# SYSTEM AND METHOD FOR RECOGNIZING SENTIMENT OF USER'S FEEDBACK

## TECHNICAL FIELD

**[0001]** Various embodiments relate to a system and a method for recognizing a sentiment of a user's feedback.

## BACKGROUND

**[0002]** To improve a quality of company's products or services, the companies may collect customer's feedbacks about their experiences with and expectations for the company's products or services. The companies may provide a space for the customers to leave their feedbacks, and the customers may leave their feedbacks on the space. For example, a customer may access an application program or a webpage operated by a company and leave a text feedback about the company's service or product.

**[0003]** However, while the customer leaves the text feedback via the application program or the webpage, the customer typically does not interact with a live person from the company. Therefore, the company may not know how the customer feels about the company's service or product, for example, a sentiment or an emotion of the customer's feedback. In particular, it may be difficult for the company to understand the sentiment or the emotion of the foreign customers' feedback written in the wide variety of languages.

## SUMMARY

**[0004]** According to various embodiments, a system for recognizing sentiment of user's feedback is provided. The system comprises: an encoder configured to receive an input text for the user's feedback, break up the input text into text tokens, and encode each text token; and a decoder connected to the encoder and configured to output a sentiment score for the input text. The encoder is further configured to read the encoded text token in a bidirectional manner to obtain a forward node value and a backward node value, and concatenate the forward node value and the backward node value together to obtain a vector representing semantic meaning of the input text. The decoder is configured to receive the vector from the encoder, obtain a probability of each sentiment score by mapping the vector into each sentiment score, and determine the sentiment score for the input text based on the probability.

**[0005]** In some embodiments, the input text relates to a sentence, and the encoder is configured to break up the input text into the text tokens each corresponding to a plurality of words included in the sentence.

**[0006]** In some embodiments, the encoder is further configured to input the encoded text token into a Bidirectional Long Short-Term Memory Network (Bi-LSTM) model configured to read the encoded text token in the bidirectional manner.

**[0007]** In some embodiments, the encoder comprises a series of Bi-LSTM layers in which the Bi-LSTM model is operated. In a first Bi-LSTM layer of the Bi-LSTM layers, the encoder is configured to input the encoded text token into the Bi-LSTM model, and the Bi-LSTM model is configured to obtain a first forward node value and a first backward node value and concatenate the first forward node value and the first backward node value together. In a next Bi-LSTM layer of the Bi-LSTM layers, the encoder is configured to input the concatenated first forward node value and first backward node value into the Bi-LSTM model, and the Bi-LSTM model is configured to obtain a next forward node value and a next backward node value and concatenate the next forward node value and the next backward node value together.

**[0008]** In some embodiments, the encoder comprises a max pooling layer in which a max pooling function is operated. In a final Bi-LSTM layer of the Bi-LSTM layers, the Bi-LSTM model is configured to obtain a final forward node value and a final backward node value, and concatenate the final forward node value and the final backward node value together. In the max pooling layer, the max pooling function is configured to process the concatenated final forward node value and final backward node value to obtain the vector.

**[0009]** In some embodiments, the encoder is further configured to encode the each text token using a Byte Pair Encoding model.

**[0010]** In some embodiments, the encoder is further configured to encode a stop token representing an end of the input text using the Byte Pair Encoding model, and input the encoded stop token into the Bi-LSTM model.

**[0011]** In some embodiments, the decoder is configured to input the vector into a Feedforward Neural Network (FFNN) model configured to map the vector to the each sentiment score.

**[0012]** In some embodiments, the decoder comprises a series of FFNN layers in which the FFNN model is operated, and a Softmax layer in which a Softmax function is operated. The decoder is configured to input the vector into the FFNN model in the series of FFNN layers to obtain an output, and input the output of the FFNN model into the Softmax function. The Softmax function is configured to obtain the probability of the each sentiment score and determine the sentiment score for the input text based on the probability.

**[0013]** In accordance with various embodiments, there is a method of recognizing a sentiment of a user's feedback. The method includes: receiving an input text for the user's feedback; breaking up the input text into text tokens; encoding each text token; reading the encoded text token in a bidirectional manner to obtain a forward node value and a backward node value; concatenating the forward node value and the backward node value together to obtain a vector representing semantic meaning of the input text; obtaining a probability of each sentiment score

by mapping the vector into each sentiment score; and determining a sentiment score for the input text based on the probability.

**[0014]** In some embodiments, the input text relates to a sentence, and breaking up the input text into text tokens includes: breaking up the input text into the text tokens each corresponding to a plurality of words included in the sentence.

**[0015]** In some embodiments, reading the encoded text token in a bidirectional manner to obtain a forward node value and a backward node value includes: inputting the encoded text token into a Bidirectional Long Short-Term Memory Network (Bi-LSTM) model configured to read the encoded text token in the bidirectional manner.

**[0016]** In some embodiments, the Bi-LSTM model is operated in a series of Bi-LSTM layers. Reading the encoded text token in a bidirectional manner to obtain a forward node value and a backward node value includes: inputting the encoded text token into the Bi-LSTM model in a first Bi-LSTM layer of the Bi-LSTM layers; obtaining a first forward node value and a first backward node value; concatenating the first forward node value and the first backward node value together; inputting the concatenated first forward node value and first backward node value into the Bi-LSTM model in a next Bi-LSTM layer of the Bi-LSTM layers; obtaining a next forward node value and a next backward node value; and concatenating the next forward node value and the next backward node value together.

**[0017]** In some embodiments, a max pooling function is operated in a max pooling layer. Reading the encoded text token in a bidirectional manner to obtain a forward node value and a backward node value further includes: obtaining a final forward node value and a final backward node value in a final Bi-LSTM layer of the Bi-LSTM layers; concatenating the final forward node value and the final backward node value together in the final Bi-LSTM layer of the Bi-LSTM layers; processing the concatenated final forward node value and final backward node value in the max pooling layer; and obtaining the vector in the max pooling layer.

**[0018]** In some embodiments, encoding each text token includes: encoding the each text token using a Byte Pair Encoding model.

**[0019]** In some embodiments, the method further includes: encoding a stop token representing an end of the input text using the Byte Pair Encoding model; and inputting the encoded stop token into the Bi-LSTM model.

**[0020]** In some embodiments, obtaining a probability of each sentiment score by mapping the vector into each sentiment score includes: inputting the vector into a Feedforward Neural Network (FFNN) model configured to map the vector to the each sentiment score.

**[0021]** In some embodiments, the FFNN model is operated in a series of FFNN layers, and a Softmax function is operated in a Softmax layer. Obtaining a probability of each sentiment score by mapping the vector into each sentiment score includes: inputting the vector into the FFNN model to obtain an output in the series of FFNN layers; inputting the output of the FFNN model into the Softmax function in the Softmax layer; obtaining the probability of the each sentiment score in the Softmax layer; and determining the sentiment score for the input text based on the probability in the Softmax layer.

**[0022]** According to various embodiments, a data processing apparatus configured to perform the method of any one of the above embodiments is provided.

**[0023]** According to various embodiments, a computer program element comprising program instructions, which, when executed by one or more processors, cause the one or more processors to perform the method of any one of the above embodiments is provided.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[0024]** The invention will be better understood with reference to the detailed description when considered in conjunction with the non-limiting examples and the accompanying drawings, in which:

- FIG. 1 shows a block diagram for a system for recognizing a sentiment of a user's feedback according to various embodiments.
- FIG. 2 shows an exemplary flowchart for a method of recognizing a sentiment of a user's feedback according to various embodiments.
- FIG. 3 shows an exemplary diagram for a system for recognizing a sentiment of a user's feedback according to various embodiments.

## DETAILED DESCRIPTION

**[0025]** The following detailed description refers to the accompanying drawings that show, by way of illustration, specific details and embodiments in which the disclosure may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the disclosure. Other embodiments may be utilized and structural, and logical changes may be made without departing from the scope of the disclosure. The various embodiments are not necessarily mutually exclusive, as some embodiments can be combined with one or more other embodiments to form new embodiments.

**[0026]** Embodiments described in the context of one of a system and a method are analogously valid for the other system and method. Similarly, embodiments described in the context of a system are analogously valid for a method, and vice-versa.

**[0027]** Features that are described in the context of an embodiment may correspondingly be applicable to the same or similar features in the other embodiments. Features that are described in the context of an embodiment may correspondingly be applicable to the other embodiments,

even if not explicitly described in these other embodiments. Furthermore, additions and/or combinations and/or alternatives as described for a feature in the context of an embodiment may correspondingly be applicable to the same or similar feature in the other embodiments.

**[0028]** In the context of various embodiments, the articles "a", "an" and "the" as used with regard to a feature or element include a reference to one or more of the features or elements.

**[0029]** As used herein, the term "and/or" includes any and all combinations of one or more of the associated listed items.

**[0030]** In the following, embodiments will be described in detail.

**[0031]** FIG. 1 shows a block diagram for a system 1000 for recognizing a sentiment of a user's feedback according to various embodiments.

**[0032]** The system 1000 may be a set of interacting elements. The elements may be, by way of example and not of limitation, one or more mechanical components, one or more electrical components, and/or one or more instructions, for example, encoded in a storage media.

**[0033]** As shown in FIG. 1, the system 1000 may include an encoder 100 and a decoder 200. The encoder 100 may convert information, for example, an input text, into another format. The decoder 200 may process the another format of the information and output a sentiment score for the input text. In some embodiments, the encoder 100 and the decoder 200 may be connected to each other. For example, the encoder 100 and the decoder 200 may be connected to each other via at least one middle element. As another example, the encoder 100 and the decoder 200 may be directly coupled to each other.

**[0034]** In some embodiments, the encoder 100 and the decoder 200 may be mounted on a single device. For example, the encoder 100 and the decoder 200 may be mounted on different processors of a single device. As another example, the encoder 100 and the decoder 200 may be mounted on a single processor of the single device, for example, as program instructions. In

some other embodiments, the encoder 100 and the decoder 200 may be mounted on different devices. The encoder 100 and the decoder 200 may be capable of data communication.

**[0035]** In some embodiments, at least one of the encoder 100 and the decoder 200 may include a microprocessor, an analogue circuit, a digital circuit, a mixed-signal circuit, a logic circuit, an integrated circuit, a Central Processing Unit (CPU), a Graphics Processing Unit (GPU), a Digital Signal Processor (DSP), a Field Programmable Gate Array (FPGA), an Application Specific Integrated Circuit (ASIC), etc., or any combination thereof. Any other kind of implementation of the respective functions, which will be described below in further detail, may also be understood as the at least one of the encoder 100 and the decoder 200.

**[0036]** In some other embodiments, the processor in which at least one of the encoder 100 and the decoder 200 is mounted may include a microprocessor, an analogue circuit, a digital circuit, a mixed-signal circuit, a logic circuit, an integrated circuit, a Central Processing Unit (CPU), a Graphics Processing Unit (GPU), a Digital Signal Processor (DSP), a Field Programmable Gate Array (FPGA), an Application Specific Integrated Circuit (ASIC), etc., or any combination thereof. Any other kind of implementation of the respective functions, which will be described below in further detail, may also be understood as the processor.

**[0037]** In some embodiments, the system 1000 may further include a memory (not shown). The memory may be used by the encoder 100 and the decoder 200 to permanently or temporarily store data. The memory may include, but not be limited to, a cloud memory, a server memory, and a physical storage, for example a RAM (random-access memory), an HDD (hard disk drive), an SSD (solid-state drive), others, or any combinations thereof.

**[0038]** In accordance with various embodiments, the encoder 100 may receive the input text for a feedback from a customer (hereinafter, referred to as a “user”). In some embodiments, the system 1000 may further include an input device (not shown). In some embodiments, the input device and the encoder 100 may be mounted on the same device. In some other embodiments,

the input device and the encoder 100 may be mounted on different devices. For example, the input device may be mounted in a client device (also referred to as a "user's device"). The input device and the encoder 100 may be capable of data communication. The client device may include, but not be limited to, a mobile phone, a tablet computer, a laptop computer, a desktop computer, a smart watch, smart glasses, a television, a navigation device, a PDA (Personal Digital Assistant) and a PMP (Portable Multimedia Player).

**[0039]** In some embodiments, the input device may receive the input text relating to the feedback for a company's products and/or services from the user. The input text may include string of words constructing a sentence that the user may input. For example, the input text may include the string of words constructing the sentence that the user may type into a feedback box of an application program or a webpage operated by the company.

**[0040]** In some embodiments, the encoder 100 may break up the input text into text tokens. For example, the encoder 100 may break up the input text into the text tokens each corresponding to a plurality of words included in the sentence that the user may type. In some embodiments, the encoder 100 may break up the input text into text tokens by tokenizing the input text. The encoder 100 may tokenize the input text using a tokenizer, for example, Moses Tokenizer, to separate punctuation from the words while preserving special tokens such as dates.

**[0041]** In some embodiments, the encoder 100 may encode each text token. For example, the encoder 100 may encode the each text token using a Byte Pair Encoding model in which the most common pair of consecutive bytes of the text token is replaced with a byte which does not occur within the text token. In this manner, the each text token may be compressed.

**[0042]** In some embodiments, the encoder 100 may read the encoded text token in a bidirectional manner to obtain a forward node value and a backward node value. The encoder 100 may then concatenate the forward node value and the backward node value together, to

obtain a vector representing semantic meaning of the input text. As an example, the encoder 100 may input the encoded text token into an autoencoder, for example, Bidirectional Long Short-Term Memory Network (Bi-LSTM) model which is configured to read the encoded text token in the bidirectional manner. The Bi-LSTM model may be a sequence processing model including two LSTMs. The Bi-LSTM model may read the encoded text token in a forward direction and in a backwards direction, and obtain the forward node value and the backward node value respectively.

**[0043]** In some embodiments, the encoder 100 may comprise a series of Bi-LSTM layers, for example, a first Bi-LSTM layer, a next Bi-LSTM layer, etc. The encoder 100 may input the encoded text token into the Bi-LSTM model in the first Bi-LSTM layer. The Bi-LSTM model may obtain a first forward node value and a first backward node value by reading the encoded text token in the forward direction and in the backward direction respectively in the first Bi-LSTM layer. The Bi-LSTM model may concatenate the first forward node value and the first backward node value together in the first Bi-LSTM layer.

**[0044]** In some embodiments, the encoder 100 may then input the concatenated first forward node value and first backward node value into the Bi-LSTM model in the next Bi-LSTM layer positioned after the first Bi-LSTM layer. The Bi-LSTM model may obtain a next forward node value and a next backward node value by reading the concatenated first forward node value and first backward node value in the forward direction and in the backward direction respectively in the next Bi-LSTM layer. The Bi-LSTM model may concatenate the next forward node value and the next backward node value together in the next Bi-LSTM layer. In some embodiments, these operations described herein may be repeated until a final Bi-LSTM layer of the Bi-LSTM layers is reached.

**[0045]** In some embodiments, the encoder 100 may include a max pooling layer in which a max pooling function is operated. The max pooling layer may be positioned after the final Bi-

LSTM layer of the Bi-LSTM layers. In some embodiments, the Bi-LSTM model in the final Bi-LSTM layer may obtain a final forward node value and a final backward node value. The Bi-LSTM model in the final Bi-LSTM layer may concatenate the final forward node value and the final backward node value together. The max pooling function may then process the concatenated final forward node value and final backward node value in the max pooling layer, and obtain a vector including the set of values corresponding to each text token. In this regard, the encoder 100 may output the vector.

**[0046]** In some embodiments, the encoder 100 may encode a stop token representing the end of the input text using the Byte Pair Encoding model. The encoder 100 may input the encoded stop token into the Bi-LSTM model. For example, the set of values constructing the vector may also include a value corresponding to the stop token.

**[0047]** In some embodiments, the decoder 200 may output the sentiment score for the input text. The decoder 200 may receive the vector from the encoder 100 as an input. The decoder 200 may obtain a probability of each sentiment score by mapping the vector into each sentiment score, and determine the sentiment score for the input text based on the probability.

**[0048]** In some embodiments, the decoder 200 may input the vector into a Feedforward Neural Network (FFNN) model which is configured to map the vector to the each sentiment score. The FFNN model may be an artificial neural network in which connections between layers do not form a cycle and data moves in one direction, for example a forward direction, from an input layer to an output layer through one or more hidden layers.

**[0049]** In some embodiments, the decoder 200 may include a series of FFNN layers, for example, a first layer as an input layer of the decoder 200, and one or more hidden layers including a final layer.

**[0050]** In some embodiments, the decoder 200 may include a Softmax layer in which a Softmax function is operated. The Softmax layer may be positioned after the final layer of the FFNN

layers. The Softmax function is configured to generalize the output of the final FFNN layer. The decoder 200 may input the vector into the FFNN model in the series of FFNN layers to obtain an output. The decoder 200 may then input the output of the series of FFNN layers into the Softmax function. The Softmax function in the Softmax layer may obtain the probability of the each sentiment score, and determine the sentiment score for the input text based on the probability.

**[0051]** It may be appreciated that the sentiment score obtained by the decoder 200 may be a polarity score, as opposed to a binary positive/negative score, for the sentiment of the user's feedback. As such, the company may correctly understand how the user feels about the company's product or service, from the user's feedback.

**[0052]** In some embodiments, the FFNN model may be trained to correctly map the vector to the each sentiment score. The FFNN model may be trained on a synthetic data set. The training data may include a sentence embedding, in the form of the vector, generated by the Bi-LSTM model of the encoder 100 and a corresponding sentiment score (for example, a sentiment score of the input text used to generate the sentence embedding).

**[0053]** To generate a polarity score, as opposed to a binary positive/negative score, for the sentiment of the user's feedback, a large volume of labelled training data may be needed to train the FFNN model. The system 1000 in accordance with various embodiments may create synthetic sentiment labels, to avoid having to manually label the training data.

**[0054]** In some embodiments, to create the synthetic sentiment labels, the system 1000 may use a rating, for example, a star rating, received from a user as a weak proxy for a sentiment score. The system 1000 may additionally use a sentiment score for a user's text feedback obtained from another sentiment analysis model, for example, off the shelf sentiment analysis model, as a weak proxy for the sentiment of the user's text feedback. One of examples of the off the shelf sentiment analysis model may be VADER (Valence Aware Dictionary and

Sentiment Reasoner). In instances where there is an agreement in the rating and the sentiment score, the system 1000 may take the two (2) weak proxies and combine them to create a strong proxy. The strong proxy may be used to train the FFNN model of the system 1000. By using the synthetic sentiment label to generate training and validation data, huge amount of data may be programmatically labelled, and it may not be required to manually label the huge amount of data which may be costly and time consuming.

**[0055]** In some embodiments, the FFNN model may be trained on data of a certain language, for example, English data. After the FFNN model is trained on the English data, the system 1000 may input data in other languages, for example, non-English data. After obtaining the weak predictions, for example, a predicted sentiment score and the rating of the user, regarding the non-English data, the system 1000 may look for instances where the predicted sentiment score is matched with the rating of the user and use the predicted sentiment score as a synthetic sentiment label. The system 1000 may combine the two (2) weak proxies, for example, the predicted sentiment score and the rating of the user, together to form a strong proxy. Then, the FFNN model may further be trained on the new non-English data.

**[0056]** As described above, to initially create the synthetic sentiment label, the off the shelf model, for example, the VADER, and the rating of the user may be used. When there is an agreement between the rating and the sentiment score of the VADER, the system 1000 may use the sentiment score as the synthetic sentiment label. For example, this may be done for English data. As an example, for non-English data, the system 1000 may use a newly trained model, for example, taking the output of the decoder 200, and combine the output with the rating of the user to obtain the synthetic sentiment label for the non-English data.

**[0057]** In some embodiments, the text feedback may be inputted, a sentence embedding for the inputted text feedback may be generated using the encoder 100, and the sentence embedding may be inputted into the decoder 200. Parameters of the decoder 200 including the FFNN

model may be updated each time when the decoder 200 makes an error on the training data. After certain rounds of training, the decoder 200 may be tested on a validation data set to check an accuracy of the decoder 200. As described above, the synthetic sentiment label may be used for training and validation of the system 1000.

**[0058]** FIG. 2 shows an exemplary flowchart for a method 2000 of recognizing a sentiment of a user's feedback according to various embodiments. According to various embodiments, the method 2000 of recognizing the sentiment of the user's feedback may be provided.

**[0059]** In some embodiments, the method 2000 may include a step 301 of receiving an input text for the user's feedback. For example, the input text may include string of words constructing a sentence that the user may input.

**[0060]** In some embodiments, the method 2000 may include a step 302 of breaking up the input text into text tokens. For example, the input text may be tokenized into the text tokens using a tokenizer, for example, Moses Tokenizer.

**[0061]** In some embodiments, the method 2000 may include a step 303 of encoding each text token. For example, each text token may be encoded using a Byte Pair Encoding model.

**[0062]** In some embodiments, the method 2000 may include a step 304 of reading the encoded text token in a bidirectional manner to obtain a forward node value and a backward node value. As an example, each encoded text token may be inputted into a Bidirectional Long Short-Term Memory Network (Bi-LSTM) model which is configured to read each encoded text token in a bidirectional manner. The forward node value and the backward node value for each text token may be obtained by the Bi-LSTM model in a series of Bi-LSTM layers.

**[0063]** In some embodiments, the method 2000 may include a step 305 of concatenating the forward node value and the backward node value together to obtain a vector representing semantic meaning of the input text. For example, the forward node value and the backward node value may be concatenated by the Bi-LSTM model in a final Bi-LSTM layer and the

concatenated forward node value and the backward node value may be provided to a max pooling layer. The concatenated forward node value and the backward node value may be processed to obtain the vector including a set of values corresponding to each text token.

**[0064]** In some embodiments, the method 2000 may include a step 306 of obtaining a probability of each sentiment score by mapping the vector into each sentiment score. For example, the vector may be inputted into a Feedforward Neural Network (FFNN) model configured to map the vector to the each sentiment score. The probability of each sentiment score may be obtained by the FFNN model.

**[0065]** In some embodiments, the method 2000 may include a step 307 of determining a sentiment score for the input text based on the probability. For example, one of sentiment scores which has the highest probability may be determined as the sentiment score for the input text.

**[0066]** In this manner, in accordance with various embodiments, the method 2000 may provide a polarity score for the sentiment of the user's feedback, as opposed to a simple binary positive/negative score. As such, a company may correctly understand how the user feels about the company's product or service, from the user's feedback. In addition, in accordance with various embodiments, the method 2000 may perform a sentence embedding in the form of the vector in a language-agnostic way, for example, by using the Bi-LSTM model. Therefore, the company may correctly understand how the user feels about the company's product or service, from the user's feedback written in wide variety of languages.

**[0067]** FIG. 3 shows an exemplary diagram for a system 1000 for recognizing a sentiment of a user's feedback according to various embodiments.

**[0068]** As shown in FIG. 3, the system 1000 may include an encoder 100 and a decoder 200 which are connected to each other. The encoder 100 may take an input text relating to a sentence, and output a sentence embedding in the form of a vector. The decoder 200 may take

the output of the encoder 100 (i.e. the sentence embedding in the form of the vector) as its input, and output a sentiment score as a sentiment prediction for the input text.

**[0069]** To perform semantic level analysis, the system 1000 may use a technique capable of encoding the input text into the sentence embedding in a language-agnostic way. For the language-agnostic way, a Bidirectional Long Short-Term Memory Network (Bi-LSTM) model trained on a multi-language training set may be used in the encoder 100. The Bi-LSTM model may be scaled across multiple languages, and is capable of operating at a semantic level as opposed to a syntactic level.

**[0070]** In some embodiments, an input to the Bi-LSTM model may be a piece of the input text. As shown in FIG. 3, each word in the sentence of the input text may be represented by $x_n$, where n is a position of the word in the sentence. The word may be converted to a Byte Pair Encoding. The encodings may then be inputted into the Bi-LSTM model. The output of a final layer of the encoder 100 may be pooled together. For example, a size of the pooled layer may be 1024 dimensions. The 1024-dimension layer may contain the embedding for the input text as the output of the encoder 100.

**[0071]** In some embodiments, the sentiment analysis of the system 1000 may be performed at a word level. The encoder 100 may break up the input text into text tokens 110 each corresponding to a plurality of words included in the sentence using a tokenizer, for example, Moses Tokenizer. As an example, the input text for the sentence including two words may be tokenized into two text tokens (hereinafter, referred to as “a first text token 110a” and “a second text token 110b”). Although not shown, if the input text for the sentence includes a plurality of words, the input text may be tokenized into the plurality of text tokens.

**[0072]** In some embodiments, the encoder 100 may include an input layer 120. A Byte Pair Encoding model may operate in the input layer 120. In the input layer 120, the encoder 100 may encode each text token using the Byte Pair Encoding model. For example, the encoder

100 may encode the first text token 110a using the Byte Pair Encoding model 120a, and encode the second text token 110b using the Byte Pair Encoding model 120b. By using the Byte Pair Encoding model, a common pair of consecutive bytes of the first text token 110a may be replaced with a byte which does not occur within the first text token 110a. Likewise, a common pair of consecutive bytes of the second text token 110b may be replaced with a byte which does not occur within the second text token 110b.

**[0073]** In some embodiments, the encoder 100 may include a series of hidden layers 130. The Bi-LSTM model may operate in the hidden layers 130. For example, the encoder 100 may include a series of Bi-LSTM layers 131-135. As shown in FIG. 3, the encoder 100 may include five (5) Bi-LSTM layers 131-135, but it may be appreciated that the number of the Bi-LSTM layers is not limited thereto. In accordance with various embodiments, the plurality of hidden layers 130 may capture a latent structure of the sentence, including overall meaning of the sentence.

**[0074]** In some embodiments, in the first Bi-LSTM layer 131, a Bi-LSTM model 131a for the first text token 110a may read the encoded first text token 110a in a bidirectional manner to obtain a first forward node value and a first backward node value for the first text token 110a. The Bi-LSTM model 131a for the first text token 110a may then concatenate the first forward node value and the first backward node value for the first text token 110a together. In the first Bi-LSTM layer 131, a Bi-LSTM model 131b for the second text token 110b may read the encoded second text token 110b in a bidirectional manner to obtain a first forward node value and a first backward node value for the second text token 110b. The Bi-LSTM model 131b for the second text token 110b may then concatenate the first forward node value and the first backward node value for the second text token 110b together.

**[0075]** In some embodiments, in a second Bi-LSTM layer 132 which is a layer positioned after the first Bi-LSTM layer 131, a Bi-LSTM model 132a for the first text token 110a may read the

concatenated first forward node value and first backward node value for the first text token 110a to obtain a second forward node value and a second backward node value for the first text token 110a. In the second Bi-LSTM layer 132, the Bi-LSTM model 132a for the first text token 110a may then concatenate the second forward node value and the second backward node value for the first text token 110a together. In the second Bi-LSTM layer 132, a Bi-LSTM model 132b for the second text token 110b may read the concatenated first forward node value and first backward node value for the second text token 110b in a bidirectional manner to obtain a second forward node value and a second backward node value for the second text token 110b. In the second Bi-LSTM layer 132, the Bi-LSTM model 132b for the second text token 110b may then concatenate the second forward node value and the second backward node value for the second text token 110b together.

**[0076]** In some embodiments, the above-described operations may be repeated until the final Bi-LSTM layer, for example, the fifth Bi-LSTM layer 135, is reached.

**[0077]** In some embodiments, the encoder 100 may include an output layer 140. The output layer 140 may be or include a max pooling layer which may perform a max pooling function.

**[0078]** The max pooling layer may be positioned after the final Bi-LSTM layer, for example, the fifth Bi-LSTM layer 135, of the hidden layers 130. In some embodiments, in the fifth Bi-LSTM layer 135, a Bi-LSTM model 135a for the first text token 110a may read the concatenated fourth forward node value and fourth backward node value for the first text token 110a to obtain a fifth forward node value and a fifth backward node value for the first text token 110a as a final forward node value and a final backward node value for the first text token 110a. In the fifth Bi-LSTM layer 135, a Bi-LSTM model 135b for the second text token 110b may read the concatenated fourth forward node value and fourth backward node value for the second text token 110b to obtain a fifth forward node value and a fifth backward node

value for the second text token 110b as a final forward node value and a final backward node value for the second text token 110b.

**[0079]** In the fifth Bi-LSTM layer 135, the Bi-LSTM model 135a for the first text token 110a may concatenate the final forward node value and the final backward node value for the first text token 110a together. In the fifth Bi-LSTM layer 135, the Bi-LSTM model 135b for the second text token 110b may concatenate the final forward node value and the final backward node value for the second text token 110b together.

**[0080]** In the max pooling layer, the max pooling function may receive the concatenated final forward node value and final backward node value for the first text token 110a and the second text token 110b. The max pooling function may then split the concatenated final forward node value and the final backward node value into pieces, and take the maximum value of each piece to obtain a final value for the first text token 110a and a final value for the second text token 110b. The max pooling function may then obtain a vector including the set of final values corresponding to each text token, for example, the final value for the first text token 110a and the final value for the second text token 110b. In this regard, the encoder 100 may output the vector including the final value for the first text token 110a and the final value for the second text token 110b.

**[0081]** For example, if the Bi-LSTM model has 2048 nodes and the sentence embedding has 1024 dimensions, the max pooling function may divide the 2048 nodes by 1024 equal pieces each containing two (2) nodes. The max pooling function may take the maximum value output by these two (2) nodes and use the maximum value as the final value for the corresponding dimensions. For example, if the values of the first four (4) nodes are “1, 4, 3, 7”, the value of the first dimension may be “4” (i.e. maximum value of 1 and 4), and the value of the second dimension may be “7” (i.e. maximum value of 3 and 7). In some embodiments, the vector may

be a 1024-dimension vector. For example, the vector representing the semantic meaning of the input text may include 1024 numbers.

**[0082]** In some embodiments, as shown in FIG. 3, a stop token 110m may be used to inform the Bi-LSTM model of the end of the input text. Since the Bi-LSTM model may use a tokenized sentence, the stop token 110m may be used as a final token. In some embodiments, the encoder 100 may encode the stop token 110m representing the end of the input text using a Byte Pair Encoding model 120m in the input layer 120. The encoder 100 may input the encoded stop token into a Bi-LSTM model 131m. In this regard, the vector may also include a final value for the stop token 110m.

**[0083]** In some embodiments, decoder 200 may include a series of a Feedforward Neural Network (FFNN) layers in which a FFNN model is operated. The FFNN model may map the vector to the each sentiment score. The decoder 200 may include an input layer 221 (also referred to as "a first layer 221") and a plurality of hidden layers, for example, three (3) hidden layers 222, 223, 224 (also referred to as "a second layer 222", "a third layer 223", and "a fourth layer 224" respectively). The plurality of hidden layers 222, 223, 224 may be connected to each other. As shown in FIG. 3, the decoder 100 may include four (4) layers 221-224 for the FFNN model, for example, the first to fourth layers 221-224, but it may be appreciated that the number of layers for the FFNN model is not limited thereto. In accordance with various embodiments, the plurality of layers 221-224 may capture a latent structure of the sentence, including overall meaning of the sentence.

**[0084]** In the first layer 221, the decoder 200 may input the vector outputted from the encoder 100 into the FFNN model 221a-221n. For example, the vector may be the 1024-dimension vector, and the each value 210a-210n of the 1024-dimension vector may be inputted into the FFNN model 221a-221n respectively. As an example, each layer in the decoder 200 may have 1024 nodes (also referred to as "cells").

**[0085]** The decoder 200 may input the vector into the FFNN model in the series of hidden layers 222-224 to obtain an output. For example, the decoder 200 may use a Rectified Linear Unit (ReLU) activation function in the hidden layers 222-224. It may be appreciated that the decoder 200 may use another activation function for the output of the hidden layers 222-224. For example, the activation function may be a linear function allowing the FFNN model to learn faster.

**[0086]** In some embodiments, the decoder 200 may include an output layer 230. The output layer 230 may be or include a Softmax layer which may perform a Softmax function. The Softmax layer may be positioned after the final layer, for example, the fourth layer 224, of the FFNN layers. The Softmax function may generalize the output of the fourth layer 224. The decoder 200 may then input the output of the series of FFNN layers 221-224 into the Softmax function. The Softmax function in the Softmax layer may obtain the probability of the each sentiment score, and determine the sentiment score for the input text based on the probability.

**[0087]** In some embodiments, the FFNN model may be trained to correctly map the sentence to the correct sentiment score. The FFNN model may be trained on a synthetic data set. The training data may include the sentence embedding generated by the Bi-LSTM model of the encoder 100 and a corresponding sentiment score (for example, a sentiment score of the input text used to generate the sentence embedding).

**[0088]** For example, the sentiment score may have five (5) possible values from “1” (for example, representing very negative) to “5” (for example, representing very positive). Each position in the Softmax function output may represent a sentiment score, with the first position representing a score of one (1), the second position representing a score of two (2), etc. Each value in the Softmax function may represent the probability of the input text being that particular sentiment score. For example, after collecting all the output values, if the collected

output values from the Softmax function are [0.01, 0.01, 0.01, 0.01, 0.96], the decoder 200 may determine that the input text may have a sentiment score of "5" with a probability of 96%.

**[0089]** As described above, in accordance of various embodiments, the system 1000 may generate a polarity score, as opposed to a binary positive/negative score, for the sentiment of the user's feedback. Having the ability to generate the sentiment score for the user's feedback may provide a number of strategic advantages. The sentiment score outputted from the system 1000 may be used for various use cases as follows, but not be limited thereto.

- High-level strategic decision making:
  - The sentiment scores of the system 1000 in accordance with various embodiments may be used in a research for a company's strategy. The research may influence the company's product strategy as sentiment score may be used to determine which projects should be prioritized and worked on by assessing volume and severity of user pain points.
  - The share of negative feedback (for example, the number of total feedback divided by the number of negative feedback) may be a metric that may be used as objectives and key results.
- Product-related:
  - The sentiment scores may be used as an experimental guardrail metric. When experiments are conducted, the sentiment scores may be used to ensure that new product features may not cause a degraded user experience.
  - The sentiment scores may be used in retrospective analysis to diagnose faults within existing product rollouts.
- Miscellaneous:
  - The sentiment scores may be used as inputs to model and predict how likely a user is to churn, and the future lifetime value of the user.

**[0090]** While the disclosure has been particularly shown and described with reference to specific embodiments, it should be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the invention as defined by the appended claims. The scope of the invention is thus indicated by the appended claims and all changes which come within the meaning and range of equivalency of the claims are therefore intended to be embraced.

**CLAIM**

1. A system for recognizing a sentiment of a user's feedback comprising:

an encoder configured to receive an input text for the user's feedback, break up the input text into text tokens, and encode each text token; and

a decoder connected to the encoder and configured to output a sentiment score for the input text,

wherein the encoder is further configured to read the encoded text token in a bidirectional manner by a Bidirectional Long Short-Term Memory Network (Bi-LSTM) model to obtain a forward node value and a backward node value, and concatenate the forward node value and the backward node value together to obtain a vector representing semantic meaning of the input text,

the decoder is configured to receive the vector from the encoder, obtain a probability of each sentiment score by mapping the vector into each sentiment score, and determine the sentiment score for the input text based on the probability,

the encoder comprises a series of Bi-LSTM layers in which the Bi-LSTM model is operated,

in a first Bi-LSTM layer of the Bi-LSTM layers, the encoder is configured to input the encoded text token into the Bi-LSTM model, and the Bi-LSTM model is configured to obtain a first forward node value and a first backward node value and concatenate the first forward node value and the first backward node value together, and

in a next Bi-LSTM layer of the Bi-LSTM layers, the encoder is configured to input the concatenated first forward node value and first backward node value into the Bi-LSTM model, and the Bi-LSTM model is configured to obtain a next forward node value and a next backward

node value and concatenate the next forward node value and the next backward node value together, and

wherein the decoder is configured to use a Feedforward Neural Network (FFNN) model, trained on data of a certain language, to determine the sentiment score for the input text based on the probability, and

where the input text is in other language, the system is configured to obtain the sentiment score output from the decoder and a rating of the user regarding the input text in the other language, combine the sentiment score and the rating of the user, and further train the FFNN model on data of the other language using the combined sentiment score and the rating of the user.

2. The system according to claim 1, wherein the input text relates to a sentence, and the encoder is configured to break up the input text into the text tokens each corresponding to a plurality of words included in the sentence.

3. The system according to claim 1 or claim 2, wherein the encoder comprises a max pooling layer in which a max pooling function is operated,

in a final Bi-LSTM layer of the Bi-LSTM layers, the Bi-LSTM model is configured to obtain a final forward node value and a final backward node value, and concatenate the final forward node value and the final backward node value together, and

in the max pooling layer, the max pooling function is configured to process the concatenated final forward node value and final backward node value to obtain the vector.

4. The system according to any one of claims 1 to 3, wherein the encoder is further configured to encode the each text token using a Byte Pair Encoding model.

5. The system according to claim 4, wherein the encoder is further configured to encode a stop token representing an end of the input text using the Byte Pair Encoding model, and input the encoded stop token into the Bi-LSTM model.

6. The system according to any one of claims 1 to 5, wherein the decoder is configured to input the vector into the FFNN model configured to map the vector to the each sentiment score.

7. The system according to claim 6, wherein the decoder comprises a series of FFNN layers in which the FFNN model is operated, and a Softmax layer in which a Softmax function is operated,

the decoder is configured to input the vector into the FFNN model in the series of FFNN layers to obtain an output, and input the output of the FFNN model into the Softmax function, and

the Softmax function is configured to obtain the probability of the each sentiment score and determine the sentiment score for the input text based on the probability.

8. A method of recognizing a sentiment of a user's feedback, the method comprising:

receiving an input text for the user's feedback;

breaking up the input text into text tokens;

encoding each text token;

reading the encoded text token in a bidirectional manner by a Bidirectional Long Short-Term Memory Network (Bi-LSTM) model to obtain a forward node value and a backward node value;

concatenating the forward node value and the backward node value together to obtain a vector representing semantic meaning of the input text;

obtaining a probability of each sentiment score by mapping the vector into each sentiment score; and

determining a sentiment score for the input text based on the probability,

wherein the Bi-LSTM model is operated in a series of Bi-LSTM layers, and

the reading the encoded text token in a bidirectional manner comprises:

inputting the encoded text token into the Bi-LSTM model in a first Bi-LSTM layer of the Bi-LSTM layers;

obtaining a first forward node value and a first backward node value;

concatenating the first forward node value and the first backward node value together;

inputting the concatenated first forward node value and first backward node value into the Bi-LSTM model in a next Bi-LSTM layer of the Bi-LSTM layers;

obtaining a next forward node value and a next backward node value; and

concatenating the next forward node value and the next backward node value together, and

wherein the determining the sentiment score for the input text based on the probability comprises: using a Feedforward Neural Network (FFNN) model, trained on data of a certain language, to determine the sentiment score for the input text based on the probability, and

the method further comprises: where the input text is in other language, obtaining the sentiment score and a rating of the user regarding the input text in the other language, combining the sentiment score and the rating of the user, and further training the FFNN model on data of the other language using the combined sentiment score and the rating of the user.

9. The method according to claim 8, wherein the input text relates to a sentence, and

the breaking up the input text into text tokens comprises: breaking up the input text into the text tokens each corresponding to a plurality of words included in the sentence.

10. The method according to claim 8 or claim 9, wherein a max pooling function is operated in a max pooling layer, and

the reading the encoded text token in a bidirectional manner comprises:

obtaining a final forward node value and a final backward node value in a final Bi-LSTM layer of the Bi-LSTM layers;

concatenating the final forward node value and the final backward node value together in the final Bi-LSTM layer of the Bi-LSTM layers;

processing the concatenated final forward node value and final backward node value in the max pooling layer; and

obtaining the vector in the max pooling layer.

11. The method according to any one of claims 8 to 10, wherein the encoding each text token comprises: encoding the each text token using a Byte Pair Encoding model.

12. The method according to claim 11 further comprising:

encoding a stop token representing an end of the input text using the Byte Pair Encoding model; and

inputting the encoded stop token into the Bi-LSTM model.

13. The method according to any one of claims 8 to 12, wherein the obtaining a probability of each sentiment score by mapping the vector into each sentiment score comprises: inputting the vector into the FFNN model configured to map the vector to the each sentiment score.

14. The method according to claim 13, wherein the FFNN model is operated in a series of FFNN layers, and a Softmax function is operated in a Softmax layer,

the obtaining a probability of each sentiment score by mapping the vector into each sentiment score comprises:

inputting the vector into the FFNN model to obtain an output in the series of FFNN layers;

inputting the output of the FFNN model into the Softmax function in the Softmax layer;

obtaining the probability of the each sentiment score in the Softmax layer; and

determining the sentiment score for the input text based on the probability in the Softmax layer.

15. A data processing apparatus configured to perform the method of any one of claims 8 to 14.

16. A computer program element comprising program instructions, which, when executed by one or more processors, cause the one or more processors to perform the method of any one of claims 8 to 14.