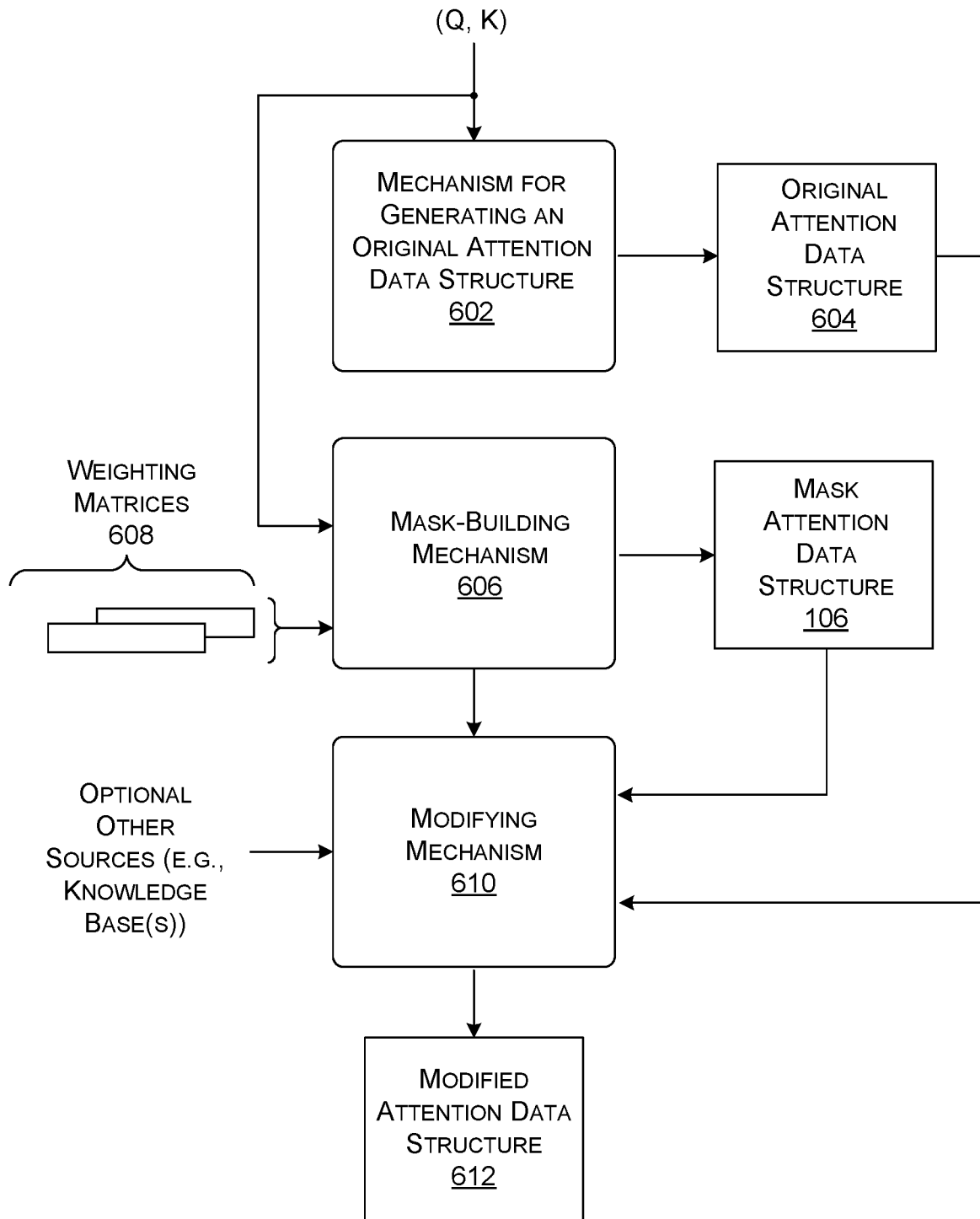
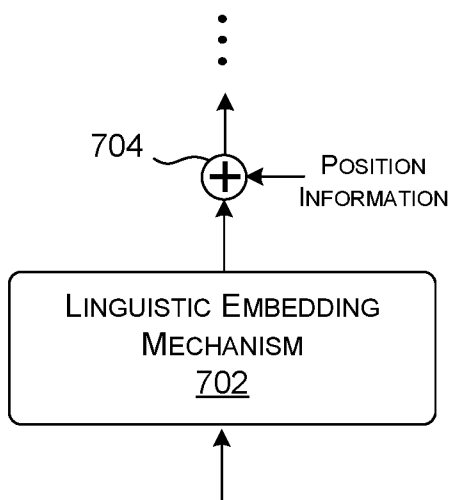
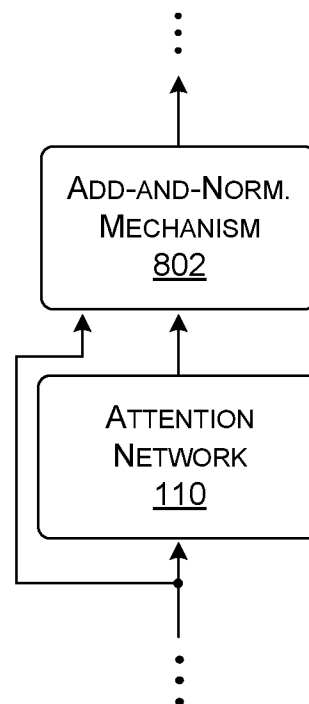
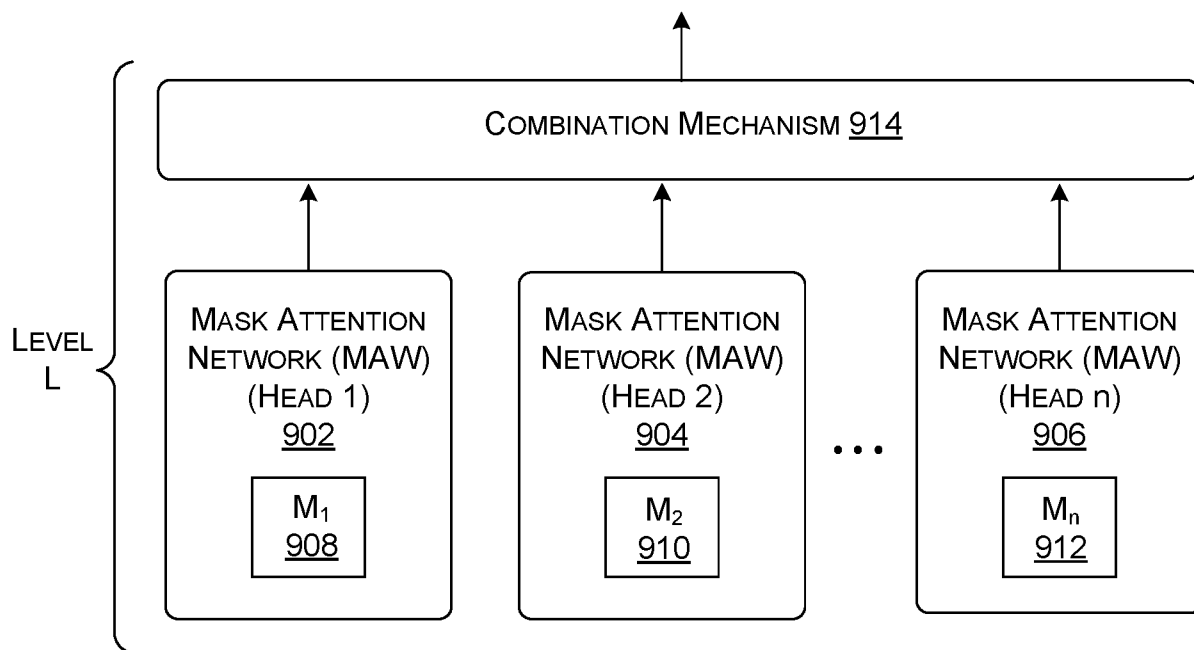


FIG. 6

**FIG. 7****FIG. 8****FIG. 9**

PROCESS FOR APPLYING A TRANSFORMER-BASED NEURAL NETWORK THAT
INCLUDES A MASK ATTENTION NETWORK 1002

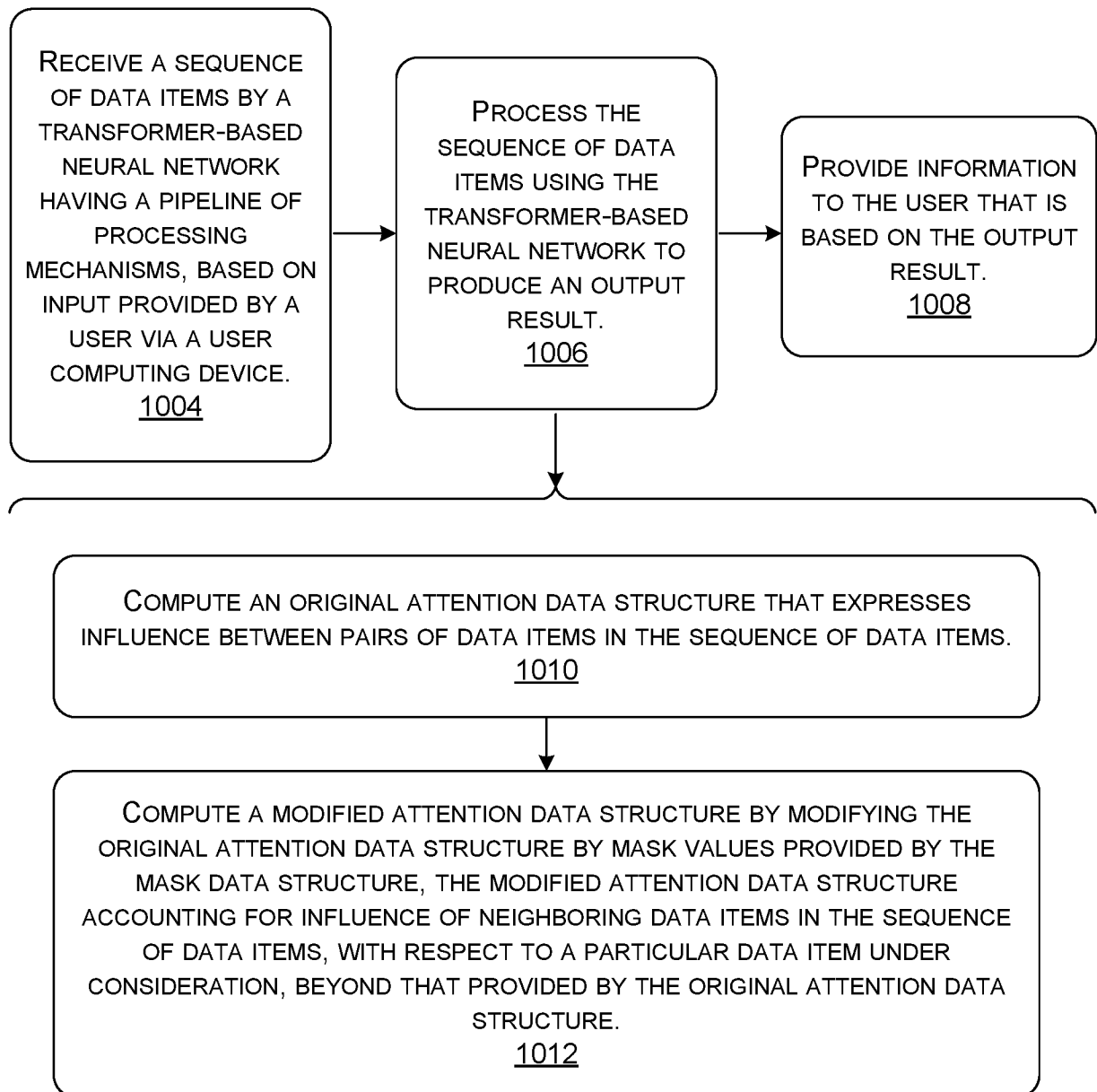


FIG. 10

PROCESS FOR BUILDING A MASK ATTENTION DATA STRUCTURE FOR A PARTICULAR PAIR OF DATA ITEMS IN THE SEQUENCE OF DATA ITEMS THAT IS MADE UP OF A FIRST DATA ITEM AND A SECOND DATA ITEM 1102

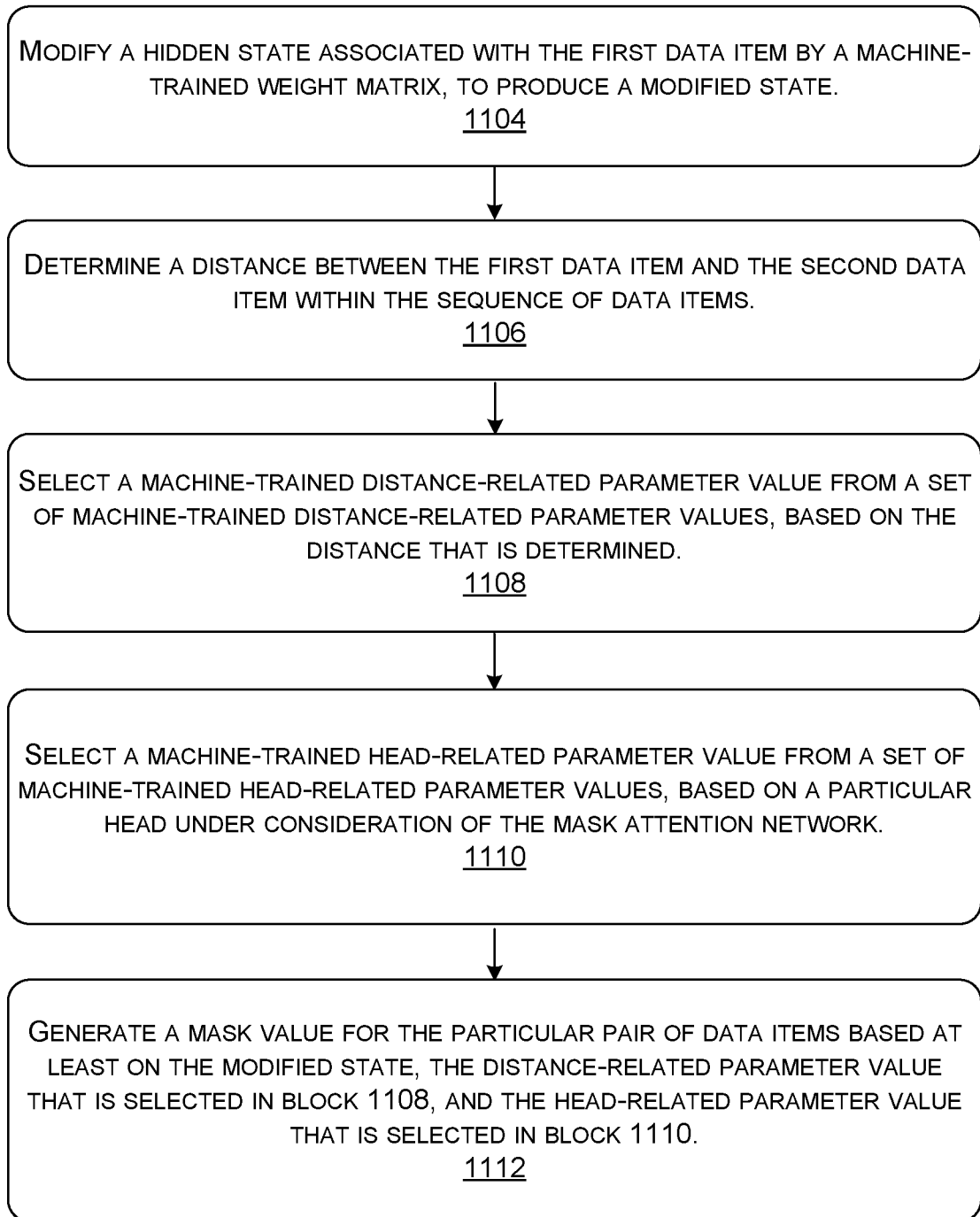


FIG. 11

PROCESS FOR APPLYING A PARTICULAR IMPLEMENTATION OF A TRANSFORMER-BASED
NEURAL NETWORK THAT INCLUDES A MASK ATTENTION NETWORK 1202

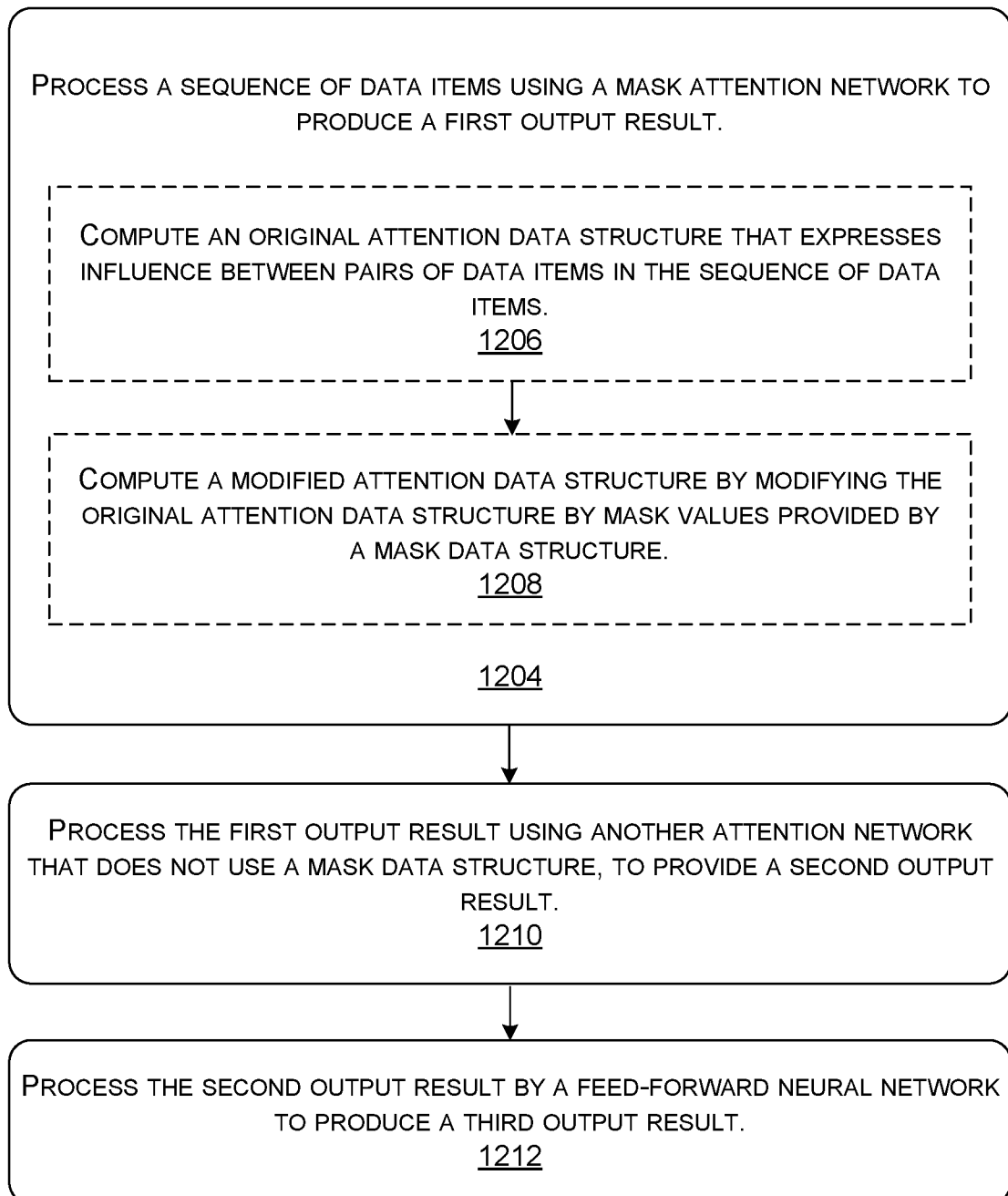


FIG. 12

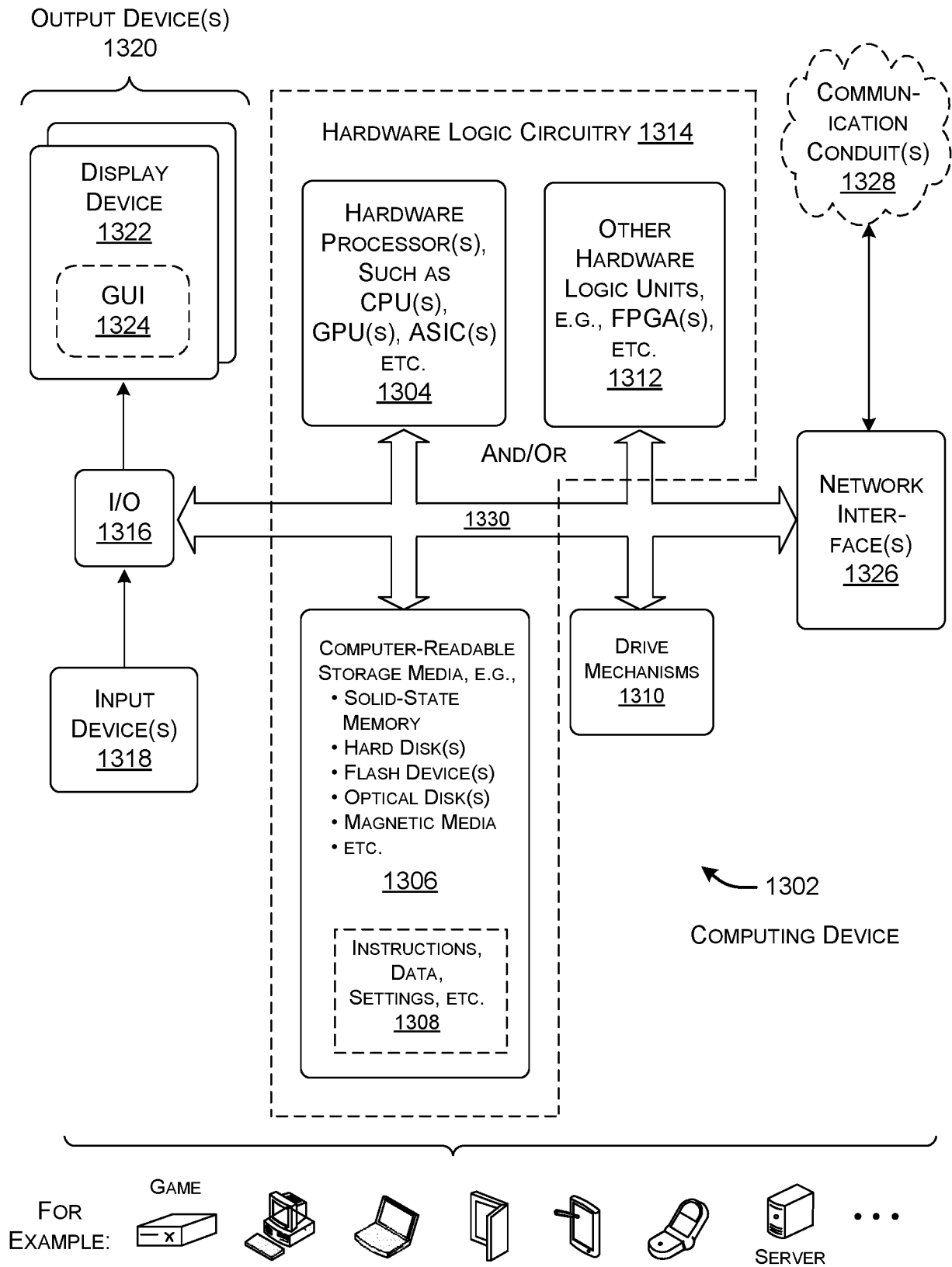


FIG. 13

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2021/032840

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06N3/04 G06N3/08
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)
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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	JINBAE IM ET AL: "Distance-based Self-Attention Network for Natural Language Inference", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 6 December 2017 (2017-12-06), XP080845156, abstract; figures 1-4 paragraph [0001] - paragraph [4.2.4] paragraph [05.2] ----- -/--	1-15



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

17 August 2021

Date of mailing of the international search report

26/08/2021

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

Cilia, Elisa

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2021/032840

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	Vaswani Ashish ET AL: "Attention Is All You Need", 31st Conference on Neural Information Processing Systems NIPS 2017, 9 December 2017 (2017-12-09), XP055832424, Long Beach, CA, USA Retrieved from the Internet: URL:https://papers.nips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf [retrieved on 2021-08-17] abstract; figures 1-2 paragraph [0001] - paragraph [0003] paragraph [0006] - paragraph [0007] -----	1-15
X	MATTHIAS SPERBER ET AL: "Self-Attentional Models for Lattice Inputs", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 4 June 2019 (2019-06-04), XP081372991, abstract; figures 2-3 paragraph [0001] - paragraph [4.1.3] paragraph [05.1] paragraph [05.3] paragraph [0006] -----	1-15
X	SUFENG DUAN ET AL: "Attention Is All You Need for Chinese Word Segmentation", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 31 October 2019 (2019-10-31), XP081651489, abstract; figures 1-3 paragraph [0001] - paragraph [02.2] paragraph [03.1] - paragraph [0005] -----	1-15
A	TAO SHEN ET AL: "DiSAN: Directional Self-Attention Network for RNN/CNN-Free Language Understanding", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 14 September 2017 (2017-09-14), XP081283052, abstract; figures 1-4 paragraph [0001] - paragraph [05.3] ----- -/--	1-15

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2021/032840

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	GUO MAOSHENG ET AL: "Gaussian Transformer: A Lightweight Approach for Natural Language Inference", PROCEEDINGS OF THE AAAI CONFERENCE ON ARTIFICIAL INTELLIGENCE, vol. 33, 17 July 2019 (2019-07-17), pages 6489-6496, XP055831388, ISSN: 2159-5399, DOI: 10.1609/aaai.v33i01.33016489 the whole document -----	1-15