CLAIMS

What is claimed is:

1. A method for generating word vectors, comprising:

obtaining words by segmenting a corpus;

establishing a feature vector for each obtained word based, at least in part, on one or more n-ary characters corresponding to the obtained word, wherein each n-ary character represents n consecutive characters of a word;

training a convolutional neural network based, at least in part, on the feature vectors of the obtained words and the feature vectors of context words associated with each obtained word in the corpus by:

traversing the corpus after word segmentation, and performing the following actions responsive to a current word being accessed during the traversing:

determining one or more context words and one or more negative sample words associated with the current word in the corpus, wherein the one or more context words are all words other than the current word in a sliding window centered at~~determined based on a location of~~ the current word in the corpus and wherein the one or more negative sample words are selected in accordance with one or more rules for neural network training convergence;

inputting feature vectors of the context words associated with the current word into a convolutional layer of the convolutional neural network for convolutional calculation;

inputting a result of the convolutional calculation into a pooling layer of the convolutional neural network for pooling calculation to obtain a first vector;

inputting a feature vector of the current word into a full connection layer of the convolutional neural network for calculation to obtain a second vector; and

inputting a feature vector of a negative sample word associated with the current word into the full connection layer of the convolutional neural network for calculation to obtain a third vector; and

updating parameters of the convolutional neural network based, at least in part, on the first vector, the second vector, the third vector, and a specified loss function; and

generating a word vector for each obtained word based, at least in part, on the feature vector of the obtained word and the trained convolutional neural network.

2. The method according to claim 1, wherein characters of an obtained word comprise each character constituting the obtained word, and mark characters added to one or more of a start position and an end position of the obtained word.

3. The method according to claim 1, wherein the establishing the feature vector for each obtained word based, at least in part, on one or more n-ary characters corresponding to the obtained word comprises:

determining a total number of distinct n-ary characters in a collection of respective n-ary characters corresponding to each of the obtained words; and

establishing, for each obtained word, a feature vector whose dimensionality is determined based, at least in part, on the total number.

4. The method according to claim 3, wherein the number of dimensions of the feature vector is the total number.

5. The method according to claim 3, wherein values are assigned to individual dimensions of the feature vector to indicate a mapping between n-ary characters and the obtained word.

6. The method according to claim 1, further comprising performing the convolutional calculation, at least in part, by concatenating the feature vectors of a subset of the context words associated with the current word.

7. The method according to claim 1, further comprising performing the pooling calculation by:

performing at least one of a maximum pooling calculation or an average pooling calculation.

8. The method according to claim 1, wherein updating the parameters of the convolutional neural network based, at least in part, on the first vector, the second vector, the third vector, and the specified loss function comprises:

calculating a first similarity between the second vector and the first vector, and a second similarity between the third vector and the first vector; and

updating the parameters of the convolutional neural network based, at least in part, on the first similarity, the second similarity, and the specified loss function.

9. The method according to claim 1, wherein the loss function includes at least a weight parameter of the convolutional layer, an offset parameter of the convolutional layer, a weight parameter of the full connection layer, an offset parameter of the full connection layer, a hyperparameter, and a similarity calculation function.

10. The method according to claim 1, wherein the generating a word vector for each obtained word based, at least in part, on the feature vector of the obtained word and the trained convolutional neural network comprises:

inputting the feature vector of each obtained word into the full connection layer of the trained convolutional neural network to obtain a vector output.

11. An apparatus for generating word vectors, comprising:

an acquisition module, configured to obtain words by segmenting a corpus;

an establishment module, configured to establish a feature vector for each obtained word based, at least in part, on one or more n-ary characters corresponding to the obtained word, wherein each n-ary character indicates n consecutive characters of a word;

a training module, configured to train a convolutional neural network based, at least in part, on the feature vectors of the obtained words and the feature vectors of context words associated with each obtained word in the corpus by:

traversing the corpus after word segmentation, and performing the following actions responsive to a current word being accessed during the traversal:

determining one or more context words and one or more negative sample words associated with the current word in the corpus, wherein the one or more context words are all words other than the current word in a sliding window centered at ~~determined based on a location of~~ the current word in the corpus and wherein the one or more negative sample words are selected in accordance with one or more rules for neural network training convergence;

inputting feature vectors of the context words associated with the current word into a convolutional layer of the convolutional neural network for convolutional calculation;

inputting a result of the convolutional calculation into a pooling layer of the convolutional neural network for pooling calculation to obtain a first vector;

inputting a feature vector of the current word into a full connection layer of the convolutional neural network for calculation to obtain a second vector; and

inputting a feature vector of a negative sample word associated with the current word into the full connection layer of the convolutional neural network for calculation to obtain a third vector; and

updating parameters of the convolutional neural network based, at least in part, on the first vector, the second vector, the third vector, and a specified loss function; and

a generation module, configured to generate a word vector for each obtained word based on the feature vector of the obtained word and the trained convolutional neural network.

12. The apparatus according to claim 11, wherein characters of an obtained word comprise each character constituting the obtained word, and mark characters added to a start position and/or an end position of the obtained word.

13. The apparatus according to claim 11, wherein the establishing, by the establishment module, of a feature vector for each obtained word based, at least in part, on one or more n-ary characters corresponding to the obtained word comprises:

determining, by the establishment module, a total number of distinct n-ary characters in a collection of respective n-ary characters corresponding to each of the obtained words; and

establishing, for each obtained word, a feature vector whose dimensionality is determined based, at least in part, on the total number.

14. The apparatus according to claim 13, wherein the number of dimensions of the feature vector is the total number.

15. The apparatus according to claim 13, wherein values are assigned to individual dimensions of the feature vector to indicate a mapping between n-ary characters and the obtained word.

16. The apparatus according to claim 11, wherein performing the convolutional calculation by the training module comprises:

performing, by the training module, the convolutional calculation according to the following formula:

$$\tilde{x}_i = x_{i:i+\theta-1} = \left[x_i^T, x_{i+1}^T, \ldots, x_{i+\theta-1}^T\right]^T$$

$$y_i = \sigma(\omega \tilde{x}_i + \zeta)$$

wherein x_i indicates a feature vector of the i^{th} context word; $x_{i:i+\theta-1}$ indicates a vector that is obtained by concatenating the feature vectors of the i^{th} to the $i+\theta-1^{\text{th}}$ context words; y_i indicates the i^{th} element in the vector that is obtained through the convolutional calculation; ω indicates one or more weight parameters of the convolutional layer; ζ indicates one or more offset parameters of the convolutional layer; σ indicates an activation function.

17. The apparatus according to claim 11, wherein performing the convolutional calculation by the training module comprises:

performing, by the training module, at least one of a maximum pooling calculation or an average pooling calculation.

18. The apparatus according to claim 11, wherein the updating, by the training module, of the parameters of the convolutional neural network based, at least in part, on the first vector, the second vector, the third vector, and the specified loss function comprises:

calculating, by the training unit, a first similarity between the second vector and the first vector, and a second similarity between the third vector and the first vector; and

updating the parameters of the convolutional neural network based, at least in part, on the first similarity, the second similarity, and the specified loss function.

19. The apparatus according to claim 11, wherein the loss function comprises:

$$l(w, c; \omega, \zeta, \varsigma, \tau) = log\left(1 + \sum_{m=1}^{\lambda} exp\left(-\gamma \cdot \left(s(w, c) - s(w'_m, c)\right)\right)\right)$$

wherein c indicates the first vector; w indicates the second vector; w'_m indicates the third vector corresponding to the m^{th} negative sample word; ω indicates one or more weight parameters of the convolutional layer; ζ indicates one or more offset parameters of the convolutional layer; ς indicates one or more weight parameters of the full connection layer; τ indicates one or more offset parameters of the full connection layer; γ indicates a hyperparameter; s indicates a similarity calculation function; and λ indicates the number of negative sample words.

20. The apparatus according to claim 11, wherein the generating, by the generation module, of a word vector for each obtained word based, at least in part, on the feature vector of the obtained word and the trained convolutional neural network comprises:

inputting, by the generation module, the feature vector of each obtained word into the full connection layer of the trained convolutional neural network to obtain a vector output.

21. A method for generating word vectors, comprising:

Act 1: establishing a vocabulary of words obtained by segmenting a corpus, wherein the obtained words exclude a word that appears for less than a predetermined number of times in the corpus;

Act 2: determining a total number of n-ary characters corresponding to all of the obtained words, wherein same n-ary characters are counted once, and each n-ary character represents n consecutive characters of an obtained word corresponding to the n-ary character;

Act 3: establishing, for each obtained word, based on each n-ary character corresponding to the obtained word, a feature vector whose dimensionality is the total number, wherein each dimension of the feature vector corresponds to a different n-ary character, and the value of each dimension indicates whether a corresponding n-ary character is mapped to the obtained word corresponding to the feature vector;

Act 4: traversing the corpus on which word segmentation is performed, and performing Act 5 with respect to the current word accessed during the traversal, and if the traversal is completed, performing Act 6; otherwise continuing the traversal;

Act 5: using a location of the current word in the corpus to determine context words by at least using the current word as a center, sliding towards both sides for at most k words to establish a window, and using the words in the window except the current word as the context words, inputting feature vectors of the context words associated with the current word into a convolutional layer of the convolutional neural network for convolutional calculation, and inputting a result of the convolutional calculation into a pooling layer of the convolutional neural network for pooling calculation to obtain a first vector; inputting a feature vector of the current word and a feature vector of a negative sample word selected in the corpus into the full connection layer of the convolutional neural network for calculation to obtain a second vector and a third vector; and updating parameters of the convolutional neural network based on the first vector, the second vector, the third vector, and a specified loss function, wherein

the convolutional calculation is performed according to the following formula:

$$\tilde{x}_i = x_{i:i+\theta-1} = \left[x_i^T, x_{i+1}^T, \ldots, x_{i+\theta-1}^T\right]^T$$

$$y_i = \sigma(\omega \tilde{x}_i + \zeta)$$

the pooling calculation is performed according to the following formula:

$c(j) = \underset{i=1,2,\ldots,t-\theta+1}{max}\{y_i(j)\}$, or$c(j) = \underset{i=1,2,\ldots,t-\theta+1}{\text{average}}\{y_i(j)\}$

and the loss function comprises:

$$l(w, c; \omega, \zeta, \varsigma, \tau) = log\left(1 + \sum_{m=1}^{\lambda} exp\left(-\gamma \cdot \left(s(w, c) - s(w'_m, c)\right)\right)\right)$$

wherein x_i indicates a feature vector of the i^{th} context word; $x_{i:i+\theta-1}$ indicates a vector that is obtained by concatenating the feature vectors of the i^{th} to the $i + \theta - 1^{\text{th}}$ context words; y_i indicates the i^{th} element in the vector that is obtained through the convolutional calculation; ω indicates weight parameters of the convolutional layer; ζ indicates offset parameters of the convolutional layer; σ indicates an activation function; max indicates a maximum value function; average indicates an average value function; $c(j)$ indicates the j^{th} element in the first vector that is obtained through pooling calculation; t indicates the number of context words; c indicates the first vector; w indicates the second vector; w'_m indicates the third vector corresponding to the m^{th} negative sample word; ω indicates weight parameters of the convolutional layer; ζ indicates offset parameters of the convolutional layer; ς indicates weight parameters of the full connection layer; τ indicates offset parameters of the full connection layer; γ indicates a hyperparameter; s indicates a similarity calculation function; and λ indicates the number of negative sample words; and

Act 6: inputting the feature vector of each obtained word into the full connection layer of the trained convolutional neural network for calculation to obtain corresponding word vectors.