

A System and Method for Incremental Learning in Machine Translation

5

TECHNICAL FIELD

The present disclosure relates broadly to a system for incremental learning in machine translation and a method of incremental learning in machine translation.

10

BACKGROUND

Machine translation systems and methods have been developed to translate words, phrases, and sentences in one language into a different language. With the development of new words and phrases over time, i.e. new translation materials being made available over time, there is a demand for machine translation systems and methods that are capable of leveraging and processing such new data to output linguistically meaningful translations.

While there are existing machine learning models to implement systems and methods for machine translation, retraining of such existing models require a substantial amount of training time and computational resources. Further, previous training data (for having sufficient data for training) may not be readily available. It has also been recognised that directly adapting new data on a trained model may lead to a technical problem referred to as “catastrophic forgetting”, in which the newly trained model may be capable of learning/acquiring knowledge from new data but the knowledge learned/acquired from previous data may be forgotten.

In view of the above, there exists a need to provide a system for incremental learning in machine translation and a method of incremental learning in machine translation that address or at least ameliorate one or more of the above-mentioned problems.

SUMMARY

In accordance with an aspect of the present disclosure, there is provided a system for incremental learning in machine translation, the system comprising an input module for obtaining a source language input to be translated into a target language output; an ensemble model module coupled to the input module, the ensemble model module comprising a baseline model trained using a first dataset, the baseline model arranged to receive the source language input to generate a baseline model target language output with associated one or more baseline model probability scores; at least one first adapted model trained using a second dataset, the second dataset being smaller than the first dataset, the at least one first adapted model arranged to receive the source language input to generate a first adapted model target language output with associated one or more first adapted model probability scores; and a determination module coupled to the ensemble model module for receiving the one or more baseline model probability scores and the one or more first adapted model probability scores, the determination module arranged to generate a system target language output based on the received one or more baseline model probability scores and the received one or more first adapted model probability scores.

The system may further comprise the at least one first adapted model being selected from a plurality of adapted models, the plurality of adapted models comprising at least a second adapted model trained using a third dataset, the third dataset being different in size from the second dataset used to train the at least one first adapted model.

One of the second dataset and the third dataset may be a subset of the other of the second dataset and the third dataset.

The second dataset and the third dataset may each be a part of an ensemble dataset.

The second dataset may comprise newer data obtained over time after establishment of the first dataset.

The first dataset and the second dataset may each comprise language pair data of a source language and a target language.

5 The determination module may be arranged to perform an averaging operation based on the received one or more baseline model probability scores and the received one or more first adapted model probability scores to generate the system target language output.

10 The baseline model and the at least one first adapted model may each be based on a transformer deep learning model.

The system may further comprise an output module coupled to the determination module, the output module being arranged to receive the system target language output from the determination module and to provide the system target language output to a user.
15

In accordance with another aspect of the present disclosure, there is provided a method of incremental learning in machine translation, the method comprising obtaining a source language input to be translated into a target language output; providing an ensemble model module, the ensemble model module comprising a baseline model
20 trained using a first dataset and at least one first adapted model trained using a second dataset, the second dataset being smaller than the first dataset; receiving, at the baseline model, the source language input; generating, using the baseline model, a baseline model target language output with associated one or more baseline model probability scores; receiving, at the at least one first adapted model, the source language input;
25 generating, using the at least one first adapted model, a first adapted model target language output with associated one or more first adapted model probability scores; receiving at a determination module the one or more baseline model probability scores and the one or more first adapted model probability scores; and generating, using the determination module, a system target language output based on the received one or
30 more baseline model probability scores and the received one or more first adapted model probability scores.

The method may further comprise selecting the at least one first adapted model from a plurality of adapted models, the plurality of adapted models comprising at least a second adapted model trained using a third dataset, wherein the third dataset is different in size from the second dataset used to train the at least one first adapted model.

5

One of the second dataset and the third dataset may be a subset of the other of the second dataset and the third dataset.

The second dataset and the third dataset may each be a part of an ensemble dataset.

10

The method may further comprise obtaining newer data over time after establishment of the first dataset to form data of the second dataset.

The first dataset and the second dataset may each comprise language pair data of a source language and a target language.

15

The method may further comprise performing at the determination module an averaging operation based on the received one or more baseline model probability scores and the received one or more first adapted model probability scores to generate the system target language output.

20

The baseline model and the at least one first adapted model may each be based on a transformer deep learning model.

25

The method may further comprise coupling an output module to the determination module; receiving the system target language output from the determination module; and providing the system target language output to a user using the output module.

In accordance with another aspect of the present disclosure, there is provided a computer readable storage medium having stored thereon software instructions that, when executed by a system processing module of a system for incremental learning in machine translation, cause the system processing module to perform a method described above.

30

BRIEF DESCRIPTION OF THE DRAWINGS

Exemplary embodiments of the present disclosure will be better understood
5 and readily apparent to one of ordinary skill in the art from the following written
description, by way of example only, and in conjunction with the drawings, in which:

FIG. 1 is a schematic diagram of a system for machine translation in an
exemplary embodiment.
10

FIG. 2 is a schematic drawing showing an overall architecture of a translation
engine in an exemplary embodiment.

FIG. 3 is a schematic framework for constructing one or more ensemble
15 models in an exemplary embodiment.

FIG. 4 is a schematic framework for generating a system target language
output for a source language input using an ensemble model in an exemplary
embodiment.
20

FIG. 5 is a graph showing the performance comparison for Chinese-to-English
translation in an experiment.

FIG. 6 is a graph showing the performance comparison for English-to-Chinese
25 translation in an experiment.

FIG. 7 is a graph showing the performance comparison for Malay-to-English
translation in an experiment.

FIG. 8 is a graph showing the performance comparison for English-to-Malay
30 translation in an experiment.

FIG. 9 is a schematic flowchart for illustrating a method of machine translation
in an exemplary embodiment.

FIG. 10 is a schematic drawing of a computer system suitable for implementing an exemplary embodiment.

5 DETAILED DESCRIPTION

Exemplary embodiments herein may provide a method of applying machine learning to incremental learning in machine translation. The exemplary method comprises the use of an ensemble model that comprises two or more models that include a baseline
10 model trained using a large dataset, and at least one adapted model trained using a relatively smaller dataset as compared to the large dataset. In exemplary embodiments, the large dataset and the smaller dataset each comprise language pair data of a source language and a target language (the source language and the target language being different languages), i.e. language pair data of the same two different languages (e.g.
15 language pair data of the English and Chinese languages). Further, the second dataset comprises data gradually obtained over time, e.g. after establishment of the large dataset.

In exemplary embodiments, the large dataset may also be referred to as baseline training data, legacy data, or a first dataset. The smaller dataset may also be referred to
20 a second dataset in various exemplary embodiments, and is a part of an ensemble dataset.

FIG. 1 is a schematic diagram of a system for machine translation in an exemplary embodiment. The system 100 is an ensemble-based machine learning system arranged
25 to perform machine translation. The system 100 can be for incremental learning in machine translation.

In the exemplary embodiment, the system 100 comprises an input module 102 for obtaining a source language input to be translated into a target language output. The
30 system 100 further comprises an ensemble model module 104 coupled to the input module 102. The input module 102 may be configured to perform input processing to process the source language input into a form that can be processed by the ensemble model module 104, e.g. the source language input may be processed into a sequence of tokens with tags as an input to the ensemble model module 104.

In the exemplary embodiment, the ensemble model module 104 comprises a baseline model 103 trained using a first dataset. The baseline model 103 is arranged to receive the source language input from the input module 102 to generate a baseline model target language output with associated one or more baseline model probability scores. The ensemble model module 104 further comprises at least one first adapted model 105 trained (fine-tuned) using a second dataset, the second dataset being smaller than the first dataset. The at least one first adapted model 105 is arranged to receive the source language input from the input module 102 to generate a first adapted model target language output with associated one or more first adapted model probability scores. As such, the baseline model 103 and the at least one first adapted model 105 operate in parallel within the ensemble model module 104 with respect to the processed source language input from the input module 102.

In the exemplary embodiment, the system 100 further comprises a determination module 106 coupled to the ensemble model module 104 for receiving the one or more baseline model probability scores and the one or more first adapted model probability scores. The determination module 106 is arranged to generate a system target language output 107 based on the received one or more baseline model probability scores and the received one or more first adapted model probability scores.

In the exemplary embodiment, the system 100 further comprises a processing module 108 that is arranged to control the operations of the components of the system 100, e.g., the input module 102, the ensemble model module 104, and the determination module 106.

In some exemplary embodiments, the system 100 may further comprise an output module 110 coupled to the determination module 106, the output module 110 being arranged to receive the system target language output 107 from the determination module 106 and to provide the system target language output 107 to a user. In some exemplary embodiments, the system target language output 107 may be processed into an output language without a sequence of tokens. In some exemplary embodiments, the system target language output 107 may be further processed into audio or text to be presented via the output module 110. In some exemplary embodiments, the output

module 110 may be provided in the form of a monitor or a display that displays a user interface (e.g., a graphical user interface).

5 In the exemplary embodiment, the at least one first adapted model is selected from a plurality of adapted models, the plurality of adapted models comprising at least a second adapted model trained using a third dataset, the third dataset being different in size from the second dataset used to train the at least one first adapted model.

10 In the exemplary embodiment, one of the second dataset and the third dataset is a subset of the other of the second dataset and the third dataset. For example, the second dataset may be smaller than the third dataset and may be a subset of the third dataset.

15 In the exemplary embodiment, the second dataset and the third dataset are each a part of an ensemble dataset. For example, the second dataset and the third dataset are each a subset of the ensemble dataset. Further, "a part" may also mean a full part, i.e. one of the datasets may be the entire ensemble dataset.

20 In the exemplary embodiment, data of the ensemble dataset comprises newer data obtained over time after establishment of the first dataset. Thus, the second dataset comprises newer data obtained over time after establishment of the first dataset. In the exemplary embodiment, newer data may be obtained a plurality of times spanning over a period of time after the establishment of the first dataset. In one exemplary embodiment, newer data may be obtained after an arbitrary amount of time has passed from the establishment of the first dataset. Newer data may additionally/alternatively be obtained
25 at a pre-determined interval of time (e.g. T months, T being a number) after the establishment of the first dataset. It will be appreciated that other intervals of time (e.g. days, weeks, years) may also be used.

30 In the exemplary embodiment, the first dataset and the second dataset each comprise language pair data of a source language and a target language. In the exemplary embodiment, the language pair data may comprise, for example, a pair of sentences comprising a sentence in the source language (e.g. a source sentence) and a sentence in the target language which is a translation of the source sentence.

In the exemplary embodiment, the determination module 106 is arranged to perform an averaging operation based on the received one or more baseline model probability scores and the received one or more first adapted model probability scores to
5 generate the system target language output 107.

In the exemplary embodiment, the baseline model and the at least one first adapted model are each based on a transformer deep learning model. For example, an encoder-decoder model is used.
10

In use, training is performed on the baseline model 103 using the first dataset. The baseline model 103 may already be pre-trained prior to training of the at least one first adapted model 105. For example, the baseline model 103 may already be in use for machine translation and previously trained on legacy data (or data obtained in the past).
15 Training is performed on the at least one first adapted model 105 using the second dataset, the second dataset being smaller than the first dataset. A source language input (e.g. English) is received at the input module 102. Processing is performed on the source language input and the processed source language input is transmitted to the ensemble model module 104 that comprises the baseline model 103 and the at least one first
20 adapted model 105. The baseline model 103 receives such source language input from the input module 102 and generates a baseline model target language output with associated one or more baseline model probability scores. The at least one first adapted model 105 receives such source language input from the input module 102 and generates a first adapted model target language output with associated one or more first adapted
25 model probability scores. The ensemble model module 104 transmits the baseline model target language output and the first adapted model target language output to the determination module 106. In other words, the determination module 106 also receives the one or more baseline model probability scores and the one or more first adapted model probability scores. Based on the one or more baseline model probability scores and the one or more first adapted model probability scores, the determination module 106
30 generates the system target language output 107.

With the above exemplary implementation, retraining of the baseline model (e.g. using a combination of legacy and newer data) may be omitted and therefore,

a substantial amount of training time and computational resources may be saved. Further, with usage of newer data to only train the at least one first adapted model and with the usage of the ensemble module comprising the trained baseline model and the newer trained at least one first adapted model, the requirement for locating
5 previous training data (e.g. for having sufficient data for training a single machine translation model) may be omitted. In addition, it has been found by the inventors that using the ensemble module comprising the trained baseline model and the newer trained at least one first adapted model, the technical issue of “catastrophic forgetting” may be prevented, e.g. since the baseline model is still trained on
10 knowledge learned/acquired from previous data or legacy data.

In the following paragraphs, a brief introduction to machine learning being applied to machine language translation is provided.

15 In general, to train a model for language translation, a dataset of sentence pairs (in the two languages that the translation is to be performed between) is used where each sentence in the pair is a translation of the other. Such a dataset is used to train a machine translation model using techniques such as sequence-to-sequence learning with attention.

20

In sequence-to-sequence learning, a model is trained to take a sequence of tokens in one language (the source language) and produce a translation of that sequence in another language (the target language). Attention mechanisms are typically used to allow the model to focus on specific parts of the input when generating the output, which can
25 help to improve the accuracy of the translation. Typically, the model may be fine-tuned using techniques such as gradient descent.

Transformer models are a type of deep learning models or neural network architectures that may be used for machine translation. A transformer model can use
30 attention mechanisms or self-attention to allow the model to focus on specific parts of a language input (from a source language) when generating a language output (to a target language). The use of self-attention allows the transformer model to parallelize computation across all of an input and it may handle longer input sequences.

A transformer model typically comprises a series of encoder layers to process an input and a series of decoder layers to generate an output. The encoder layers may use self-attention to compute representations of an input and the decoder layers may use both self-attention and attention over the encoder representations to generate an output.

5

In exemplary embodiments, a translation engine may be provided based on a transformer deep learning model. FIG. 2 is a schematic drawing showing an overall architecture of a translation engine in an exemplary embodiment. In exemplary embodiments, the baseline model and the at least one first adapted model may each be based on the architecture shown in FIG. 2.

10

For the deep learning model 200 (a transformer model), a language input 202 with e.g. a code switch and tags may be provided. The language input is typically processed as a sequence of tokens (e.g., words or subwords) in a source language. A language input with a code switch and tags typically refers to a sequence of tokens in which some tokens are in one language (the source language) and other tokens are in another language (e.g. the target language). The tokens may be annotated with a "code-switch" tag to indicate that they are written in either a source or a target language.

15

20

At an input processing layer 201, the language input 202 is processed so that input token embedding 206 and input tag embedding 208 are automatically obtained through an embedding lookup operation. The token embedding 206 and tag embedding 208 are then concatenated together as the input embedding 204. Positional encoding 210 is performed on the position index of tokens of the processed language input. The positional encoding 210 is used to incorporate an order of the tokens as a representation of the input. Typically, positional encoding is implemented as a fixed-length vector that is added to the embedding of each input token. The vector encodes the position of the token in the input sequence and allows the model 200 to use this information when generating an output. For example, sinusoidal functions may be used for positional encoding.

25

30

After performing positional encoding, the input embedding and position embedding are added and transmitted to a first multi-head attention layer 212 and a first addition and normalization layer 214.

In the exemplary embodiment, a multi-head attention layer (such as the first multi-head attention layer 212) is used to compute a weighted sum of its input. Typically, the input sequence is transformed into multiple different representation subspaces, and the attention mechanism is applied independently to each of these subspaces. The attention mechanism is applied multiple times in the multi-head attention layer. The output of the attention mechanism from different subspaces is then concatenated and projected to produce a final weighted sum. The multi-head attention layer may therefore attend to different parts of the input sequence simultaneously, capture dependencies between the input tokens and use such information when generating its output (or the output of the layer).

In the exemplary embodiment, the output of the first multi-head attention layer 212 is transmitted to the first addition and normalization layer 214.

In the exemplary embodiment, an addition and normalization layer is provided as a residual layer to improve the stability and generalization of the model 200. Each addition and normalization layer performs an addition operation (between an output of a previous layer and an input to the current layer) and a normalization operation.

For the first addition and normalization layer 214, the addition operation adds the input to the first multi-head attention layer 212 to the output of the first multi-head attention layer 212, and the normalization operation scales and shifts the input for a fixed mean and variance. In the exemplary embodiment, the output of the first addition and normalization layer 214 is transmitted to a first feedforward layer 216 and a second addition and normalization layer 218.

In the exemplary embodiment, a feedforward layer is provided to transform its input into another representation. The feedforward layer may comprise a series of connected layers to apply a series of linear transformations to produce a new representation of its input. An activation function may be used to introduce nonlinearity.

In the exemplary embodiment, the output of the first feedforward layer 216 is transmitted to the second addition and normalization layer 218. The operations of the second addition and normalization layer 218 are substantially similar to the operations of

the first addition and normalization layer 214, i.e. processing with respect to its own input from the first addition and normalization layer 214 in relation to the output of the first feedforward layer 216.

5 With reference to FIG. 2, the left side of the transformer model 200 is typically called an encoder while the right side of the transformer model 200 is typically called the decoder. There can be N number of encoder blocks and N number of decoder blocks. For FIG. 2, each encoder block contains the same layers illustrated therein e.g. see numerals 204, 210, 212, 214, 216, 218. The output of the second addition and
10 normalization layer 218 from a final encoder block is transmitted to a second multi-head attention layer 220 in each decoder block.

 The second multi-head attention layer 220 receives the output of the second addition and normalization layer 218 and an output associated with a target language
15 output. The processing of the target language output is described as follows.

 A target language output 222 may be provided. In the exemplary embodiment, the input to the decoder (the right side of FIG. 2) is the target language sequence shifted one position to the right by an artificial token "<start>" that typically indicates the beginning of
20 a sentence in order to generate a next token (as the generation of a target output at each position is based on the receipt and use of the previous token; i.e. the generation of a current target output at each position depends on a received previous token). Similar to the description of the language input 202, a language output is typically processed as a sequence of tokens (e.g., words or subwords) in a target language. At a target output
25 processing layer 224, the target language output 222 is processed so that output token embedding is obtained via an output embedding lookup. Positional encoding 230 is performed on the index of tokens of the processed target language output 222. Similar to the positional encoding 210, the positional encoding 230 is used to incorporate an order of the tokens as a representation of the target language output.

30

 After performing positional encoding, target output embedding and position embedding are added. The processed target language output 222 is transmitted to a masked multi-head attention layer 232 and a third addition and normalization layer 234. The masked multi-head attention layer 232 functions substantially similarly to the first

multi-head attention layer 212, with the additional usage of a masking mechanism to mask future positions/tokens in the target language output. With its functions, the masked multi-head attention layer 232 may be able to generate its output in a correct order. In the exemplary embodiment, the output of the masked multi-head attention layer 232 is transmitted to the third addition and normalization layer 234.

The operations of the third addition and normalization layer 234 are substantially similar to the operations of the first addition and normalization layer 214, i.e. adding the input to the masked multi-head attention layer 232 to the output of the masked multi-head attention layer 232. In the exemplary embodiment, the output of the third addition and normalization layer 234 is transmitted to the second multi-head attention layer 220 and to a fourth addition and normalization layer 236.

In the exemplary embodiment, the second multi-head attention layer 220 receives the output of the second addition and normalization layer 218 and the output of the third addition and normalization layer 234. The second multi-head attention layer 220 functions substantially similarly to the first multi-head attention layer 212. The output of the second addition and normalization layer 218 and the output of the third addition and normalization layer 234 are processed with an attention mechanism multiple times for a decoding operation. In the exemplary embodiment, the output of the second multi-head attention layer 220 is transmitted to the fourth addition and normalization layer 236.

In the exemplary embodiment, the operations of the fourth addition and normalization layer 236 are substantially similar to the operations of the first addition and normalization layer 214, i.e. processing with respect to its own input from the third addition and normalization layer 234 in relation to the output received from the second multi-head attention layer 220. In the exemplary embodiment, the output of the fourth addition and normalization layer 236 is transmitted to a second feedforward layer 238 and a fifth addition and normalization layer 240.

In the exemplary embodiment, the second feedforward layer 238 functions substantially similarly to the first feedforward layer 216, e.g. to produce a new representation of its input. The output of the second feedforward layer 238 is transmitted to the fifth addition and normalization layer 240.

The operations of the fifth addition and normalization layer 240 are substantially similar to the operations of the first addition and normalization layer 214, i.e. processing with respect to its own input from the fourth addition and normalization layer 236 in relation to the output received from the second feedforward layer 238. In the exemplary embodiment, the output of the fifth addition and normalization layer 240 is transmitted to a linear layer 242.

In the exemplary embodiment, the linear layer 242 performs a linear transformation of its input, e.g. scaling and shifting its input according to values in a weight matrix and a bias vector. The output of the linear layer 242 is transmitted to a softmax layer 244.

In the exemplary embodiment, the softmax layer 244 applies the softmax function to its input which maps the input values (from the output of the linear layer 242) to a probability distribution over the target vocabulary. The output of the softmax layer 244 is a probability distribution over the target vocabulary, with the probability of each word in the target vocabulary being given by the corresponding element of the output of the softmax layer 244. See output probabilities 246. As such, the softmax layer 244 functions as the output layer of the model 200. In the exemplary embodiment, the translation with the highest probability may then be chosen as the final prediction of the model 200. In the exemplary embodiment, the output may be a sequence of tokens in the target language.

FIG. 3 is a schematic framework for constructing one or more ensemble models in an exemplary embodiment. In the exemplary embodiment, one or more steps of the method are computer-implemented. Alternatively, the method is a computer-implemented method. In the exemplary embodiment, the method may be implemented by a processing module (e.g. see processing module 108 described with reference to FIG. 1). For ease of reference, like naming conventions are used for exemplary implementations of similar modules as described in FIG. 1.

30

In the exemplary embodiment, a first dataset (may be also referred to as baseline training data or legacy data or old/older training data) is obtained. The first dataset comprises language pair data of a first language (or a source language) and a second language (or a target language).

In the exemplary embodiment, a baseline model 302 is trained using the first dataset. The baseline model 302 is based on a transformer deep learning model and uses an encoder-decoder architecture. In the training, the first dataset is fed as an input to the baseline model 302. The encoder extracts features from the source input and sends the features to the decoder of the baseline model 302.

In machine learning, it is known that a training objective is a goal that a machine learning model is attempting to achieve during training. The training objective is typically defined in terms of a loss function, which measures the difference between the model's predictions and the true labels or values that the model is trying to predict. The training objective is typically to minimise the loss function during training and is typically achieved by an adjustment of the model's parameters.

In the exemplary embodiment, in the training of the models, cross-entropy loss is used as the training objective L and may be calculated based on equation (1) :

$$L = -\sum_{w=1}^{|S|} \sum_{e=1}^{|V|} y_{w,e} \log(\hat{y}_{w,e}) \quad \text{--- (1)}$$

Where

- |S| = Length of Sentence
- |V| = Length of Vocabulary
- $\hat{y}_{w,e}$ = predicted probability of vocabulary entry V_e at position w
- $y_{w,e} = 1$ when the vocabulary entry is the correct word
- $y_{w,e} = 0$ when the vocabulary entry is not the correct word

Cross-entropy loss may be used to measure the distance/difference between the model's predicted probability distribution and the true probability distribution. For the equation, S is a target sentence in the training data. |S| is length of the sentence S or number of words in sentence S. V is the set of predefined target vocabulary with |V| being the length of vocabulary or size of the vocabulary set V. $\hat{y}_{w,e}$ is the predicted probability for vocabulary entry V_e when the correct word is S_w . $y_{w,e} = 1$ when the predicted target word or vocabulary entry V_e is the correct word S_w . $y_{w,e} = 0$ when the predicted target word or vocabulary entry V_e is not the correct word S_w . In the exemplary embodiment, teacher forcing strategy is used during training where the correct target word from the

previous time step is used as the decoder input to the current time step. Therefore, this loss function is in essence a sum over the negative log likelihoods that the model gives to the correct word at each time step or position in the output sentence.

5 In the exemplary embodiment, when training, the loss of a few training sentence pairs in a batch is summed together to result in a batch loss. The batch loss is then used to perform a mini-batch gradient descent to update parameters of the model being trained including both the encoder and the decoder. By performing the above process iteratively over batches, parameters of the model being trained are updated until convergence. In
10 testing (decoding/ translating), the predicted output (not like the true output in the training phrase) from the previous time step is fed as a target input to the current time step. The predicted outputs from all time steps are then sequentially combined to produce the final translation output from the trained model.

15 It will be appreciated that the exemplary embodiments are not limited to the training method/technique described herein. That is, other training techniques may be used for the training of the models (e.g. baseline model, adapted model etc.).

 In the exemplary embodiment, after the baseline model 302 is trained and
20 obtained, another dataset is obtained. This dataset may be arbitrarily termed as a second dataset in this example for ease of explanation. For example, the second dataset may be formed with new training data available after the baseline model 302 is trained. See e.g. New Data (Batch 1) 304 and/or New Data (Batch 2) 310 of FIG. 3. In the exemplary embodiment, New Data (Batch 1) 304 is smaller than the first dataset in size. For that
25 matter, all new data obtained after establishment of the first dataset e.g. until New Data (Batch K) 316, when added together, may be smaller than the first dataset in size. The New Data (Batch 1) 304 comprises language pair data of a first language (or a source language) and a second language (or a target language). In the exemplary embodiment, the New Data (Batch 1) 304 is obtained after a period of time has passed after
30 obtaining/establishment of the first dataset.

 In the exemplary embodiment, after the New Data (Batch 1) 304 is obtained, an adapted model is trained using the New Data (Batch 1) 304. See Adapt Model 1 306 of FIG. 3. In the exemplary embodiment, each adapted model may be an adaptation of the

baseline model 302 trained on an accumulation of new batches of data. For example, the Adapt Model 1 306 may be based on the baseline model 302. The adapted model may be arbitrarily termed as a first adapted model in this example for ease of explanation. In the training of the Adapt Model 1 306, the New Data (Batch 1) 304 is fed as an input to the Adapt Model 1 306 (e.g. an adaptation of the baseline model 302). The training/adaptation process for Adapt Model 1 306 is substantially similar to the training process for the baseline model 302 except that parameters of the baseline model 302 may be randomly initialized while parameters of the Adapt Model 1 306 can be initialized with parameters of the trained baseline model 302.

10

In the exemplary embodiment, the baseline model 302 and the Adapt Model 1 306 are then used to construct an ensemble model. See Ensemble Model 1 308. In some exemplary embodiments, the Ensemble Model 1 308 may be a final translation model. In other exemplary embodiments, an ensemble of the baseline model 302 with at least one adapted model is performed to form a final translation model.

15

In the exemplary embodiment, over time, more new data may be obtained. See e.g. New Data (Batch 2) 310 of FIG. 3. The New Data (Batch 2) 310 may be obtained after a period of time has passed from obtaining the first dataset and the New Data (Batch 1) 304.

20

In the exemplary embodiment, after the New Data (Batch 2) 310 is obtained, another adapted model (see Adapt Model 2 312 of FIG. 3) is trained using the New Data (Batch 2) 310 as well as the New Data (Batch 1) 304, which collectively/accumulatively forms yet another dataset. As such, this dataset is an incremental dataset as compared to the dataset New Data (Batch 1) 304 used to train the previous adapted model Adapt Model 1 306. The another adapted model may be arbitrarily termed as a second adapted model in this example for ease of explanation. The yet another dataset (the incremental dataset described above) may be arbitrarily termed as a third dataset in this example for ease of explanation. As such, a second dataset (e.g. New Data (Batch 1) 304) is smaller than a later generated third dataset (e.g. New Data (Batch 1) 304 added with the New Data (Batch 2) 310) and the above second dataset is a subset of the above third dataset.

25

30

The Adapt Model 2 312 may be an adaptation of the baseline model 302 trained on the accumulation of new batches of data, i.e. New Data (Batch 2) 310 and New Data (Batch 1) 304. Each new adapted model (e.g. the Adapt Model 2 312) may also be an adaptation of the previous adapted model (in this case, the Adapt Model 1 306) trained on the accumulation of new batches of data, i.e. the New Data (Batch 2) 310 added to the New Data (Batch 1) 304. For example, for an incremental learning task, after obtaining the newest data (in this case, the New Data (Batch 2) 310), the newest data is added to the previous new/ensemble data (i.e. dataset(s) after establishment of the first dataset, or legacy data) for training of the baseline model or the previous adapted model (in this case, the Adapt Model 1 306) as the new adapted model (e.g. the Adapt Model 2 312). In the training, the third dataset, comprising the New Data (Batch 1) 304 and the New Data (Batch 2) 310, is fed as input to the Adapt Model 2 312..

In the exemplary embodiment, the trained baseline model 302 and the trained Adapt Model 2 312 are used to construct another ensemble model. See Ensemble Model 2 314. In some exemplary embodiments, the Ensemble Model 2 314 may supersede the Ensemble Model 1 308 constructed previously.

In the exemplary embodiment, over time, even newer data may be obtained. See e.g. New Data (Batch k) 316 of FIG. 3. The New Data (Batch k) 316 may be obtained after a period of time has passed from obtaining the first dataset, the New Data (Batch 1) 304, and the New Data (Batch 2) 310.

In the exemplary embodiment, after the New Data (Batch k) 316 is obtained, a k^{th} adapted model (see Adapt Model k 318 of FIG. 3) is trained using the New Data (Batch k) 316 as well as the New Data (Batch 1) 304, the New Data (Batch 2) 310, and other data obtained/accumulated between the time when New Data (Batch 2) 310 is obtained and when New Data (Batch k) 316 is obtained. These data collectively/accumulatively form a latest dataset, see $(k+1)^{\text{th}}$ dataset 317. As such, the $(k+1)^{\text{th}}$ dataset 317 is an incremental dataset as compared to e.g. the third dataset. The Adapt Model k 318 may be an adaptation of the baseline model 302 trained on the accumulation of new batches of data, i.e. the accumulative batches of data making up the $(k+1)^{\text{th}}$ dataset 317. The Adapt Model k 318 may also be an adaptation of the previous adapted model (in this case, Adapt Model $(k-1)$ (not shown) trained on the accumulation of new batches of data,

i.e. accumulation of the New Data (Batch 2) 310 and the New Data (Batch 1) 304 until the New Data (Batch k) 316. For example, for an incremental learning task, after obtaining the newest data (in this case, New Data (Batch k) 316), the newest data is added to the previous new/ensemble data (i.e. dataset(s) after establishment of the first dataset, or legacy data) for training of the baseline model or Adapt Model (k-1) (not shown) as the new Adapt Model k 318. In the training, the (k+1)th dataset 317 is fed as input to the Adapt Model k 318.

In the exemplary embodiment, the trained baseline model 302 and the trained Adapt Model k 318 are used to construct a latest ensemble model, or kth ensemble model. See Ensemble Model k 320. In some exemplary embodiments, the Ensemble Model k 320 may supersede the ensemble model(s) constructed previously.

In the exemplary embodiment, for selection of an adapted model to construct an ensemble model, one or more predefined or curated test sets from a latest accumulated dataset or new training data may be used to test the performance of each ensembled model. Thereafter, the ensembled model that can provide the best performance among such test sets may be selected for use.

In the exemplary embodiment, if the New Data (Batch k) 316 comprises the latest or newest data, the accumulated dataset, i.e. the (k+1)th dataset 317, can be termed as ensemble data or the ensemble dataset, i.e. total accumulated new data as compared to the first dataset or legacy data. That is, the ensemble data or the ensemble dataset comprises accumulated data after establishment of the first dataset.

25

It will be appreciated that the naming/designation of a first or second adapted models etc. and a second or third datasets etc. are flexibly used within the context of the framework example described above. The naming is to distinguish the different adapted models and datasets. For example, the Adapt Model k 318 may be termed the first adapted model trained with a second dataset (in this case, the accumulated (k+1)th dataset 317) and the naming convention is flexibly used with the other adapted models (e.g. the Adapt Model 1 306) and other datasets (e.g. the dataset New Data (Batch 1) 304) etc.

30

As such, using the flexible naming convention, with the framework 300, there can be at least one or a first adapted model (e.g. Adapt Model k 318) being selected from a plurality of adapted models (e.g. Adapt Model 1 306, Adapt Model 2 312), the plurality of adapted models comprising at least a second adapted model (e.g. Adapt Model 2 312) trained using a third dataset (e.g. accumulated New Data (Batch 2) 310 and New Data (Batch 1) 304), the third dataset being different in size from another dataset (e.g. accumulation of New Data (Batch 2) 310 and New Data (Batch 1) 304 until New Data (Batch k) 316, making up the $(k+1)^{\text{th}}$ dataset 317, and which may be termed as a second dataset) used to train the at least one adapted model (e.g. Adapt Model k 318). Similarly, the third dataset is also different in size from a second dataset that is used to train yet another adapted model (e.g. Adapt Model 1 306).

Thus, in the exemplary embodiment, constructing an ensemble model involves training a baseline model using a first dataset, e.g. "old" baseline training data or legacy data. With new training data available over time after the baseline model is trained, the baseline model is adapted based on using accumulated new batches of data (e.g. see e.g. $(k+1)^{\text{th}}$ dataset 317). These new batches of data are a part of (or full part of, in the case of the $(k+1)^{\text{th}}$ dataset 317) an ensemble dataset. In the exemplary embodiment, each adapted model is based on the baseline model being trained on a smaller and newer dataset after the establishment of the first dataset (or legacy data). In other exemplary embodiments, this may translate to a previous adapted model being trained on an accumulated dataset. For example, the Adapt Model 2 312 may be seen as the Adapt Model 1 306 (in turn based on the baseline model 302) being trained on the accumulated data, i.e. New Data (Batch 1) 304 and New Data (Batch 2) 310. Ensemble of the trained baseline model and the trained adapted model is then performed to construct an ensemble model. In the exemplary embodiment, the ensemble model may be stored in or comprised within an ensemble model module (see e.g. ensemble model module 104 described with reference to FIG. 1) and may be used to generate a system target language output (a translation of a source language input) via a determination module (see e.g. determination module 106 described with reference to FIG. 1).

In use, the above method of constructing an ensemble model may alleviate issues in machine translations related to "catastrophic forgetting" in which a newly trained model may be capable of learning knowledge from new data but the knowledge

learned from previous data may be forgotten. Thus, the ensemble model comprising the baseline model 302 and at least one adapted model (e.g. Adapt Model 1 306 and/or Adapt model k 318) may perform incremental learning tasks which is able to learn or capture new knowledge learned while retaining knowledge learned from previous/legacy data using the baseline model 302.

In an ensemble of the baseline model 302 and at least one adapted model (e.g. Adapt Model 1 306 and/or Adapt model k 318), the technical issue of “catastrophic forgetting” may be alleviated by averaging their respective output probabilities, for example, a baseline model probability score calculated by the baseline model 302 for a first (or source) language input and a first adapted model probability score calculated by the at least one adapted model for the first language input. The averaging operation may be performed by a determination module (see e.g. determination module 106 described with reference to FIG. 1) to generate a system second (or target) language output.

15

In the exemplary embodiment, the averaging operation is as follows. Let the probability score for a word in the target language vocabulary calculated by an i -th model (may be a baseline model or an adapted model) be P_i and the number of models involved be n . The ensemble probability score or final probability score P_{final} is calculated by equation (2).

20

$$P_{\text{final}} = \log(\sum_i^n \exp(P_i) / n) \text{ --- (2)}$$

FIG. 4 is a schematic framework for generating a system second or target language output for a first or source language input using an ensemble model in an exemplary embodiment. In the exemplary embodiment, the framework 400 may be implemented/performed by a processing module of a system (see e.g. processing module 108 described with reference to FIG. 1). For ease of reference, like naming conventions are used for exemplary implementations of similar modules as described in FIG. 1.

30

In the exemplary embodiment, a first or source language input 402 to be translated into a second or target language is provided to an input module (see e.g. input module 102 described with reference to FIG. 1). In the exemplary embodiment, the first language

(or source language) is English and the first language input is the sentence “The COVID-19 has brought challenges to the world”. The second language (or target language) is Chinese.

5 In the exemplary embodiment, the first language input 402 is processed and transmitted to an ensemble model module 404 (see e.g. ensemble model module 104 described with reference to FIG. 1) comprising an ensemble model, Ensemble Model k. Ensemble Model k comprises a Baseline Model 406 and an adapted model Adapt Model k 408. The Baseline Model 406 and the Adapt Model k 408 are trained and are ensembled
10 to construct Ensemble Model k according to framework 300 of FIG. 3. Compare e.g. Ensemble Model k 320 of FIG. 3.

 In the exemplary embodiment, the source language input 402 is transmitted to Baseline Model 406 to generate a baseline model target language output 410 comprising
15 associated one or more (at least one) baseline model probability scores (see probability scores in parentheses in FIG. 4). In the exemplary embodiment, the baseline model target language output 410 is “科维德 19(0.6), 给(0.8), 世界(0.9), 带来了(0.7), 挑战(0.9)。”。 Thus, in the example, there are five tokens with the associated probability scores 0.6, 0.8, 0.9, 0.7 and 0.9 respectively.

20 In the exemplary embodiment, the source language input 402 is also transmitted to Adapt Model k to generate a k^{th} adapted model target language output 412 comprising associated one or more (at least one) k^{th} adapted model probability scores (see probability scores in parentheses in FIG. 4). In the exemplary embodiment, the k^{th}
25 adapted model target language output 412 is “冠状病毒 19(0.9), 带给(0.7), 世界(0.9), 带来了(0.7), 挑战(0.9)。”。 Thus, in the example, there are five tokens with the associated probability scores 0.9, 0.7, 0.9, 0.7 and 0.9 respectively.

 In the exemplary embodiment, the baseline model probability scores for the
30 baseline model target language output 410 and the k^{th} adapted model probability scores for the k^{th} adapted model target language output 412 are transmitted to a determination module (see e.g. determination module 106 described with reference to FIG. 1) to generate a system target language output 416 based on the received baseline model probability scores and the first adapted model probability scores. The system target

language output 416 may comprise an ensemble model probability score for each portion of the system target language output 416. In the exemplary embodiment, the system target language output 416 is “冠状病毒 19(0.55), 给(0.48), 世界(0.9), 带来了(0.7), 挑战(0.9)”。Thus, in the example, there are five tokens with final probability scores 0.55, 0.48, 0.9, 0.7 and 0.9 respectively.

In the exemplary embodiment, the ensemble model probability scores are calculated using equation (2). For output 科维德 19(0.6) from the Baseline Model 406 where an output probability score for the same word 科维德 19 from the Adapt Model k 408 is 0 (which is omitted in the exemplary embodiment for simplicity of illustration), the ensemble probability score is calculated as $\log((\exp(0.6)+\exp(0))/2)=0.34$, where $n=2$ as there are two models to be ensembled here. That is, there is consideration provided for the same word 科维德 19 from the other model(s) comprised in the ensemble model module 404. Similarly, for output 冠状病毒 19 (0.9) from the Adapt Model k 408 where an output probability score for the same word 冠状病毒 19 from the Baseline Model 406 is 0 (which is not shown in the exemplary embodiment for simplicity of illustration), the ensemble probability score is calculated as $\log((\exp(0)+\exp(0.9))/2)=0.55$. Similarly, for output 给(0.8) from the Baseline Model 406 where the output probability score for the same word 给 from the Adapt Model k 408 is 0 (which is not shown in the exemplary embodiment for simplicity of illustration), the ensemble probability score is calculated as $\log((\exp(0.8)+\exp(0))/2)=0.48$. For output 带给(0.7) from the Adapt Model k 408 where the output probability score for the same word 带给 from the Baseline Model 406 is 0 (which is not shown in the exemplary embodiment for simplicity of illustration), the ensemble probability score is calculated as $\log((\exp(0)+\exp(0.7))/2)=0.41$. For output 世界(0.9) from the Baseline Model 406 and output for the same word 世界(0.9) from the Adapt Model k 408, the ensemble probability score is calculated as $\log((\exp(0.9)+\exp(0.9))/2)=0.9$. For output 带来了(0.7) from the Baseline Model 406 and output for the same word 带来了(0.7) from the Adapt Model k 408, the ensemble probability score is calculated as $\log((\exp(0.7)+\exp(0.7))/2)=0.7$. For output 挑战(0.9) from the Baseline Model 406 and output 挑战(0.9) from the Adapt Model k 408, the ensemble probability score is calculated as $\log((\exp(0.9)+\exp(0.9))/2)=0.9$.

In the exemplary embodiment, the final probability score for every possible token/output in the target vocabulary is computed at each position by averaging probability scores for every possible token/output from the Baseline Model 406 and from the Adapt Model k 408, using the averaging operation in equation (2). See above description. Thereafter, in a determination process at the determination module, a beam search with a beam size k is performed where the top k possible tokens with higher probability scores are chosen for that each position by the determination module. In the exemplary embodiment, at each position, only the best possible token from each model is shown for illustration purposes. For example, for the first position, 科维德 19 and 冠状病毒 19, are the best possible tokens from the Baseline Model 406 and the Adapt Model k 408, respectively (other tokens are omitted from consideration here). The final probability scores for 科维德 19 and 冠状病毒 19 are calculated as 0.34 and 0.55, respectively, using equation (2). For example, for the second position, 给 and 带给 are the best possible tokens from the Baseline Model 406 and the Adapt Model k 408, respectively (other tokens are omitted from consideration). The final probability scores for 给 and 带给 are calculated as 0.48 and 0.41, respectively, using equation (2). The output sequence with the highest probability at the end of the determination process is chosen as the final target output. For example, in the determination process, for the first position, 科维德 19(0.34) is not chosen and for the second position, 带给(0.41) is not chosen. As such, in the exemplary embodiment, the system target language output 416 is chosen as “冠状病毒 19(0.55), 给(0.48), 世界(0.9), 带来了(0.7), 挑战(0.9)。”.

In the exemplary embodiment, the system target language output 416 may be transmitted to an output module (see e.g. the output module 110 of FIG. 1) for output to a user. In some exemplary embodiments, the system target language output 416 may be transmitted to another server computer for storage and/or for further processing.

In the exemplary embodiment, the framework 400 shows an example of how an ensemble model can learn from a new batch of data (with respect to the Adapt Model k 408) while keeping the knowledge it learns from a first dataset, e.g. “old” baseline training data or legacy data (with respect to the Baseline Model 406). The framework 400 illustrates, for example, that while the Baseline Model 406 generates a Chinese translation of the English word COVID-19 “科维德 19” (not an accurate translation), the

adapted model Adapt Model k 408 can bring in new knowledge on translation, such as by generating a different Chinese translation of the English word COVID-19 that should be “冠状病毒 19” (a more accurate translation). As will be appreciated, “COVID-19” may not have been in the lexicon in the past e.g. in the legacy data and may have only entered
5 the lexicon after year 2020 resulting in newer data. Thus, if using the Baseline Model 406 trained on legacy data alone, a resulting translation of “COVID-19” is likely not to be accurate. With the ensemble, usage of the adapted model trained on newer data, e.g. on data obtained after year 2020 but with a trade-off that such data can result in a smaller dataset as compared to the legacy data, i.e. as an incremental learning task, can result
10 in a more accurate translation of the newer lexicon. Furthermore, in the exemplary embodiment, the framework 400 usefully allows the Baseline Model 406 to keep the knowledge learned from the first dataset, e.g. “old” baseline training data or legacy data, to avoid translation mistakes that may be generated by the Adapt Model k 408 (i.e. due to “catastrophic forgetting”).

15

In use, the system for machine translation (e.g. see system 100 described with reference to FIG. 1) may perform machine translations for various language pairs, e.g. Chinese-English and Malay-English.

20

In an example experiment, the performance of an ensemble model for translating Chinese to English and English to Chinese, and the performance of an ensemble model for translating from Malay to English and English to Malay are assessed respectively.

25

In the example experiment, before machine translations are performed, a baseline model and at least one adapted model are trained and are used to construct an ensemble model. This may be performed according to framework 300 of FIG. 3. In the exemplary embodiment, the training dataset (for training models) may include at least two datasets, namely, a first dataset (may be referred to as baseline training data or legacy data) to train the baseline model, and a second dataset (may be referred to as ensemble training
30 data) to train the at least one adapted model. The trained baseline model and the trained at least one adapted model are ensembled to construct an ensemble model. In the example experiment, the performance of the ensemble model (for English and Chinese pair and for English and Malay pair; i.e. in four language directions) is compared to the

performance of a single trained baseline model (used stand-alone) and a single trained adapted model (used stand-alone).

Baseline Data

5

In the experiment, for training, the first dataset may comprise data of varying quality and domain.

For training a baseline model for Chinese to English translation and English to Chinese translation, baseline training data including 25 million sentence pairs of Chinese and English sentences are obtained (e.g. having passed basic quality assurance). Separately, evaluation test set(s) comprising a plurality of sentences with human annotated references for all four translation directions (Chinese to English, English to Chinese, Malay to English and English to Malay) may also be obtained.

15

In the experiment, for training a baseline model for Malay to English translation, baseline training data including 5.73 million sentence pairs of Malay and English sentences are obtained. Separately, evaluation test set(s) comprising 500 sentences with four human annotated references are also created.

20

In the experiment, for training a baseline model for English to Malay translation, baseline training data including 6.76 million sentence pairs of English and Malay sentences are obtained.

25

In the experiment, the dataset for training a baseline model for Malay-to-English and English-to-Malay translation contain Malay-English sentence pairs available to the inventors.

Ensemble Data

30

In the experiment, for training at least one adapted model for Chinese-to-English and English-to-Chinese translation, 14 thousand sentence pair data are collected/obtained to construct ensemble training data. The collected/obtained data are split into 5 datasets, each dataset containing about 2.8 thousand sentence pairs. The

datasets may be further combined into a plurality of incremental datasets, with one incremental dataset having more/increasing data than another.

As an example, the collected/obtained data are split into 5 equal parts, e.g. sub1, sub2, sub3, sub4, sub5. Ensemble training data may then be constructed, e.g. EnsembleData1, EnsembleData2, EnsembleData3, EnsembleData4, EnsembleData5, by assigning sub1 to EnsembleData1; sub1 and sub2 to EnsembleData2; sub1, sub2, and sub3 to EnsembleData3; sub1, sub2, sub3, and sub4 to EnsembleData4; and all sub1 to sub5 to EnsembleData5. Therefore, there can be provided a plurality of incremental datasets with one incremental dataset having more/increasing data than another, e.g. compare EnsembleData1 with sub1 against EnsembleData2 with sub1 and sub2.

The at least one adapted model may be trained with the incremental datasets for the accumulative effect (compare framework 300 of FIG. 3). The inventors recognise that if each adapted model is only trained on a separate subset (non-accumulated) of the full ensemble data, all the adapted models are to be ensembled in order to capture the respective new knowledge learned from each adapted model. However, the inventors recognise that such ensembling is likely to be expensive in terms of both time and space with an increasing number of models to be ensembled. Therefore, the inventors recognise that it is significantly useful to adapt accumulatively on the full ensemble data which can capture all the new knowledge embedded within the accumulated data and then perform ensembling with the baseline model.

In the experiment, for training at least one adapted model for Malay-to-English and English-to-Malay translation, 47 thousand sentence pair data are collected/obtained to construct ensemble training data. The collected/obtained data may then be split into 5 data sets, each dataset containing about 9.4 thousand sentence pairs. The datasets may be further combined into a plurality of incremental datasets, with one incremental dataset having more/increasing data than another. See the example above described in the context of constructing ensemble training data for training at least one adapted model for Chinese-to-English and English-to-Chinese translation.

It is noted that the baseline data are usually on the order of millions (for high resource language pairs) as compared to the ensemble data that is usually on the order of thousands.

5 Training and Performance

In the experiment, after the baseline model and the at least one first adapted model are trained using the baseline training data and the ensemble training data respectively to construct an ensemble model (e.g. see framework 300 of FIG. 3), the performance of the ensemble model may be evaluated. In the experiment, for this purpose, the translation performance at each epoch (or a pass) of a baseline model (stand-alone), an adapted model (stand-alone) (baseline model adapted to new data), and the ensemble model are compared.

In the experiment, for evaluating the performance of Chinese-to-English and English-to-Chinese translations, 14 thousand new parallel Chinese-English data are used for adaptation to compare the performance using the average BLEU (BiLingual Evaluation Understudy) score on three Chinese-English evaluation test sets. The test sets comprise a plurality of source sentences with human annotated reference translations. It will be appreciated that other number of test sets may also be used. In the experiment, the baseline model is first trained by using the 25 million parallel data and then adapted on the new data (i.e. 14 thousand new data) for 11 epochs. 11 checkpoints are thus obtained. The comparison results (average performance using the three test sets) for Chinese-to-English translation and English-to-Chinese translation are shown in FIG. 5 and FIG. 6 respectively.

In the experiment, for evaluating the performance of Malay-to-English and English-to-Malay translations, 47 thousand new parallel Malay-English data are used for adaptation to compare the performance using the average BLEU score on the three Malay-English evaluation test sets. It will be appreciated that other number of test sets may also be used. In the experiment, the baseline model is first trained by using baseline parallel data and then adapted on the new data (i.e. 47 thousand new data) for 11 epochs. 11 checkpoints are thus obtained. The comparison results (average performance using

the three test sets) for Malay-to-English translation and English-to-Malay translation are shown in FIG. 7 and FIG. 8 respectively.

5 In FIGs. 5 to 8, each Baseline plot (502, 602, 702, 802) in dotted lines refers to the performance of the baseline model.

10 In FIGs. 5 to 8, each Single-Checkpoint plot (504, 604, 704, 804) refers to the performance of the adapted model (i.e. simple adaptation of the baseline model using the new data). In the experiment, based on the performance seen from the Single-Checkpoint plots (504, 604, 704, 804) shown in FIGs. 5 to 8, it is observed that simple adaptation resulting in the adapted model may make the models' performance significantly unstable. For example, the Single-Checkpoint plot 704 of FIG. 7 shows a wild swing in performance. For example, the Single-Checkpoint plots 504 of FIG. 5 and 604 of FIG. 6 show a large under-performance as compared to the baseline model and the ensemble model.

15

In FIGs. 5 to 8, each Ensemble plot (506, 606, 706, 806) refers to the performance of the ensemble model, which is an ensemble of the baseline model with each adapted model (of the respective different language pairs).

20 In the experiment, based on the performance of the Ensemble plots (506, 606, 706, 806) shown in FIGs. 5 to 8, it is shown that the ensemble models in most cases can achieve better performance than the single adapted model (see e.g. Ensemble plot 506 against Single-Checkpoint plot 504; Ensemble plot 606 against Single-Checkpoint plot 604; Ensemble plot 706 against Single-Checkpoint plot 704 in FIGs. 5 to 8). It is also
25 shown that the ensemble models in most cases have higher probability to perform equal to or significantly better than the baseline model (see e.g. Ensemble plot 506 against Baseline plot 502 (similar performances); Ensemble plot 606 against Baseline plot 602; Ensemble plot 706 against Baseline plot 702; Ensemble plot 806 against Baseline plot 802 in FIGs. 5 to 8).

30

In the experiment, from the performance evaluation performed based on an incremental learning process, for Chinese-to-English translations, the checkpoint (or the relevant model) that had the highest average BLEU score on the three evaluation test sets can be deployed for the machine translation system. In the experiment, if there is a

separate evaluation test set provided / made available (e.g. a user-specific test set), a checkpoint may be deployed on the condition that the performance using the user-specific test set should not be lower than the performance by a baseline model (stand-alone).

5 In the experiment, for Malay-to-English translation, the checkpoint that has the highest BLEU score on the test sets is deployed. The checkpoint may be deployed for the system also on the condition that its performance is higher than the performance by a baseline model (stand-alone). In the experiment, if no checkpoint satisfies the above requirements, the baseline model is used instead of an ensemble model.

10

 With the described exemplary embodiments and examples, the inventors have recognised that the ensemble approach is a method for neural networks that involve training multiple models, applying the trained models to test data, and averaging the outputs. In described exemplary embodiments, this method is applied both in an
15 incremental learning task and for machine translation, wherein a baseline model is trained using a first dataset (a large dataset) and at least one adapted model is trained using a second dataset (a relatively small dataset compared to the large dataset). In the exemplary embodiments, this method can be applied to various language pairs, such as Chinese-English and Malay-English, under an industry environment (with the baseline
20 model training dataset typically being much larger). In an experiment, it is observed that this method may be usefully applied to incremental learning tasks for machine translation.

 In one example described herein, there can be provided a method of incremental learning for machine translation, the method may comprise using an ensemble
25 method/process for two or more models; training at least one baseline model using a large dataset (with reference to training a baseline model using baseline data); training at least one adapted model using a smaller new training dataset as compared to the large dataset size for training the at least one baseline model (with reference to training one or more adapt models using ensemble data); wherein the smaller dataset comprises data
30 obtained over time; and the datasets (large and smaller) comprise data of the same language pair for language translation.

 In a further example, the smaller datasets comprising data processed into a plurality of parts are further combined into two or more incremental datasets, with one

incremental dataset having more/increasing data than another (with reference to constructing ensemble data comprising datasets with increasing number of data).

5 In the described exemplary embodiments, references to a language input are intended to cover the language input being in text form. Further, although a language input is exemplarily described as one of Chinese, Malay, or English languages, it will be appreciated that the exemplary embodiments are not limited to these languages and other languages may be used.

10 In the described exemplary embodiments, the first dataset may be a legacy dataset and/or a large dataset. The second dataset may be (or a part) of an ensemble dataset and/or a smaller dataset (compared to the large dataset).

15 In the described exemplary embodiments, references have been made to the phrase "a part of". It will be appreciated that the exemplary embodiments are not limited to a conventional understanding of "a part of" and the references made can also include "a full part of", e.g. references to "a part of" an ensemble dataset may refer to "a full part of" the ensemble dataset.

20 In the described exemplary embodiments, references have been made to "newer data". In the described exemplary embodiments, "newer data" may refer to data obtained after an arbitrary amount of time has passed from the establishment of a first dataset. "Newer data" may additionally/alternatively refer to data obtained at a pre-determined interval of time (e.g. T, T being a number) after the establishment of the first dataset. It
25 will be appreciated that other intervals of time (e.g. days, weeks, years) may also be used. In addition, between incremental datasets of ensemble data (or data after the first dataset), an arbitrary amount of time may also have passed between obtaining of different batches for the incremental datasets. For example, referring to FIG. 3, an amount of time may have passed after obtaining New Data (Batch 1) 304 prior to obtaining New Data
30 (Batch 2) 310. That is, the incremental dataset (of New Data (Batch 1) 304 added with New Data (Batch 2) 310) is obtained after an arbitrary amount of time has passed after obtaining New Data (Batch 1) 304. As another example, the latest incremental dataset (of New Data (Batch 1) 304 added with New Data (Batch 2) 310 and added with accumulated data until New Data (Batch k) 316) is obtained after an arbitrary amount of

time has passed after obtaining the incremental dataset of New Data (Batch 1) 304 added with New Data (Batch 2) 310.

5 In the described exemplary embodiments, the baseline model and/or the at least one adapted model are based on a transformer deep learning model. However, it will be appreciated that the exemplary embodiments are not limited as such. For example, one or both of the baseline model and the at least one adapted model may be based on other deep learning models.

10 In the described exemplary embodiments, the ensemble module is described as comprising a baseline model and at least one adapted model. It will be appreciated that any number of adapted models may be comprised with the baseline model in the ensemble module. For example, the described exemplary embodiments are not limited to using one adapted model with the baseline model. The inventors recognise that
15 ensembling an increasing number of models can likely be expensive in terms of both time and space with the increasing number of models to be ensembled. Therefore, it may be significant useful to adapt on a full set of ensemble data which can capture all the new knowledge embedded within the accumulated data and perform an ensemble on the baseline model and a single adapted model.

20

In the description herein, the terms "coupled" or "connected" as used are intended to cover both directly connected or connected through one or more intermediate means, unless otherwise stated.

25 The use of "a", "an" or "the" is intended to mean "one or more" unless it is described specifically to the contrary.

The terms "configured to (perform a task/action)", "configured for (performing a task/action)" and the like as used in this description include being programmable,
30 programmed, connectable, wired or otherwise constructed to have the ability to perform the task/action when arranged or installed as described herein. The terms "configured to (perform a task/action)", "configured for (performing a task/action)" and the like are intended to cover "when in use, the task/action is performed", e.g. specifically to and/or

specifically configured to and/or specifically arranged to and/or specifically adapted to do or perform a task/action.

5 The term "and/or", e.g., "X and/or Y" is understood to mean either "X and Y" or "X or Y" and should be taken to provide explicit support for both meanings or for either meaning. The use of "or" is intended to mean an "inclusive or," and not an "exclusive or" unless it is described specifically to the contrary.

10 The terms "associated with", "related to" and the like used herein when referring to two elements refers to a broad relationship between the two elements. The relationship includes, but is not limited to, a physical, a chemical or a biological relationship. For example, when element A is associated with element B, elements A and B may be directly or indirectly attached to each other or element A may contain element B or vice versa.

15 The terms "exemplary embodiment", "example embodiment", "exemplary implementation", "exemplarily" and the like used herein are intended to indicate an example of matters described in the present disclosure. Such an example may relate to one or more features defined in the claims and is not necessarily intended to emphasise a best example or any essentialness of any features.

20

The description herein may be, in certain portions, explicitly or implicitly described as algorithms and/or functional operations that operate on data within a computer memory or an electronic circuit. These algorithmic descriptions and/or functional operations are usually used by those skilled in the information/data processing arts for efficient description. An algorithm is generally relating to a self-consistent sequence of steps leading to a desired result. The algorithmic steps can include physical manipulations of physical quantities, such as electrical, magnetic or optical signals capable of being stored, transmitted, transferred, combined, compared, and otherwise manipulated.

25
30

Further, unless specifically stated otherwise, and would ordinarily be apparent from the following, a person skilled in the art will appreciate that throughout the present specification, discussions utilizing terms such as "scanning", "calculating", "determining", "replacing", "generating", "initializing", "outputting", and the like, refer to action and

processes of an instructing processor/computer system, or similar electronic circuit/device/component, that manipulates/processes and transforms data represented as physical quantities within the described system into other data similarly represented as physical quantities within the system or other information storage, transmission or display devices etc.

The description also discloses relevant device/apparatus for performing the steps of the described methods. Such apparatus may be specifically constructed for the purposes of the methods, or may comprise a general purpose computer/processor or other device selectively activated or reconfigured by a computer program stored in a storage member. The algorithms and displays described herein are not inherently related to any particular computer or other apparatus. It is understood that general purpose devices/machines may be used in accordance with the teachings herein. Alternatively, the construction of a specialized device/apparatus to perform the method steps may be desired.

In addition, it is submitted that the description also implicitly covers a computer program, in that it would be clear that the steps of the methods described herein may be put into effect by computer code. It will be appreciated that a large variety of programming languages and coding can be used to implement the teachings of the description herein. Moreover, the computer program if applicable is not limited to any particular control flow and can use different control flows without departing from the scope of the invention.

Furthermore, one or more of the steps of the computer program if applicable may be performed in parallel and/or sequentially. Such a computer program if applicable may be stored on any computer readable medium. The computer readable medium may include storage devices such as magnetic or optical disks, memory chips, or other storage devices suitable for interfacing with a suitable reader/general purpose computer. In such instances, the computer readable storage medium is non-transitory. Such storage medium also covers all computer-readable media e.g. medium that stores data only for short periods of time and/or only in the presence of power, such as register memory, processor cache and Random Access Memory (RAM) and the like. The computer readable medium may even include a wired medium such as exemplified in the Internet system, or wireless medium such as exemplified in Bluetooth technology. The computer

readable medium may be, for example, cloud storage in the Internet or within an intranet. The computer program when loaded and executed on a suitable reader effectively results in an apparatus that can implement the steps of the described methods, e.g. in a physical embodiment. The computer readable medium is intended to be transferable and is
5 reproducible in that the computer program if applicable is reproducible.

The exemplary embodiments may also be implemented as hardware modules. A module is a functional hardware unit designed for use with other components or modules. For example, a module may be implemented using digital or discrete electronic
10 components, or it can form a portion of an entire electronic circuit such as an Application Specific Integrated Circuit (ASIC). A person skilled in the art will understand that the exemplary embodiments can also be implemented as a combination of hardware and software modules.

15 Additionally, when describing some embodiments, the disclosure may have disclosed a method and/or process as a particular sequence of steps. However, unless otherwise required, it will be appreciated the method or process should not be limited to the particular sequence of steps disclosed. Other sequences of steps may be possible. The particular order of the steps disclosed herein should not be construed as undue
20 limitations. Unless otherwise required, a method and/or process disclosed herein should not be limited to the steps being carried out in the order written. The sequence of steps may be varied and still remain within the scope of the disclosure.

Further, in the description herein, the word "substantially" whenever used is
25 understood to include, but not restricted to, "entirely" or "completely" and the like. In addition, terms such as "comprising", "comprise", and the like whenever used, are intended to be non-restricting descriptive language in that they broadly include elements/components recited after such terms, in addition to other components not explicitly recited. For an example, when "comprising" is used, reference to a "one" feature
30 is also intended to be a reference to "at least one" of that feature. Terms such as "consisting", "consist", and the like, may, in the appropriate context, be considered as a subset of terms such as "comprising", "comprise", and the like. Therefore, in embodiments disclosed herein using the terms such as "comprising", "comprise", and the like, it will be appreciated that these embodiments provide teaching for corresponding

embodiments using terms such as "consisting", "consist", and the like. Further, terms such as "about", "approximately" and the like whenever used, typically means a reasonable variation, for example a variation of +/- 5% of the disclosed value, or a variance of 4% of the disclosed value, or a variance of 3% of the disclosed value, a
5 variance of 2% of the disclosed value or a variance of 1% of the disclosed value.

Furthermore, in the description herein, certain values may be disclosed in a range. The values showing the end points of a range are intended to illustrate a preferred range. Whenever a range has been described, it is intended that the range covers and teaches
10 all possible sub-ranges as well as individual numerical values within that range. That is, the end points of a range should not be interpreted as inflexible limitations. For example, a description of a range of 1% to 5% is intended to have specifically disclosed sub-ranges 1% to 2%, 1% to 3%, 1% to 4%, 2% to 3% etc., as well as individually, values within that range such as 1%, 2%, 3%, 4% and 5%. It is to be appreciated that the individual
15 numerical values within the range also include integers, fractions and decimals. Furthermore, whenever a range has been described, it is also intended that the range covers and teaches values of up to 2 additional decimal places or significant figures (where appropriate) from the shown numerical end points. For example, a description of a range of 1% to 5% is intended to have specifically disclosed the ranges 1.00% to 5.00%
20 and also 1.0% to 5.0% and all their intermediate values (such as 1.01%, 1.02% ... 4.98%, 4.99%, 5.00% and 1.1%, 1.2% ... 4.8%, 4.9%, 5.0% etc.,) spanning the ranges. The intention of the above specific disclosure is applicable to any depth/breadth of a range.

In the exemplary embodiments, the baseline model target language output and the
25 adapted model target language output are exemplarily described/shown as sentences. It will be appreciated that the exemplary embodiments are not limited as such. That is, for example, the baseline model target language output and/or the adapted model target language output may be in the form of a phrase or even a word/token, and such a phrase / word may be a part / portion of a sentence.

30

FIG. 9 is a schematic flowchart for illustrating a method of incremental learning in machine translation in an exemplary embodiment. One or more steps of the method are computer-implemented. Alternatively, the method is a computer-implemented method.

At step 902, a source language input to be translated into a target language output is obtained. At step 904, an ensemble model module is provided. The ensemble model module comprises a baseline model trained using a first dataset and at least one first adapted model trained using a second dataset, the second dataset being smaller than the first dataset. At step 906, the source language input is received at the baseline model. At step 908, a baseline model target language output with associated one or more baseline model probability scores is generated using the baseline model. At step 910, the source language input is received at the at least one first adapted model. That is, the source language input is received at both the baseline model and the first adapted model. At step 912, a first adapted model target language output with associated one or more first adapted model probability scores is generated using the at least one first adapted model. At step 914, the one or more baseline model probability scores and the one or more first adapted model probability scores are received at a determination module. At step 916, a system target language output based on the received one or more baseline model probability scores and the received one or more first adapted model probability scores is generated using the determination module.

In another exemplary embodiment, there may be provided a computer readable storage medium having stored thereon software instructions that, when executed by a system processing module of a system for machine translation, cause the system processing module to perform the method of FIG. 9. The computer readable storage medium may also store thereon software instructions that, when executed by the system processing module of the system for machine translation, cause the system processing module to perform any of the steps described herein.

25

Different exemplary embodiments can be implemented in the context of data structure, program modules, program and computer instructions executed in a computer implemented environment. A computing environment is briefly disclosed herein. One or more exemplary embodiments may be embodied in one or more computer systems, such as is schematically illustrated in FIG. 10.

30

One or more exemplary embodiments may be implemented as software, such as a computer program being executed within a computer system 1000, and instructing the computer system 1000 to conduct a method of an exemplary embodiment.

The computer system 1000 comprises a computer unit 1002, input modules such as a keyboard 1004 and a pointing device 1006 and a plurality of output devices such as a display 1008, and printer 1010. A user can interact with the computer unit 1002 using
5 the above devices. The pointing device can be implemented with a mouse, track ball, pen device or any similar device. One or more other input devices (not shown) such as a joystick, game pad, satellite dish, scanner, touch sensitive screen or the like can also be connected to the computer unit 1002. The display 1008 may include a cathode ray tube (CRT), liquid crystal display (LCD), field emission display (FED), plasma display or any
10 other device that produces an image that is viewable by the user.

The computer unit 1002 can be connected to a computer network 1012 via a suitable transceiver device 1014, to enable access to e.g. the Internet or other network systems such as Local Area Network (LAN) or Wide Area Network (WAN) or a personal
15 network. The network 1012 can comprise a server, a router, a network personal computer, a peer device or other common network node, a wireless telephone or wireless personal digital assistant. Networking environments may be found in offices, enterprise-wide computer networks and home computer systems etc. The transceiver device 1014 can be a modem/router unit located within or external to the computer unit 1002, and may
20 be any type of modem/router such as a cable modem or a satellite modem.

It will be appreciated that network connections shown are exemplary and other ways of establishing a communications link between computers can be used. The existence of any of various protocols, such as TCP/IP, Frame Relay, Ethernet, FTP,
25 HTTP and the like, is presumed, and the computer unit 1002 can be operated in a client-server configuration to permit a user to retrieve web pages from a web-based server. Furthermore, any of various web browsers can be used to display and manipulate data on web pages.

30 The computer unit 1002 in the example comprises a processor 1018, a Random Access Memory (RAM) 1020 and a Read Only Memory (ROM) 1022. The ROM 1022 can be a system memory storing basic input/output system (BIOS) information. The RAM 1020 can store one or more program modules such as operating systems, application programs and program data.

The processor 1018 may be a processing module of described exemplary embodiments. For example, compare the processing module 108 of FIG. 1. The RAM 1020 and/or the ROM 1022 may be used to store instructions/code executable by the processor 1018 e.g. to implement the functions of an ensemble model module (e.g. ensemble model module 104 of FIG. 1) and/or a determination module (e.g. determination module 106 of FIG. 1).

The computer unit 1002 further comprises a number of Input/Output (I/O) interface units, for example I/O interface unit 1024 to the display 1008, and I/O interface unit 1026 to the keyboard 1004. The components of the computer unit 1002 typically communicate and interface/couple connectedly via an interconnected system bus 1028 and in a manner known to the person skilled in the relevant art. The bus 1028 can be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures.

It will be appreciated that other devices can also be connected to the system bus 1028. For example, a universal serial bus (USB) interface can be used for coupling a video or digital camera to the system bus 1028. An IEEE 1394 interface may be used to couple additional devices to the computer unit 1002. Other manufacturer interfaces are also possible such as FireWire developed by Apple Computer and i.Link developed by Sony. Coupling of devices to the system bus 1028 can also be via a parallel port, a game port, a PCI board or any other interface used to couple an input device to a computer. It will also be appreciated that, while the components are not shown in the figure, sound/audio can be recorded and reproduced with a microphone and a speaker. A sound card may be used to couple a microphone and a speaker to the system bus 1028. It will be appreciated that several peripheral devices can be coupled to the system bus 1028 via alternative interfaces simultaneously.

The system bus 1028 may be used to couple to an input module (e.g. input module 102 of FIG. 1) to receive a source input language to be translated into a target language output. The input module may for example be a physical wired link from a web server, or a wireless link from a web server, or the keyboard 1004 etc.

An application program can be supplied to the user of the computer system 1000 being encoded/stored on a data storage medium such as a CD-ROM or flash memory carrier. The application program can be read using a corresponding data storage medium drive of a data storage device 1030. The data storage medium is not limited to being portable and can include instances of being embedded in the computer unit 1002. The data storage device 1030 can comprise a hard disk interface unit and/or a removable memory interface unit (both not shown in detail) respectively coupling a hard disk drive and/or a removable memory drive to the system bus 1028. This can enable reading/writing of data. Examples of removable memory drives include magnetic disk drives and optical disk drives. The drives and their associated computer-readable media, such as a floppy disk provide nonvolatile storage of computer readable instructions, data structures, program modules and other data for the computer unit 1002. It will be appreciated that the computer unit 1002 may include several of such drives. Furthermore, the computer unit 1002 may include drives for interfacing with other types of computer readable media.

The application program is read and controlled in its execution by the processor 1018. Intermediate storage of program data may be accomplished using RAM 1020. The method(s) of the exemplary embodiments can be implemented as computer readable instructions, computer executable components, or software modules. One or more software modules may alternatively be used. These can include an executable program, a data link library, a configuration file, a database, a graphical image, a binary data file, a text data file, an object file, a source code file, or the like. When one or more computer processors execute one or more of the software modules, the software modules interact to cause one or more computer systems to perform according to the teachings herein.

In one example, baseline training data (for training a baseline model) and newer ensemble data e.g. comprising incremental datasets (for training at least one adapted model) may be provided with or stored within the data storage device 1030. A trained baseline model and/or a trained at least one adapted model may be provided with or stored within the data storage device 1030.

The operation of the computer unit 1002 can be controlled by a variety of different program modules. Examples of program modules are routines, programs, objects,

components, data structures, libraries, etc. that perform particular tasks or implement particular abstract data types. The exemplary embodiments may also be practiced with other computer system configurations, including handheld devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, personal digital assistants, mobile telephones and the like. Furthermore, the exemplary embodiments may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a wireless or wired communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

It will be appreciated by a person skilled in the art that other variations and/or modifications may be made to the specific embodiments without departing from the scope of the claimed invention as broadly described. For example, in the description herein, features of different exemplary embodiments may be mixed, combined, interchanged, incorporated, adopted, modified, included etc. or the like across different exemplary embodiments. For example, exemplary embodiments are not necessarily mutually exclusive as some may be combined with one or more embodiments to form new exemplary embodiments. Furthermore, it will be appreciated that while the present disclosure provides embodiments having one or more of the features/characteristics discussed herein, one or more of these features/characteristics may also be disclaimed in other alternative embodiments and the present disclosure provides support for such disclaimers and these associated alternative embodiments. The present embodiments are, therefore, to be considered in all respects to be illustrative and not restrictive.

CLAIMS

1. A system for incremental learning in machine translation, the system comprising
- 5 an input module for obtaining a source language input to be translated into a target language output;
- an ensemble model module coupled to the input module, the ensemble model module comprising
- 10 a baseline model trained using a first dataset, the baseline model arranged to receive the source language input to generate a baseline model target language output with associated one or more baseline model probability scores;
- at least one first adapted model trained using a second dataset, the second dataset being smaller than the first dataset, the at least one first adapted model arranged to receive the source language input to generate a first adapted model
- 15 target language output with associated one or more first adapted model probability scores; and
- a determination module coupled to the ensemble model module for receiving the one or more baseline model probability scores and the one or more first adapted model probability scores, the determination module arranged to generate a system target
- 20 language output based on the received one or more baseline model probability scores and the received one or more first adapted model probability scores.
2. The system as claimed in claim 1, further comprising the at least one first adapted model being selected from a plurality of adapted models, the plurality of adapted
- 25 models comprising at least a second adapted model trained using a third dataset, the third dataset being different in size from the second dataset used to train the at least one first adapted model.
3. The system as claimed in claim 2, wherein one of the second dataset and
- 30 the third dataset is a subset of the other of the second dataset and the third dataset.
4. The system as claimed in claims 2 or 3, wherein the second dataset and the third dataset are each a part of an ensemble dataset.

5. The system as claimed in claim 1, wherein the second dataset comprises newer data obtained over time after establishment of the first dataset.

6. The system as claimed in any one of claims 1 to 5, wherein the first dataset
5 and the second dataset each comprises language pair data of a source language and a target language.

7. The system as claimed in any one of claims 1 to 5, wherein the
determination module is arranged to perform an averaging operation based on the
10 received one or more baseline model probability scores and the received one or more
first adapted model probability scores to generate the system target language output.

8. The system as claimed in any one of claims 1 to 7, wherein the baseline
model and the at least one first adapted model are each based on a transformer deep
15 learning model.

9. The system as claimed in any one of claims 1 to 8, further comprising an
output module coupled to the determination module, the output module being arranged
to receive the system target language output from the determination module and to
20 provide the system target language output to a user.

10. A method of incremental learning in machine translation, the method
comprising
obtaining a source language input to be translated into a target language output;
25 providing an ensemble model module, the ensemble model module comprising
a baseline model trained using a first dataset and at least one first adapted
model trained using a second dataset, the second dataset being smaller than the
first dataset;
receiving, at the baseline model, the source language input;
30 generating, using the baseline model, a baseline model target language output
with associated one or more baseline model probability scores;
receiving, at the at least one first adapted model, the source language input;

generating, using the at least one first adapted model, a first adapted model target language output with associated one or more first adapted model probability scores;

receiving at a determination module the one or more baseline model probability scores and the one or more first adapted model probability scores; and

generating, using the determination module, a system target language output based on the received one or more baseline model probability scores and the received one or more first adapted model probability scores.

11. The method as claimed in claim 10, further comprising selecting the at least one first adapted model from a plurality of adapted models, the plurality of adapted models comprising at least a second adapted model trained using a third dataset, wherein the third dataset is different in size from the second dataset used to train the at least one first adapted model.

15

12. The method as claimed in claim 11, wherein one of the second dataset and the third dataset is a subset of the other of the second dataset and the third dataset.

13. The method as claimed in claims 11 or 12, wherein the second dataset and the third dataset are each a part of an ensemble dataset.

14. The method as claimed in claim 10, further comprising obtaining newer data over time after establishment of the first dataset to form data of the second dataset.

15. The method as claimed in any one of claims 10 to 14, wherein the first dataset and the second dataset each comprises language pair data of a source language and a target language.

16. The method as claimed in any one of claims 10 to 15, further comprising performing at the determination module an averaging operation based on the received one or more baseline model probability scores and the received one or more first adapted model probability scores to generate the system target language output.

17. The method as claimed in any one of claims 10 to 16, wherein the baseline model and the at least one first adapted model are each based on a transformer deep learning model.

5 18. The method as claimed in any one of claims 10 to 17, further comprising coupling an output module to the determination module;
receiving the system target language output from the determination module; and
providing the system target language output to a user using the output module.

10 19. A computer readable storage medium having stored thereon software instructions that, when executed by a system processing module of a system for incremental learning in machine translation, cause the system processing module to perform a method of any one of claims 10 to 18.

15