Claims

1. A method performed by a computer for pre-processing a sequence of words for a neural machine translation (NMT), the method comprising:

obtaining an input string;

tokenizing the input string to form a sequence of words;

comparing each word in the sequence of words to a data structure to determine a corresponding named entity of each word;

tagging the corresponding named entities to each word;

splitting the sequence of words to form a plurality of word tokens;

combining the plurality of word tokens and named entities to form the processed string; and

processing the processed string using the NMT to convert the processed string into an alternative representation for the input string, wherein the step of combining the plurality of word tokens and named entities to form the processed string comprises:

aligning the word tokens with the corresponding named entities; and

generating word boundary tags (B,I,E) and adding the word boundary tags (B,I,E) between each of the word tokens and the corresponding named entities, where B is added to the first word token, E is added to the last word token, and I is added to word tokens between the first and last word tokens.

2. The method according to claim 1 wherein the step of splitting the sequence of words to form the plurality of word tokens further comprises:

determining an out of vocabulary (OOV) word in each of the sequence of words,

in response to determining the OOV word, splitting the OOV word into subword tokens using byte pair encoding;

in response to determining a non-OOV word, the non-OOV word is taken as the word token; and

adding subword connectors to each subword token other than the last subword token.

3. The method according to claim 2 wherein the step of combining the plurality of word tokens and named entities to form the processed string further comprises:

aligning the word tokens and subword tokens with the corresponding named entities; and

generating subword boundary tags (B_, I_, E_) and adding subword boundary tags (B_, I_, E_) between the subword token and the corresponding named entities,

where B_ is added to the first subword token, E_ is added to the last subword token, and I_ is added to subword tokens between the first and last subword tokens.

4. The method according to claim 1 further comprises:

splitting the sequence of words to form a plurality of character tokens; and

combining the plurality of character tokens and named entities to form the processed string.

5. The method according to claim 4 wherein the step of combining the plurality of character tokens and named entities to form the processed string comprises:

aligning the plurality of character tokens with the corresponding named entities; and

generating boundary tags (B,I,E) and adding the boundary tags (B,I,E) between each of the character tokens and the corresponding named entities, where B is added to the first character token, E is added to the last character token, and I is added to character tokens between the first and last character tokens.

6. A processing system for pre-processing a sequence of words for a neural machine translation (NMT), the processing system comprising:

a processor, a memory and instructions stored on the memory and executable by the processor to:

obtain an input string;

tokenize the input string to form a sequence of words;

compare each word in the sequence of words to a data structure to determine a corresponding named entity of each word; and

tag the corresponding named entities to each word;

split the sequence of words to form a plurality of word tokens; and

combine the plurality of word tokens and named entities to form the processed string; and

process the processed string using the NMT to convert the processed string into an alternative representation for the input string and wherein the instruction to combine the plurality of word tokens and named entities to form the processed string comprises instructions to:

align the word tokens with the corresponding named entities;

generate word boundary tags (B,I,E) and adding the word boundary tags (B,I,E) between each of the word token and the corresponding named

entities, where B is added to the first word token, E is added to the last word token, and I is added to word tokens between the first and last word tokens.

7. The processing system according to claim 6 wherein the instruction to split the sequence of words to form the plurality of word tokens further comprises instructions to:

determine an out of vocabulary (OOV) word in each of the sequence of words;

in response to determining the OOV word, split the OOV word into subword tokens using byte pair encoding;

in response to determining a non-OOV word, the non-OOV word is taken as the word token; and

add subword connectors to each subword token other than the last subword token.

8. The processing system according to claim 7 wherein the instruction to combine the plurality of word tokens and named entities to form the processed string further comprises instructions to:

align the word tokens and subword tokens with the corresponding named entities; and

generate subword boundary tags (B_, I_, E_) and adding subword boundary tags (B_, I_, E_) between the subword token and the corresponding named entities, where B_ is added to the first subword token, E_ is added to the last subword token, and I_ is added to subword tokens between the first and last subword tokens.

9. The processing system according to claim 11 wherein the instructions further comprise instructions to:

split the sequence of words to form a plurality of character tokens; and

combine the plurality of character tokens and named entities to form the processed string.

10. The processing system according to claim 9 wherein the instruction to combine the plurality of character tokens and named entities to form the processed string comprises instructions to:

align the plurality of character tokens with the corresponding named entities; and

generate boundary tags (B,I,E) and adding the boundary tags (B,I,E) between each of the character tokens and the corresponding named entities, where B is added

to the first character token, E is added to the last character token, and I is added to character tokens between the first and last character tokens.