CLAIMS

What is claimed is:

1. A hierarchical machine learning system implemented by one or more computers comprising:

a storage device; and

a processing device implementing a master artificial intelligence (AI) engine to:

receive, from each of one or more real-time AI engines, a machine learning algorithm, parameters associated with the machine learning algorithm, and features employed to train the parameters;

receive, from each of the one or more real-time AI engines, labeled data used to calculate the features; and

construct, based on a combination rule, a master machine learning model using the features, the machine learning algorithm, and the parameters associated with the machine learning algorithm received from each of the one or more real-time AI engines, wherein each of the one or more real-time AI engine is to:

responsive to receiving a document, parse the document to generate a document object model (DOM) tree, wherein each node of the DOM tree comprises one of a sentence or a paragraph of the document;

tokenize each node of the DOM tree to generate one or more tokens;

present, on a user interface, the document including the one or more tokens;

label, based on user actions on the user interface, at least one of the one or more the tokens in the DOM tree as one of a strong positive, a strong negative, or one of a weak positive or a weak negative token; and

provide the DOM tree including the labeled tokens to train a machine learning algorithm.

2. The system of claim 1, wherein each of the one or more real-time AI engines is further to:

responsive to detecting a first action on a first GUI element to positively mark a first token, label the first token as a strong positive token in the DOM tree and assign the strong positive token with a first weight value;

responsive to detecting a second action on a second GUI element to negatively mark a second token, label the second token as a strong negative token in the DOM tree and assign the strong negative token with a second weight value;

predict a third token as a weak negative token based on the strong positive token, wherein the weak negative token is at least one of spatially close to the strong positive token in the document or unmarked but identical to the strong positive token, and assign the weak negative token with a third weight value; and

label a fourth token as a weak positive token, wherein the fourth token is identified by the machine learning model as a candidate positive token, and assign the weak positive token with a fourth weight value.

3. The system of claim 2, wherein each of the one or more real-time AI engines is further to:

calculate the first features using the labeled tokens, wherein the first features comprise at least one a format feature or a natural language processing (NLP) feature;

reduce the first features to the features, wherein the features are a subset of the first features;

select the machine learning algorithm based on the labeled tokens; and

train the machine learning algorithm using the features to adjust the parameters associated with the machine learning algorithm taking into account of different weight values associated with different types of the labeled tokens.

4. The system of claim 3, wherein each of the one or more real-time AI engines is to:

provide the machine learning algorithm, the parameters associated with the machine learning algorithm, and the features to the master AI engine;

provide the labeled tokens as the labeled data to the master AI engine; and

provide unlabeled data to the master AI engine.

5. The system of claim 4, wherein to construct, based on a combination rule, a master machine learning model using the features, the machine learning algorithm, and the parameters associated with the machine learning algorithm received from each of the one or more real-time AI engines, the processing device is to:

label each of the unlabeled data as a weak negative;

calculate a respective engine weight value associated with each of the one or more real-time AI engines, wherein the engine weight value is calculated based on:

applying all of the labeled data and unlabeled data received from the one or more real-time AI engines to each of the one or more real-time AI engines to calculate a respective score; and

calculating the respective engine weight value as a ratio between the respective score and a sum of all scores; and

determine the master machine learning model as a weighted average of the one or more real-time AI engines by applying the engine weight value to the corresponding real-time AI engine.

6. The system of claim 5, wherein the processing device implementing a master artificial intelligence (AI) engine further to:

apply the master machine learning model to the labeled data received from the one or more real-time AI engines to determine updated labels for the labeled data;

determine, based on the updated labels, whether the labeled data are correctly labeled or incorrectly labeled; and

provide a feedback comprising at least one of the determined correctly labeled or and incorrectly labeled data to the one or more real-time AI engines for a further training of the one or more real-time AI engines.

7. The system of claim 6, wherein the processing device is further to:

receive, from each of further trained one or more real-time AI engines, an updated machine learning algorithm, updated parameters associated with the updated machine learning algorithm, and updated features employed to train the parameters;

receive the correctly labeled data used to train the updated parameters associated with the machine learning algorithm;

construct, based on the combination rule, a new version of the master machine learning model using the updated features, the updated machine learning algorithm, and the updated parameters associated with the updated machine learning algorithm;

assemble a global master machine learning model using different versions of the machine learning model according to an assemble rule; and

provide the global machine learning model to a machine learning task.

8. The system of claim 1, wherein the processing device is further to provide the master machine learning model to a machine learning task.

9. A method of hierarchical machine learning, the method comprising:

receiving, by a processing device from each of one or more real-time AI engines, a machine learning algorithm, parameters associated with the machine learning algorithm, and features employed to train the parameters;

receiving, from each of the one or more real-time AI engines, labeled data used to calculate the features; and

constructing, based on a combination rule, a master machine learning model using the features, the machine learning algorithm, and the parameters associated with the machine learning algorithm received from each of the one or more real-time AI engines,

wherein each of the one or more real-time AI engines is to perform:

responsive to receiving a document, parsing the document to generate a document object model (DOM) tree, wherein each node of the DOM tree comprises one of a sentence or a paragraph of the document;

tokenizing each node of the DOM tree to generate one or more tokens;

presenting, on a user interface, the document including the one or more tokens;

labeling, based on user actions on the user interface, at least one of the one or more the tokens in the DOM tree as one of a strong positive, a strong negative, or one of a weak positive or a weak negative token; and

providing the DOM tree including the labeled tokens to train a machine learning algorithm.

10. The method of claim 9, wherein each of the one or more real-time AI engine is further to perform:

responsive to detecting a first action on a first GUI element to positively mark a first token, labeling the first token as a strong positive token in the DOM tree and assign the strong positive token with a first weight value;

responsive to detecting a second action on a second GUI element to negatively mark a second token, labeling the second token as a strong negative token in the DOM tree and assign the strong negative token with a second weight value;

predicting a third token as a weak negative token based on the strong positive token, wherein the weak negative token is at least one of spatially close to the strong positive token in the document or unmarked but identical to the strong positive token, and assign the weak negative token with a third weight value; and

labeling a fourth token as a weak positive token, wherein the fourth token is identified by the machine learning model as a candidate positive token, and assign the weak positive token with a fourth weight value.

11. The method of claim 10, wherein each of the one or more real-time AI engines is further to perform:

calculating the first features using the labeled tokens, wherein the first features comprise at least one a format feature or a natural language processing (NLP) feature;

reducing the first features to the features, wherein the features are a subset of the first features;

selecting the machine learning algorithm based on the labeled tokens; and

training the machine learning algorithm using the features to adjust the parameters associated with the machine learning algorithm taking into account of different weight values associated with different types of the labeled tokens.

12. The method of claim 11, wherein each of the one or more real-time AI engines is further to perform:

providing the machine learning algorithm, the parameters associated with the machine learning algorithm, and the features to the master AI engine;

providing the labeled tokens as the labeled data to the master AI engine; and

providing unlabeled data to the master AI engine.

13. The method of claim 12, wherein constructing, based on a combination rule, a master machine learning model using the features, the machine learning algorithm, and the parameters associated with the machine learning algorithm received from each of the one or more real-time AI engines, further comprises:

labeling each of the unlabeled data as a weak negative;

calculating a respective engine weight value associated with each of the one or more real-time AI engines, wherein the engine weight value is calculated based on:

applying all of the labeled data and unlabeled data received from the one or more real-time AI engines to each of the one or more real-time AI engines to calculate a respective score; and

calculating the respective engine weight value as a ratio between the respective score and a sum of all scores; and

determining the master machine learning model as a weighted average of the one or more real-time AI engines by applying the engine weight value to the corresponding real-time AI engine.

14. The method of claim 13, further comprising:

applying the master machine learning model to the labeled data received from the one or more real-time AI engines to determine updated labels for the labeled data;

determining, based on the updated labels, whether the labeled data are correctly labeled or incorrectly labeled; and

providing a feedback comprising at least one of the determined correctly labeled data or incorrectly labeled data to the one or more real-time AI engines for a further training of the one or more real-time AI engines.

15. The method of claim 14, further comprising:

receiving, from each of further trained one or more real-time AI engines, an updated machine learning algorithm, updated parameters associated with the updated machine learning algorithm, and updated features employed to train the parameters;

receiving the correctly labeled data used to train the updated parameters associated with the machine learning algorithm;

constructing, based on the combination rule, a new version of the master machine learning model using the updated features, the updated machine learning algorithm, and the updated parameters associated with the updated machine learning algorithm;

assembling a global master machine learning model using different versions of the machine learning model according to an assemble rule; and

providing the global machine learning model to a machine learning task.

16. The method of claim 9, further comprising providing the master machine learning model to a machine learning task.

17. A machine-readable non-transitory storage media encoded with instructions that, when executed by one or more computers, cause the one or more computer to implement a system for real-time machine learning, to perform operations of any of claims 1 to 16.