METHODS AND SYSTEMS FOR MULTI-INTENT RECOGNITION

TECHNICAL FIELD

The present disclosure relates generally to recognition of multi-intent in user utterances, and in particular to methods and systems of utilising a capsule neural network model to determine multiple intentions in user speech.

BACKGROUND

With the accelerated implementation of artificial intelligence technology in various industries, intelligent outbound calling systems have emerged. The use of intelligent calling systems may address issues such as insufficient manpower, and inefficiency in traditional human-operated call centers under high call volume scenarios. Intelligent calling systems are now widely used in finance, banking, e-commerce, and other fields.

Part of user interaction with intelligent calling systems may require the recognition of intent in a user's speech or utterances. The recognition of intent can be complicated by a user providing multiple intents in a single utterance. For example, a single user sentence could comprise one or more intents, or multi-intents, which may be difficult to parse or detect using current intelligent calling systems.

Existing semantic understanding technology for intelligent outbound calling is based on task-oriented dialogue scripts to recognize single intents, lacking dynamic adjustment of subsequent dialogue scripts for multiple intents of users. The long duration of understanding user intents may lead to poor user experience.

Therefore, it is desirable to provide a system and method that enhances multi-intent recognition of user utterances to address the technical disadvantages or limitations of the existing technologies or, at the very least, provide the public with a useful alternative.

SUMMARY

Embodiments of the present disclosure provide a method and system for multi-intent recognition. The systems and methods as disclose herein integrate technologies in capsule neural network model training and operation for recognition of multi-intent in user utterances. This may allow rapid and accurate understanding of user intents in a call scenario, providing an improved user experience.

In accordance with a first aspect of the present disclosure, a method for multi-intent recognition of user utterances is provided. The method comprises receiving a first multi-intent annotation dataset, training a capsule neural network model using the first multi-intent annotation dataset

to obtain a trained capsule neural network model, receiving first data to be identified, the first data comprising user utterances, inputting the first data into the trained capsule neural network model to obtain a first output result, wherein the first output result includes a first predicted user intent set, determining a current scene intent set based on the first predicted user intent set, determining whether there are dependencies between the intents in the current scene intent set to obtain a first determination result, identifying a first intent statement and a second intent statement based on the first determination result, and merging the first intent statement and the second intent statement to obtain a response statement.

In an embodiment, the step of training the capsule neural network model comprises obtaining one or more input sentence vector(s) by representing the first multi-intent annotation dataset through vectorization using a pre-trained model, extracting features from the input sentence vector(s) by sequentially passing the input sentence vectors through a convolutional layer, a primary capsule layer, a convolutional capsule layer, and a fully connected capsule layer of the capsule neural network model, simultaneously utilizing a dynamic routing mechanism to learn the mapping relationship of the capsule neural network model from low-level features to high-level features to obtain a first output capsule, outputting one or more probability of intent labels based on a norm of the first output capsule, and utilizing a margin loss function to minimize the sum of losses for the training data, wherein, after a finite number of iterations, the trained capsule neural network model is obtained.

In an embodiment, the step of obtaining one or more input sentence vectors by representing the first multi-intent annotation dataset through vectorization using a pre-trained model comprises a text encoding step, and a model reasoning step.

In an embodiment, the text encoding step comprises mapping vocabulary to indexes in the vocabulary, adding tags or identifiers to the mapped vocabulary, and generating positional encoding of the tagged and mapped vocabulary to produce encoded text data.

In an embodiment, the model reasoning step comprises inputting encoded text data into a pre-trained neural network model to calculate interference of the encoded text data, generating corresponding vectorized representations of the encoded text data, and extracting one or more selected vectorized representations of the encoded text data for further processing.

In an embodiment, the step of training the capsule neural network model further comprises extracting low-level features of images or text from the input sentence vectors via the one or more convolution layers, and generating a feature map output.

In an embodiment, the step of training the capsule neural network model further comprises converting the feature map output into one or more vector representations via the one or more primary capsule layers.

In an embodiment, the step of training the capsule neural network model further comprises processing the vector representations via the one or more convolution capsule layers to capture higher level feature relationships within the input sentence vectors.

In an embodiment, the step of training the capsule neural network model further comprises outputting one or more predicted vectors for each category of a set of categories corresponding to the one or more sentence vectors via the one or more fully connected capsule layers.

In an embodiment, the step of inputting the first data into the trained capsule neural network model to obtain a first output result comprises pre-processing the first data to be identified to convert the first data to be identified into a first sentence vector to be identified, performing deep feature extraction on the first sentence vector to be identified using the capsule neural network model, and calculating a second output capsule using the trained capsule neural network model parameters, calculating a probability of each intent label in the second output capsule using a norm of the second output capsule, obtaining a predetermined judgment threshold and determining whether the probability of each intent label in the second output capsule is greater than the predetermined judgment threshold to obtain a second judgment result, and obtaining the first predicted user intent set based on the second judgment result.

In an embodiment, the step of determining a current scene intent set based on the first predicted user intent set comprises obtaining structured information from a knowledge base, the knowledge base comprising a structured repository of knowledge information from a source, determining whether each intent in the first predicted user intent set belongs to the current scenario based on the structured information from the knowledge base, and determining if an intent in the first predicted user intent set belongs to a current scenario where if the determination is made that the intent in the first predicted user intent set does not belong to the current scenario, the intent is removed, and if the determination is made that the intent in the first predicted user intent set does belong to the current scenario, the intent is retained, and generating a current scenario intent set of intents in the first predicted user intent set that belong to the current scenario.

In an embodiment, the knowledge base is a structured repository of knowledge information that organizes textual content from a source, such as the internet, into a structural graph with entities as the basic semantic nodes.

In an embodiment, the step of identifying a first intent statement and a second intent statement based on the first determination result further comprises carrying out a judgement of dependencies between intents in the current scene intent set, wherein, if the first judgment result indicates that there are dependencies among the intents in the current scenario intent set, the method further comprises processing and storing the first intent statement according to the order of dependencies, where the first intent statement is the response statement for the last intent.

In an embodiment, the step of identifying a first intent statement and a second intent statement based on the first determination result further comprises carrying out a judgement of dependencies between intents in the current scene intent set, wherein, if the first judgment result indicates that there are no dependencies among the intents in the current scenario intent set, the method further comprises obtaining and storing the second intent statement(s), where the second intent statement(s) are the response statements for each intent.

In an embodiment, the step of merging the first intent statement and the second intent statement to obtain a response statement comprises combining the first intent statement and the second intent statement to obtain a third intent statement, obtaining a reply priority based on structured information in a knowledge base, the knowledge base comprising a structured repository of knowledge information from a source, and sorting the third intent statement according to the reply priority of each intent to obtain a reply statement.

According to a second aspect of the present disclosure, a method for multi-intent recognition of user utterances is provided. The method comprises receiving first data to be identified, the first data comprising user utterances, inputting the first data into the trained capsule neural network model to obtain a first output result, wherein the first output result includes a first predicted user intent set, determining a current scene intent set based on the first predicted user intent set, determining whether there are dependencies between the intents in the current scene intent set to obtain a first determination result, identifying a first intent statement and a second intent statement based on the first determination result, and merging the first intent statement and the second intent statement to obtain a response statement.

In an embodiment, the step of inputting the first data into the trained capsule neural network model to obtain a first output result comprises pre-processing the first data to be identified to convert the first data to be identified into a first sentence vector to be identified, performing deep feature extraction on the first sentence vector to be identified using the capsule neural network model, and calculating a second output capsule using the trained capsule neural network model parameters, calculating a probability of each intent label in the second output capsule using a norm of the second output capsule, obtaining a predetermined judgment

threshold and determining whether the probability of each intent label in the second output capsule is greater than the predetermined judgment threshold to obtain a second judgment result, and obtaining the first predicted user intent set based on the second judgment result.

In an embodiment, the step of determining a current scene intent set based on the first predicted user intent set comprises obtaining structured information from a knowledge base, the knowledge base comprising a structured repository of knowledge information from a source, determining whether each intent in the first predicted user intent set belongs to the current scenario based on the structured information from the knowledge base, determining if an intent in the first predicted user intent set belongs to a current scenario, where if the determination is made that the intent in the first predicted user intent set does not belong to the current scenario, the intent is removed, and if the determination is made that the intent in the first predicted user intent set does belong to the current scenario, the intent is retained, and generating a current scenario intent set of intents in the first predicted user intent set that belong to the current scenario.

In an embodiment, the step of identifying a first intent statement and a second intent statement based on the first determination result further comprises carrying out a judgement of dependencies between intents in the current scene intent set, where if the first judgment result indicates that there are dependencies among the intents in the current scenario intent set, the method further comprises processing and storing the first intent statement according to the order of dependencies, where the first intent statement is the response statement for the last intent.

In an embodiment, the step of identifying a first intent statement and a second intent statement based on the first determination result further comprises a carrying out a judgement of dependencies between intents in the current scene intent set, where if the first judgment result indicates that there are no dependencies among the intents in the current scenario intent set, the method further comprises obtaining and storing the second intent statement(s), where the second intent statement(s) are the response statements for each intent.

In an embodiment, the step of merging the first intent statement and the second intent statement to obtain a response statement comprises combining the first intent statement and the second intent statement to obtain a third intent statement, obtaining a reply priority based on structured information in a knowledge base, the knowledge base comprising a structured repository of knowledge information from a source, and sorting the third intent statement according to the reply priority of each intent to obtain a reply statement.

According to a third aspect of the present disclosure, a system for multi intent recognition of user utterances is provided. The system comprises a capsule neural network module configured to extract features from one or more input sentence vectors corresponding to the user utterances, the capsule neural network module comprising a convolutional layer configured to extract low-level features of images or text corresponding the one or more input sentence vectors to produce a feature map output, a primary capsule layer configured to convert the feature map output into one or more vector representations, a convolutional capsule layer configured to process the vector representation outputs, and a fully connected capsule layer configured to output predicted vectors for each category of a set of categories corresponding to the one or more sentence vectors. The system also comprises a dynamic routing module to learn the mapping relationship of the capsule neural network model from low-level features to high-level features to obtain at least a first output capsule.

In an embodiment, the capsule neural network model is configured to extract features from the input sentence vector(s) by sequentially passing the input sentence vectors through each of the layers of the capsule neural network model.

In an embodiment, the dynamic routing module is configured to simultaneously utilize a dynamic routing mechanism to learn a mapping relationship of the capsule neural network model from low-level features to high-level features to obtain at least the first output capsule.

Embodiments of the present disclosure may be provided as a network of communicating devices (i.e. a “computerized network”). Embodiments of the invention may be also provided as a software application downloadable into a computer device to facilitate the method. The software application may be a computer program product, which may be stored on a non-transitory computer-readable medium on a tangible data-storage device (such as a storage device of a server, or one within a user device).

A multi-intent recognition system and method according to the present disclosure are thus disclosed herein. Various features, aspects, and advantages of the present disclosure will become more apparent from the following detailed description of the embodiments of the present disclosure, by way of non-limiting examples only, along with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1A and Fig.1B are flow diagrams illustrating a method of multi-intent recognition, according to an embodiment of the present disclosure.

FIG. 2 is a flowchart illustrating process steps of a method of multi-intent recognition according to an embodiment of the present disclosure.

FIG. 3 is a block diagram of layers of a capsule neural network model for multi-intent recognition according to an embodiment of the present disclosure.

FIG. 4 is a flowchart illustrating process steps of a method of multi-intent recognition according to an embodiment of the present disclosure.

FIG. 5 is a flowchart illustrating process steps of a method of multi-intent recognition, according to an embodiment of the present disclosure.

FIG. 6 is a flowchart illustrating process steps of a method of multi-intent recognition, according to an embodiment of the present disclosure.

FIG. 7 is a flowchart illustrating process steps of a method of multi-intent recognition, according to an embodiment of the present disclosure.

FIG. 8 is a block diagram illustrating a technical architecture of a computing server for implementing the processes of FIGs 1-7, according to an embodiment of the present disclosure.

DETAILED DESCRIPTION

In the following, embodiments of the present invention will be described as non-limiting examples with reference to the accompanying drawings.

For purposes of brevity and clarity, descriptions of embodiments of the present disclosure are directed to a multi-intent recognition system and method in accordance with the drawings. While aspects of the present disclosure will be described in conjunction with the embodiments provided herein, it will be understood that they are not intended to limit the present disclosure to these embodiments. On the contrary, the present disclosure is intended to cover alternatives, modifications and equivalents to the embodiments described herein, which are included within the scope of the present disclosure as defined by the appended claims. Furthermore, in the following detailed description, specific details are set forth in order to provide an understanding of the present disclosure. However, it will be recognized by an individual having ordinary skill in the art, i.e. a skilled person, that the present disclosure may be practiced without specific details, and/or with multiple details arising from combinations of aspects of particular embodiments. In a number of instances, well-known systems, methods, procedures, and components have not been described in detail so as to not unnecessarily obscure aspects of the embodiments of the present disclosure.

In embodiments of the present disclosure, depiction of a given element or consideration or use of a particular element number in a particular figure or a reference thereto in corresponding descriptive material can encompass the same, an equivalent, or an analogous element or element number identified in another figure or descriptive material associated therewith.

References to “an embodiment / example”, “another embodiment / example”, “some embodiments / examples”, “some other embodiments / examples”, and so on, indicate that the embodiment(s) / example(s) so described may include a particular feature, structure, characteristic, property, element, or limitation, but that not every embodiment / example necessarily includes that particular feature, structure, characteristic, property, element or limitation. Furthermore, repeated use of the phrase “in an embodiment / example” or “in another embodiment / example” does not necessarily refer to the same embodiment / example.

The terms “comprising”, “including”, “having”, and the like do not exclude the presence of other features / elements / steps than those listed in an embodiment. Recitation of certain features / elements / steps in mutually different embodiments does not indicate that a combination of these features / elements / steps cannot be used in an embodiment.

As used herein, the terms “a” and “an” are defined as one or more than one. The use of “/” in a figure or associated text is understood to mean “and/or” unless otherwise indicated. The term “set” is defined as a non-empty finite organization of elements that mathematically exhibits a cardinality of at least one (e.g. a set as defined herein can correspond to a unit, singlet, or single-element set, or a multiple-element set), in accordance with known mathematical definitions. The terms “first”, “second”, etc. are used merely as labels or identifiers and are not intended to impose numerical requirements on their associated terms.

As used in the present disclosure, the terms “component”, “module”, “system”, “interface” and the like are generally intended to refer to a computer-related entity, either hardware, a combination of hardware and software, software, or software in execution. For example, a module may be, but is not limited to being, a process running on a processor, a processor, an object, an executable, a thread of execution, a program, and/or a computer. By way of illustration, both an application running on a controller and the controller can be a component. One or more modules may reside within a process and/or thread of execution and a module may be localized on one computer and/or distributed between two or more computers. As another example, an interface can include I/O components as well as associated processor, application, and/or API components. In the context of the present disclosure, the multi-intent recognition systems and methods, and their constituent parts, may be implemented as hardware, software, or a combination thereof.

Furthermore, various embodiments of the present disclosure may be implemented as a method, apparatus, or article of manufacture using standard programming and/or engineering techniques to produce software, firmware, hardware, or any combination thereof to control a computer to implement the disclosed subject matter. For instance, the claimed subject matter may be implemented as a computer-readable medium embedded with a computer executable program, which encompasses a computer program accessible from any computer-readable storage device or storage media. For example, computer readable media can include but are not limited to magnetic storage devices (e.g., hard disk, floppy disk, magnetic strips, etc.), optical disks (e.g., compact disk (CD), digital versatile disk (DVD), etc.), smart cards, and flash memory devices (e.g., card, stick, key drive, etc.).

As used herein, the term "multi-intent recognition" refers to the recognition of multiple intentions within user utterances. Part of user interaction with intelligent calling systems may require the recognition of intent in a user's speech or utterances. The recognition of intent can be complicated by a user providing multiple intents in a single utterance. For example, a single user sentence could comprise one or more intents, or multi-intents, which may be difficult to parse or detect using current intelligent calling systems. Multi-intent recognition may be applied in scenarios such as intelligent customer service, voice assistants, smart home control, etc., where users may express multiple needs or requests in a single utterance. Multi-intent recognition should be as fast and accurate as possible to minimize user waiting time and thereby improve user satisfaction.

As used herein, the term "knowledge base" may refer to a structured repository of knowledge information that organizes textual content from a source, such as the internet, into a structural graph with entities as the basic semantic nodes.

As used herein, the term "norm" or "vector norm" may refer to a function that may quantify the magnitude of a vector. The norm (or magnitude) of an output vector from a capsule may be used as a measure of a certainty or confidence that a feature represented by a capsule is present in input data. In an example embodiment of a capsule neural network as described herein, a length of a capsule output vector may be used to determine a probability of the existence of an object or feature. In an example embodiment of a multi-intent recognition process such as that disclosed herein, the term "norm" may refer to a set of standards and rules for processing input data, model training, prediction output, and result processing.

In an example embodiment, a Data Pre-processing Norm may comprise conducting pre-processing steps such as cleaning, tokenization, and vectorization on input data to ensure data quality and compliance with model input requirements.

In an example embodiment, a Model Training Norm may comprise the use of annotated multi-intent labelled datasets to train the capsule network model, including data input, model parameter settings, iteration counts, loss function selection, etc.

In an example embodiment, a Prediction Output Norm may comprise inputting data to be recognized into a trained model for prediction, outputting a predicted user intent set, and further processing the output results based on a knowledge base and decision thresholds.

In an example embodiment, a Result Processing Norm may comprise determinations according to a predicted user intent set and structured information in a knowledge base. For example: to determine whether each intent in a plurality of user intents belongs to a current scenario, to handle dependencies between intents, and to merge intent statements to generate a final response.

Currently, there are two main existing technologies for multi-intent recognition: one is to convert multi-intent recognition into multiple single-intent recognitions, such as dividing user input sentences into multiple sub-sentences through syntactic analysis or sentence segmentation models, and then inputting each sub-sentence into a single-intent recognition model to identify the intent of each sub-sentence; the other is to use an encoder or pre-trained model to convert user input text into sentence vectors, and then treat the multi-intent recognition problem as a multi-label classification problem, using deep learning multi-label classification models to identify multiple intents of users. Some differences between these implementations and the invention as disclosed herein include:

Model Structure: Multi-label classification methods typically use traditional machine learning models or deep learning models (such as neural networks) to process text data and treat the multi-intent recognition problem as a combination of multiple binary classification problems. In contrast, the invention as disclosed herein may employ a capsule network model, sucha s that disclosed herein, that may preserve the positional relationship between low-level and high-level features through a dynamic routing mechanism, enabling more accurate text modeling.

Feature Representation: Multi-label classification methods generally convert text data into feature representations such as bag-of-words models, TF-IDF vectors, or word embeddings. In contrast, the invention as disclosed herein may use a capsule neural network model, such as that disclosed herein, to perform deep feature extraction on input sentence vectors, leveraging the vector output and dynamic routing mechanism of capsules to capture complex relationships in text.

Intent Processing: Multi-label classification methods may directly output a prediction probability for each intent and determine the presence of intents based on thresholds. In contrast, after predicting intents, the invention as disclosed herein may further process the intents based on a knowledge base, and dependencies between intents, as described herein, to generate responses that are more in line with, or accurate to, actual scenarios.

Fig. 1A is a flow chart illustrating a process flow 100 of a training process of a capsule neural network model according to an embodiment of the present disclosure. The process may comprise step 102 of obtaining a first multi-intent annotation dataset, and step 104, where the first multi-intent annotation dataset is input into a capsule neural network model to train the capsule neural network model.

Fig. 1B is a flow chart illustrating a process flow 102 for multi-intent recognition according to an embodiment of the present disclosure. The process 102 may comprise receiving, at step 112, first data to be identified, where the first data comprises user utterances or speech. At step 114, the process 102 may input the first data into the trained capsule neural network model (which may be obtained via process 100 as illustrated in FIG. 1A) to obtain a first output result, where the first output result includes a first predicted user intent set that may be recognized or obtained.

At step 116, the process 102 may determine or calculate or obtain a current scene intent set based on the first predicted user intent set from step 114.

At step 118, the process 102 may further determine whether there are dependencies between the intents in the current scene intent set to obtain a first determination result.

At step 120, the process 102 may obtain a first intent statement and a second intent statement based on the first determination result from step 118.

Finally, at step 122, the process 102 may merge the first intent statement and the second intent statement to obtain a response statement, which may comprise a suitable response to the user utterances or speech obtained at step 112 as part of the data to be identified.

Turning back to FIG.1A, the process 100 for training the capsule neural network model may further comprise step 105, where the first multi-intent annotation dataset is represented through vectorization using a pre-trained model to obtain one or more input sentence vectors. At step 106, the process 100 may extract features from the input sentence vector(s) by

sequentially passing the input sentence vectors through a convolutional layer 106a, a primary capsule layer 106b, a convolutional capsule layer 106c, and a fully connected capsule layer 106d of the capsule neural network model . Simultaneously, at step 107, the process 100 may utilize a dynamic routing mechanism to learn the mapping relationship of the capsule neural network model from low-level features to high-level features to obtain a first output capsule.

The output of steps 106 and 107 is shown at step 108, outputting one or more probability of intent labels based on a norm of the first output capsule from step 107. The norm of a capsule output may refer to the length of the vector produced by the capsule, which may reflect the confidence of the capsule's prediction.

The process 100 may further utilize a margin loss function at step 109, to minimize the sum of losses for the training data. It will be understood that, after a finite number of iterations of steps 105 to 109, a trained capsule neural network model may be obtained, such as a capsule neural network model as described herein, capable of multi-intent recognition of user utterances.

FIG. 2 is a flowchart illustrating a process 200 for generating or producing vectorized representations of a first multi-intent annotation dataset, according to an embodiment of the present disclosure. In an example embodiment, the vectorized representations of a first multi-intent annotation dataset may be used to train or pre-train a capsule neural network model suitable for multi-intent recognition as described herein, for example at step 105 in process 100 shown in FIG.1A. The vectorized representations may be produced by a pre-trained neural network model, such as BERT or ALBERT, or any other suitable natural language processing machine learning framework.

At step 202, the process 200 may include text encoding, for example to encode a cleaned text data version of the first multi-intent dataset, using a pre-trained tokenizer. This may allow conversion into an input format that a neural network model can utilize as an input.

The encoding process 202 may include the steps of mapping vocabulary to indexes in the vocabulary 204, adding specific tags or identifiers 206 (such as [CLS], [SEP], etc.), and generating positional encoding 208. The step 202 of text encoding may provide for encoded or input text data that a neural network model can utilize.

At step 212 the process 200 may include model reasoning. The model reasoning step may include inputting the encoded text data into a pre-trained neural network model for inference calculation at step 214. The model may then generate corresponding vectorized

representations at step 216, based on the encoded or input text data produced at step 202, which contains semantic information of the text.

At step 222, the process 200 may extract selected vectorized representations from the vectorized representations produced at step 216. The step 222 may comprise extracting a required vectorized representation from an output of the model. For pre-trained models such as BERT and ALBERT, the output of the hidden layer (such as the hidden layer corresponding to the [CLS] tag) may be extracted as the vectorized representation of the text.

FIG. 3 is a diagram illustrating the layers that may form a capsule neural network model 300 according to an embodiment of the present disclosure. In an example embodiment, the capsule neural network model 300 may comprise layers 302 to 308, through which input sentence vectors may be passed during a training of the capsule neural network model, as set out in process steps 106 to 108 of process 100 as described in relation to Fig. 1A. Simultaneously, the process may utilize dynamic routing mechanisms to learn a mapping relationship of the capsule neural network model 300 from low to high level features, to obtain output capsules, as set out in process step 107 of process 100 described in Fig. 1A.

The capsule neural network model 300 may consist of convolutional layers 302, primary capsule layers 304, convolutional capsule layers 306, and fully connected capsule layers 308.

As described above, the training process of capsule neural network model 300 may extract features from an input sentence vector by sequentially passing the input sentence vector through a convolutional layer 302, a primary capsule layer 304, a convolutional capsule layer 306, and a fully connected capsule layer 308 of the capsule neural network model 300. Simultaneously, a dynamic routing mechanism may be utilized to learn the mapping relationship of the capsule layers from low-level features to high-level features, which may result in the first output capsule.

Training the capsule neural network model 300 may initially require representing the first multi-intent annotation dataset through vectorization using a pre-trained model, such as a machine learning (ML) framework for natural language processing, which can be Bert, AlBert, etc., as set out in Fig.2 and described herein. The vectorized representation of the first multi-intent annotation dataset may then serve as an input sentence vector for the capsule neural network model 300.

The sentence vector may be sequentially passed through a convolutional layer 302, a primary capsule layer 304, a convolutional capsule layer 306, and a fully connected capsule layer 308

of the capsule neural network model 300 for feature extraction from shallow to deep or low to high levels.

Simultaneously, a dynamic routing mechanism may be employed to learn the mapping relationship from low-level or shallow features to high-level or deep features, which may result in an output capsule, i.e., the first output capsule.

Finally, the norm of the output capsule, which may be the vector length, may be used to calculate the probability of the output intent labels.

Additionally, a margin loss function may be utilized to minimize the sum of losses for the training data.

After a finite number of iterations, a trained capsule neural network model 300 may be obtained. Put another way, after completing the above steps, the training of a capsule neural network model 300 may be achieved, which may enhance the reliability and accuracy of the model's output results.

The aforementioned first multi-intent annotation dataset may refer to annotated data encompassing various intents, including multiple intents expressed in user input utterances. For instance, the user utterance "This is my bank card, but I don't use this card anymore" conveys two intents: "I have this card" and "I don't use this card anymore". Furthermore, the first multi-intent annotation dataset can serve as a small training annotation dataset.

The capsule network 300 may replace a scalar output feature detector of traditional Convolutional Neural Networks (CNNs) with vector output capsules, and may substitute pooling operations with dynamic routing. Compared to traditional CNNs, Capsule Networks, such as capsule neural network model 300 as described herein, may preserve the precise positional relationships between low-level and high-level features through a dynamic routing process, which may advantageously enable more accurate modeling of text. The dynamic routing mechanism may primarily address the issue of determining the appropriate weights for connections from low-level capsules to high-level capsules. By inputting a first multi-intent annotation dataset into the capsule neural network model 300 as described herein, the model may be trained, which may allow for effective utilization of training data for model training even with small datasets. This may enable multi-intent recognition in user utterances through the capsule neural network model 300 as described herein, reducing the cost of manual annotation.

As set out herein, the capsule neural network model 300 may consist of convolutional layers 302, primary capsule layers 304, convolutional capsule layers 306, and fully connected

capsule layers 308. A training process, such as that set out in process 100 in Fig. 1A, is described in more detail below with reference to Fig. 3:

At step 106a, the process 100 may utilize one or more convolutional layers 302 to extract low-level features of images or text from an input vectorized first multi-intent annotation dataset. The step 106a of processing using one or more convolution layers may output a feature map output.

At step 106b, the process 100 may utilize one or more primary capsule layers 304 to convert the feature map output by the convolutional layer 302 into vector representations, which may represent different entities or features within the feature map output of the input vectorized first multi-intent annotation dataset.

At step 106c, the process 100 may utilize one or more convolutional capsule layers 306 to further process the vector representation outputs from the primary capsule layers 304. This may allow capture of higher-level feature relationships within the input vectorized first multi-intent annotation dataset.

At step 106d, the process 100 may utilize a fully connected capsule layer 308, which may ultimately output predicted vectors for each category, with the magnitude of the vectors representing the probability of belonging to that category. In an example embodiment, the category may comprise one or more categories associated with the one or more input sentence vectors corresponding to user utterances.

In addition to the above process, additional processes for capsule neural network model 300 training may include:

Define loss function:

For multi intent classification tasks, margin loss can be used to measure a difference between predicted vectors and true labels. If the model also includes a reconstruction network (used to reconstruct input data to assist training), reconstruction loss (such as mean square error loss) can also be incorporated.

Iterative training:

Input the vectorized dataset into the capsule neural network model 300 and calculating the prediction vector and loss may be accomplished through forward propagation. Further, a backpropagation algorithm may be used to calculate gradients and update model parameters to minimize losses. This process may be repeated until performance of the model 300 on a validation dataset no longer significantly improves or reaches a preset number of iterations.

Dynamic routing algorithm:

In capsule networks, such as capsule neural network 300, dynamic routing algorithms may be used to transfer information between primary capsule layers 304 and convolutional capsule layers 306, as well as between convolutional capsule layers 306 and fully connected capsule layers 308. A dynamic routing algorithm may iteratively calculate and update a coupling coefficient between low-level capsules and high-level capsules to ensure that each high-level capsule can receive correct information from the relevant low-level capsules.

The process 100 may comprise different implementations of routing mechanisms. The core of a capsule network 300 lies in its routing mechanism, which determines the connection weights between capsules 302 to 308 through dynamic routing algorithms, which may be learned at step 107 as set out in Fig. 1A. In addition to basic dynamic routing algorithms, other routing mechanisms such as EM routing and attention routing may be utilized to achieve more efficient and accurate information transfer between capsules.

Differences in network depth:

Capsule networks, such as capsule neural network model 300, may adjust their depth according to task requirements. Deeper networks may capture more complex feature representations, but may also lead to increased computational complexity and training difficulty. Therefore, in practical applications, it may be necessary to choose an appropriate network depth according to a specific situation.

Different capsule dimensions:

The vector dimension output by each capsule 302 to 308 in capsule neural network 300 may be important hyperparameters that determines the complexity of the features that the capsule can represent. For example, different tasks carried out by model 300 may require capsules of different dimensions to represent features.

Integration with other neural network architectures:

Capsule networks, such as capsule neural network model 300, may be combined with other neural network architectures such as convolutional neural networks, recurrent neural networks, etc., to achieve improved performance. For example, convolutional layers may be added before the capsule network 300 to extract finer features, or fully connected layers may be added after the capsule network 300 for more complex classification or regression tasks.

Different training strategies:

The training process of capsule networks, such as capsule neural network 300, may be relatively complex, and may require careful design of loss functions and optimization algorithms. In addition to a basic marginal loss function, other loss functions and optimization strategies may be included to accelerate the training process and improve the performance of the model 300.

The process 100 of Fig. 1A may determine the capsule neural network model 300 of Fig. 3 as trained in several ways. Some example embodiments are set out below:

Loss function is a method to determine if a model, such as capsule neural network model 300, is sufficiently or well trained. This method may include monitoring one or more loss function values during the training process, such as training process 100 as described in Fig. 1A. In an example embodiment, a stop criteria may be reached when a loss function value as described herein no longer significantly decreases, or begins to fluctuate. In this scenario, it can be considered that the model 300 has converged, and training may be stopped at this corresponding point.

Verification set error is a method to divide the dataset into a training set and a validation set. The method may comprise regularly evaluating the performance of the model, such as capsule neural network model 300, during processing of the validation set during the training process. Further, the method may observe changes in validation set errors during the training process on the validation data set. In an example embodiment, a stop criteria may be reached when the validation set error no longer significantly decreases, or begins to fluctuate. In this scenario, it can be considered that the model 300 has reached optimal performance, and training can be stopped at this point.

Overfitting detection is a method to detect overfitting by the model, such as capsule neural network model 300, by monitoring a performance difference of the model 300 on a training data set and a validation data set. In an example embodiment, a stop criteria may be reached when the performance of the validation data set being processed by the model no longer improves or begins to decline, while the performance of the training set being processed by the model continues to improve. In this scenario, it can be considered that the model 300 is overfitting, and training may be stopped at this corresponding point.

Early Stopping Method is an automatic stop strategy based on validation set performance, or performance of processing of a validation data set by a model, such as capsule neural network model 300. In an example embodiment, a value may be set as a threshold or a number of consecutive unimproved epochs. In an example embodiment, a stop criteria may be reached when a performance of a validation set being processed by the model 300 does not improve

for multiple consecutive epochs, such as 10 consecutive epochs without reaching an optimal validation set accuracy. In, this scenario, it may be considered that the model has reached its optimal performance, or a sufficient performance, and training can be stopped at this point.

Regularization techniques may be a method of determining a model, such as capsule neural network model 300, is sufficiently trained by introducing regularization terms in a loss function to limit a size of model parameters. In this way, the complexity of the model 300 can be controlled. In an example embodiment, a stop criteria may be reached when an impact of regularization on the loss function gradually increases. In this scenario, it can be considered that the model has reached a balance between optimal performance and complexity, and model training can be stopped at this point.

Cross validation may be a method to divide a dataset into multiple subsets, and conduct multiple training and validation processes on the multiple data subsets with a model, such as capsule neural network model 300. In an example embodiment, a stop criteria may be reached when the model 300 performs well (or to a satisfactory standard for the requirements of the model), in cross validation without overfitting. In this scenario, it can be considered that the model has been trained.

Evaluation indicators may use various evaluation metrics to assess a performance of the model, such as capsule neural network model 300. The evaluation metrics may include accuracy, recall, F1 score, etc. In an example embodiment, a stop criteria may be reached when the model 300 performs well and tends to stabilize on the evaluation metrics. At this point, it can be considered that the model has been trained.

Iteration times and computing resources may be used to identify when a model, such as capsule neural network model 300, is suitable trained. For example, the maximum number of iterations may be set or pre-set, or it may be determined when to stop training based on a usage of computing resources by the model 300. In an example embodiment, a stop criteria may be reached when the preset maximum number of iterations is reached, or when computing resources are exhausted, and the model training may be correspondingly stopped or completed. Whilst it will be understood that this method may not be the optimal choice based on maximizing model performance, it may be useful for time or resource limited applications.

A convergence situation may be used to observe a convergence of the model, such as capsule neural network model 300, to identify when a model is suitably trained. In an example embodiment, a stop criterion may be reached when a loss function value of the model 300

tends to stabilize and no longer decreases. In this scenario, it can be considered that the model has converged and may be sufficiently trained.

Fig. 4 illustrates an example embodiment of a process 400 for inputting first data to be identified into a trained Capsule neural network model for prediction to obtain a first output result, wherein the first output result includes a first predicted user intent set. In an example embodiment, process step 114 in Fig. 1B may comprise at least one of the steps of process 400 as described herein.

In an example embodiment, the input information of the capsule neural network model, such as model 300, may need to be in vector form. In this case, it may be necessary to preprocess the first data to be identified, converting each intent text in the first data to be identified into a first sentence vector to be identified. Deep feature extraction may then be performed on the first sentence vector using the capsule network, and the output capsule may be calculated using the trained model parameters. The output capsule may comprise a second output capsule. Finally, a probability of each intent label may be calculated using a norm of the output capsule, and a judgment threshold may be set to obtain the predetermined judgment threshold. The systems and methods as described herein may then determine whether the probability is greater than the predetermined judgment threshold, and the result of the determination may be the second judgment result. Based on the second judgment result, the first predicted user intent set may be obtained, which is the user intent set predicted by the capsule neural network model.

The process 400 may include step 402, to preprocess the first data to be identified to convert the first data to be identified it into a first sentence vector to be identified. In an example embodiment, the first data to be identified may be data obtained or received at step 112 as shown in Fig. 1B and described herein.

At step 404, the process 400 may perform deep feature extraction on the first sentence vector to be identified using the capsule neural network model, such as trained capsule neural network model 300 as described in Fig. 3. Further, at step 404 the process may then calculate a second output capsule using the trained capsule neural network model parameters.

At step 406, the process 400 may calculate a probability of each intent label in the second output capsule from step 404. In an example embodiment, probability may be calculated using a norm of the second output capsule.

At step 408, the process 400 may obtain a predetermined judgment threshold, which may be used to filter intent labels during model predictions as described herein. The pre-determination of the judgement or decision threshold may involve one or more of the following steps:

Initial Setting: During an initial setting step, the model may set an initial decision threshold based on experience or preliminary experimental results. In an example embodiment, this value may be 0.5 (i.e., a 50% probability), however it will be understood that the specific value will be adjusted according to the actual conditions to which the model 300 is experiencing.

Model Training and Validation: During model training, the model may use a validation data set as described herein to evaluate the model 300. The evaluation may include observing model performance (such as accuracy, recall, F1 score, etc.) under different decision thresholds.

Performance Tuning: During a performance turning step, the decision threshold may be gradually adjusted based on the output evaluation results on the validation data set as outlined above. In this way, a threshold may be determined that optimizes model performance. It will be understood that this process may be an iterative process as described herein, and may require multiple attempts and validations.

Final Determination: During a final determination step, after model training is completed or determined to be complete, a final decision threshold may be determined based on optimal performance of the model on the validation data set. The output threshold may then be used during step 408 to filter intent labels during model predictions as outlined herein.

At step 410, the process 400 may determine if the probability calculated at step 406 is greater than the predetermined judgment threshold obtained at step 408, to obtain a second judgment result.

At step 412, the process 400 may therefore obtain a first predicted user intent set based on the second judgment result of step 410.

Fig. 5 illustrates an example embodiment of a process 500 for obtaining a current scene intent set based on the first predicted user intent set. In an example embodiment, process step 116 in Fig. 1B may comprise at least one of the steps of process 500 as described herein.

In an example embodiment, a knowledge base as described herein may be a structured repository of knowledge information that organizes textual content from a source, such as the internet into a structural graph with entities as the basic semantic nodes. The knowledge base may enable machines to understand and search for keywords, thereby facilitating direct

access to answers through search. Based on the structured information from the knowledge base, it can be determined whether each intent in a first predicted user intent set belongs to a current scenario.

In an example embodiment, if an intent in a first predicted user intent set belongs to a current scenario, it may be retained. Similarly, if an intent does not belong to a current scenario, it may be removed, thereby obtaining a current scenario intent set, or a set of intents determined to belong to a current scenario.

The process 500 of Fig. 5 may comprise the following steps:

At step 502, the process 500 may obtain structured information from a knowledge base. As outlined herein, the knowledge base may be a structured repository of knowledge information that organizes textual content from a source, such as the internet into a structural graph with entities as the basic semantic nodes.

At step 504, the process 500 may determine whether each intent in a first predicted user intent set belongs to a current scenario based on structured information from the knowledge base. In an example embodiment, the first predicted user intent set may comprise the set obtained at step 412 of Fig. 4. In this scenario, the process 500 may consider the structured information from the knowledge base, and determine if each intent in the set belongs to a current scenario or scene.

At step 506, the process 500 may determine if an intent in the first predicted user intent set does not belong to the current scenario. If the determination is made that the intent in the first predicted user intent set does not belong to the current scenario, the intent may be removed.

At step 508, the process 500 may determine if an intent in the first predicted user intent set belongs to the current scenario. If the determination is made that the intent in the first predicted user intent set does belong to the current scenario, the intent may be retained.

In this way, at step 510 a current scenario intent set may be obtained, comprising intents from the first predicted user intent set that do belong to the current scenario, and compiling these to form a current scenario intent set.

The abovementioned first data to be identified, for example as obtained at step 112 of Fig. 1B, may refer to any data information that needs to be identified, such as confirming user identity or verifying a user's bank card number. The first data to be identified may be input into the trained Capsule neural network model 300 for prediction, yielding a first output result. The first output result may include a first predicted user intent set, which in an example embodiment

may represent all predicted user intents. Based on the first predicted user intent set, a current scenario intent set may be obtained. The current scenario intent set may refer to scenarios where the current user intent set is applied, such as practical application scenarios in finance, banking, e-commerce, and other domains. By obtaining the scenario intent set, user intent understanding within the current scenario may be facilitated, which can expedite the speed of understanding user intents and improve the accuracy of understanding.

Fig. 6 illustrates an example embodiment of a process 600 to obtain a first set of determination results, and for obtaining intent statements based on the first set of determination results. In an example embodiment, process step 118 in Fig. 1B may comprise at least one of the steps of process 600 as described herein.

In an example embodiment, the process 600 may judge whether there are dependencies among the intents, for example the intents as produced at step 510 in Fig. 5, to obtain a first judgment result. In this context, dependency may refer to a situation where a response statement of one intent is dependent on a response statement of another intent.

Based on the first judgment result, the process 600 may obtain a first intent statement and a second intent statement, respectively. That is, when a first judgment result indicates that there are dependencies among the intents in the current scenario intent set (e.g. the set as produced at step 510 in Fig. 5), the process 600 may extract and store the first intent statement according to an order of dependencies. In this case, a first intent statement may be the response statement for the last intent.

To provide an example, after a system confirms a bank card number, a user may say, "This is my bank card, but I don't use this card anymore." In this case, the system and methods as described herein may use the capsule neural network 300 to identify that the user's utterance expresses two intents: "I have this card" and "I don't use this card anymore." Both intents belong to the same scenario (e.g., referring to a bank or payment card), and the intent "I don't use this card anymore" is dependent on the intent "I have this card." Therefore, there is a dependency between the intents, and in this case the systems and methods as described herein may directly process and store the first intent statement, which is the intent "I don't use this card anymore."

If the first judgment result indicates that there are no dependencies among the intents in the current scenario intent set, the process 600 may obtain and store the second intent statements, where the second intent statements may be the response statements for each intent.

To provide an example, after a system confirms a bank card number, a user may say, "Who are you? This is my card. What's up?" In this case, the systems and methods as described herein may use the Capsule Network 300 to identify that the user's utterance expresses two intents: i.e. "asking for the other party's identity" and "I have this card." According to structured information, although the intent "asking for the other party's identity" is not in this process, it also belongs to the same scenario, and there is no dependency between the two intents. Therefore, the systems and methods as disclosed herein may obtain and store the second intent statements. This may implement a multi-intent processing strategy, laying the foundation for subsequent dialogue flow adjustments.

Therefore, in an example embodiment, the process 600 of Fig. 6 may comprise the following steps:

At step 602, the process 600 may determine whether there are dependencies between the intents in the current scene intent set to obtain a first determination result. In an example embodiment, process step 118 in Fig. 1B may comprise process step 602 as described herein.

At step 604, the process 600 may obtain a first intent statement and a second intent statement based on the first determination result. In an example embodiment, process step 120 in Fig. 1B may comprise at least one of the steps of process 604 as described herein. The process step 604 may comprise the following steps:

At step 606, the process 600 may determine if there are dependencies among intents in the current scenario intent set. This may be via a first judgement result to indicate any dependencies as described herein.

At step 608, if the first judgment result at 606 indicates that there are dependencies among the intents in the current scenario intent set, i.e. “Yes”, the step 608 may process and store the first intent statement according to the order of dependencies, where the first intent statement is the response statement for the last intent.

Alternatively, at step 610, if the first judgment result at 606 indicates that there are no dependencies among the intents in the current scenario intent set, i.e. “No”, the step 610 may obtain and store the second intent statements, where the second intent statements are the response statements for each intent.

Fig. 7 illustrates an example embodiment of a process 700 for merging the first intent statement and the second intent statement output from process 600 (shown in Fig. 6) to obtain

a response statement. In an example embodiment, process step 122 in Fig. 1B may comprise at least one of the steps of process 700 as described herein.

In an example embodiment, a reply statement may be obtained by combining the first intent statement and the second intent statement, such as those obtained at steps 608 and 610 of Fig. 6. In an example embodiment, the systems and methods as described herein may combine the first intent statement and the second intent statement to obtain a third intent statement, and may obtain a reply priority based on structured information in the knowledge base. Further, the systems and methods as described herein may sort the intents according to their reply priority, and output them sequentially, which may therefore form a reply statement.

By designing a multi-intent processing strategy, the systems and methods as described herein can dynamically adjust a subsequent dialogue flow based on the multiple intents provided by the user. This may allow reduction in a number of dialogue rounds with a user, thereby improving human-computer interaction efficiency and enhancing user experience.

Therefore, in an example embodiment the process 700 of Fig. 7 may comprise the following steps:

At step 702, the process may comprise merging the first intent statement and the second intent statement to obtain a response statement. The process step 702 may include:

At step 704, the process 700 may combine a first intent statement and a second intent statement to obtain a third intent statement. In an example embodiment, the first intent statement and the second intent statement may be output from process 600 (shown in Fig. 6) at steps 608 and 610.

At step 706, the process may obtain a reply priority based on structured information in the knowledge base. As described herein, a knowledge base may be a structured repository of knowledge information that organizes textual content from a source, such as the internet into a structural graph with entities as the basic semantic nodes. The knowledge base may enable machines to understand and search for keywords, thereby facilitating direct access to answers through search. Based on the structured information from the knowledge base, a reply priority for the intents can be determined.

At step 708, the process 700 may sort the third intent statement according to the reply priority of each intent from step 706, to obtain a reply statement comprising suitable replies for each intent. In this way, multi-intent of user utterances may be recognized, and a suitable reply to the multi-intents may be provided.

FIG. 8 is a block diagram illustrating a technical architecture 800 of a computing server capable of implementing the multi-intent recognition of FIGs 1-7, according to an embodiment of the present disclosure. The technical architecture 800 represents a computer system suitable for executing one or more embodiments of the invention. While a single computing server is shown, the methods may be implemented across multiple computers in a distributed computing environment.

The technical architecture 800 includes a processor 822 (which may be referred to as a central processor unit or CPU) that is in communication with memory devices including secondary storage 824 (such as disk drives), read only memory (ROM) 826, random access memory (RAM) 828. The RAM 828 may be partitioned to efficiently process different tasks, such as data pre-processing, feature engineering, model training, and evaluation. The partitioning of RAM 828 allows for efficient processing of large-scale data processing tasks, which is particularly beneficial for complex predictive modelling operations. The processor 822 may be implemented as one or more CPU chips. The technical architecture may further comprise input/output (I/O) devices 830, and network connectivity devices 832.

The secondary storage 824 is typically comprised of one or more disk drives or tape drives and is used for non-volatile storage of data and as an over-flow data storage device if RAM 828 is not large enough to hold all working data. Secondary storage 824 may be used to store programs which are loaded into RAM 828 when such programs are selected for execution.

In this embodiment, the secondary storage 824 includes an analytics or predictive component 824a comprising non-transitory instructions operative by the processor 822 to perform various operations of the multi-intent recognition method as disclosed herein. For example, the secondary storage 824 may comprise an analytics or predictive component 824a to implement the functionality of the capsule neural network model 300 as described herein.

The ROM 826 is used to store instructions and perhaps data which are read during program execution. The secondary storage 824, the RAM 828, and/or the ROM 826 may be referred to in some contexts as computer readable storage media and/or non-transitory computer readable media.

I/O devices 830 may include printers, video monitors, liquid crystal displays (LCDs), plasma displays, touch screen displays, keyboards, keypads, switches, dials, mice, track balls, voice recognizers, card readers, paper tape readers, or other well-known input devices.

The processor 822 executes instructions, codes, computer programs, scripts which it accesses from hard disk, floppy disk, optical disk (these various disk-based systems may all be considered secondary storage 824), flash drive, ROM 826, RAM 828, or the network connectivity devices 832. While only one processor 822 is shown, multiple processors may be present. Thus, while instructions may be discussed as executed by a processor, the instructions may be executed simultaneously, serially, or otherwise executed by one or multiple processors.

Although the technical architecture is described with reference to a computer, it should be appreciated that the technical architecture may be formed by two or more computers in communication with each other that collaborate to perform a task. For example, but not by way of limitation, an application may be partitioned in such a way as to permit concurrent and/or parallel processing of the instructions of the application. Alternatively, the data processed by the application may be partitioned in such a way as to permit concurrent and/or parallel processing of different portions of a data set by the two or more computers. In an embodiment, virtualization software may be employed by the technical architecture 800 to provide the functionality of a number of servers that is not directly bound to the number of computers in the technical architecture 800. In an embodiment, the functionality disclosed above may be provided by executing the application and/or applications in a cloud computing environment. Cloud computing may comprise providing computing services via a network connection using dynamically scalable computing resources. A cloud computing environment may be established by an enterprise and/or may be hired on an as-needed basis from a third-party provider.

The technical architecture 800 may leverage parallel processing systems, such as CUDA, for model computation acceleration. This may include targeted compilation of the required operations into a structure that can be run by the parallel processing system. For example, the system can utilize GPU acceleration to enhance the speed and efficiency of model training and evaluation, particularly for computationally intensive tasks in risk control modelling. Additionally, the system supports containerized deployment using technologies like Docker, enabling distributed computing for risk control management with advantages of high availability, reliability, and scalability. This combination of parallel processing and containerized deployment allows the system to efficiently handle large volumes of data and complex models, making it suitable for real-time multi-intent recognition of user utterances. The parallel processing capabilities are particularly beneficial for tasks such as model training and evaluation, where multiple model variations can be tested simultaneously.

By programming and/or loading executable instructions onto the technical architecture, at least one of the CPU 822, the RAM 828, and the ROM 826 are changed, transforming the technical architecture in part into a specific purpose machine or apparatus having the novel functionality taught by the present disclosure. It is fundamental to the electrical engineering and software engineering arts that functionality that can be implemented by loading executable software into a computer can be converted to a hardware implementation by well-known design rules. Such hardware implementations may include field-programmable gate arrays (FPGAs) or application-specific integrated circuits (ASICs), which can further enhance the system's performance for specific predictive modelling and risk control tasks.

In the foregoing detailed description, embodiments of the present disclosure in relation to a multi-intent recognition system and method are described with reference to the provided figures. The description of the various embodiments herein is not intended to call out or be limited only to specific or particular representations of the present disclosure, but merely to illustrate non-limiting examples of the present disclosure. The present disclosure serves to address at least one of the mentioned problems and issues associated with the prior art. Although only some embodiments of the present disclosure are disclosed herein, it will be apparent to a person having ordinary skill in the art in view of the present disclosure that a variety of changes and/or modifications can be made to the disclosed embodiments without departing from the scope of the present disclosure. It will be understood by those skilled in the art that many variations of the embodiments can be made within the scope of the following claims. Moreover, features of one or more embodiments may be mixed and matched with features of one or more other embodiments. Therefore, the scope of the disclosure as well as the scope of the following claims is not limited to embodiments described herein.

CLAIMS

1. A method for multi-intent recognition of user utterances, the method comprising:

receiving a first multi-intent annotation dataset;

training a capsule neural network model using the first multi-intent annotation dataset to obtain a trained capsule neural network model;

receiving first data to be identified, the first data comprising user utterances;

inputting the first data into the trained capsule neural network model to obtain a first output result, wherein the first output result includes a first predicted user intent set;

determining a current scene intent set based on the first predicted user intent set;

determining whether there are dependencies between the intents in the current scene intent set to obtain a first determination result;

identifying a first intent statement and a second intent statement based on the first determination result; and

merging the first intent statement and the second intent statement to obtain a response statement.

2. The method of claim 1, wherein the step of training the capsule neural network model comprises:

obtaining one or more input sentence vector(s) by representing the first multi-intent annotation dataset through vectorization using a pre-trained model;

extracting features from the input sentence vector(s) by sequentially passing the input sentence vectors through a convolutional layer, a primary capsule layer, a convolutional capsule layer, and a fully connected capsule layer of the capsule neural network model;

simultaneously utilizing a dynamic routing mechanism to learn the mapping relationship of the capsule neural network model from low-level features to high-level features to obtain a first output capsule;

outputting one or more probability of intent labels based on a norm of the first output capsule; and

utilizing a margin loss function to minimize the sum of losses for the training data;

wherein, after a finite number of iterations, the trained capsule neural network model is obtained.

3. The method of claim 2, wherein the step of obtaining one or more input sentence vectors by representing the first multi-intent annotation dataset through vectorization using a pre-trained model comprises a text encoding step, and a model reasoning step.

4. The method of claim 3, wherein the text encoding step comprises:

mapping vocabulary to indexes in the vocabulary,

adding tags or identifiers to the mapped vocabulary, and

generating positional encoding of the tagged and mapped vocabulary to produce encoded text data.

5. The method of claim 3, wherein the model reasoning step comprises:

inputting encoded text data into a pre-trained neural network model to calculate interference of the encoded text data,

generating corresponding vectorized representations of the encoded text data, and

extracting one or more selected vectorized representations of the encoded text data for further processing.

6. The method of claim 2, wherein the step of training the capsule neural network model further comprises:

extracting low-level features of images or text from the input sentence vectors via the one or more convolution layers, and

generating a feature map output.

7. The method of claim 6, wherein the step of training the capsule neural network model further comprises converting the feature map output into one or more vector representations via the one or more primary capsule layers.

8. The method of claim 7, wherein the step of training the capsule neural network model further comprises processing the vector representations via the one or more convolution capsule layers to capture higher level feature relationships within the input sentence vectors.

9. The method of claim 8, wherein the step of training the capsule neural network model further comprises outputting one or more predicted vectors for each category of a set of categories corresponding to the one or more sentence vectors via the one or more fully connected capsule layers.

10. The method of claim 1, wherein the step of inputting the first data into the trained capsule neural network model to obtain a first output result comprises:

pre-processing the first data to be identified to convert the first data to be identified into a first sentence vector to be identified,

performing deep feature extraction on the first sentence vector to be identified using the capsule neural network model, and calculating a second output capsule using the trained capsule neural network model parameters,

calculating a probability of each intent label in the second output capsule using a norm of the second output capsule,

obtaining a predetermined judgment threshold and determining whether the probability of each intent label in the second output capsule is greater than the predetermined judgment threshold to obtain a second judgment result, and

obtaining the first predicted user intent set based on the second judgment result.

11. The method of claim 1, wherein the step of determining a current scene intent set based on the first predicted user intent set comprises:

obtaining structured information from a knowledge base, the knowledge base comprising a structured repository of knowledge information from a source,

determining whether each intent in the first predicted user intent set belongs to the current scenario based on the structured information from the knowledge base, and

determining if an intent in the first predicted user intent set belongs to a current scenario, wherein:

if the determination is made that the intent in the first predicted user intent set does not belong to the current scenario, the intent is removed, and

if the determination is made that the intent in the first predicted user intent set does belong to the current scenario, the intent is retained, and

generating a current scenario intent set of intents in the first predicted user intent set that belong to the current scenario.

12. The method of claim 1, wherein the step of identifying a first intent statement and a second intent statement based on the first determination result further comprises carrying out a judgement of dependencies between intents in the current scene intent set, wherein, if the first judgment result indicates that there are dependencies among the intents in the current scenario intent set, the method further comprises processing and storing the first intent statement according to the order of dependencies, where the first intent statement is the response statement for the last intent.

13. The method of claim 1, wherein the step of identifying a first intent statement and a second intent statement based on the first determination result further comprises carrying out a judgement of dependencies between intents in the current scene intent set, wherein, if the first judgment result indicates that there are no dependencies among the intents in the current

scenario intent set, the method further comprises obtaining and storing the second intent statement(s), where the second intent statement(s) are the response statements for each intent.

14. The method of claim 1, wherein the step of merging the first intent statement and the second intent statement to obtain a response statement comprises:

combining the first intent statement and the second intent statement to obtain a third intent statement,

obtaining a reply priority based on structured information in a knowledge base, the knowledge base comprising a structured repository of knowledge information from a source, and

sorting the third intent statement according to the reply priority of each intent to obtain a reply statement.

15. A method for multi-intent recognition of user utterances, the method comprising:

receiving first data to be identified, the first data comprising user utterances;

inputting the first data into the trained capsule neural network model to obtain a first output result, wherein the first output result includes a first predicted user intent set;

determining a current scene intent set based on the first predicted user intent set;

determining whether there are dependencies between the intents in the current scene intent set to obtain a first determination result;

identifying a first intent statement and a second intent statement based on the first determination result; and

merging the first intent statement and the second intent statement to obtain a response statement.

16. The method of claim 15, wherein the step of inputting the first data into the trained capsule neural network model to obtain a first output result comprises:

pre-processing the first data to be identified to convert the first data to be identified into a first sentence vector to be identified,

performing deep feature extraction on the first sentence vector to be identified using the capsule neural network model, and calculating a second output capsule using the trained capsule neural network model parameters,

calculating a probability of each intent label in the second output capsule using a norm of the second output capsule,

obtaining a predetermined judgment threshold and determining whether the probability of each intent label in the second output capsule is greater than the predetermined judgment threshold to obtain a second judgment result, and

obtaining the first predicted user intent set based on the second judgment result.

17. The method of claim 15, wherein the step of determining a current scene intent set based on the first predicted user intent set comprises:

obtaining structured information from a knowledge base, the knowledge base comprising a structured repository of knowledge information from a source,

determining whether each intent in the first predicted user intent set belongs to the current scenario based on the structured information from the knowledge base,

determining if an intent in the first predicted user intent set belongs to a current scenario, wherein:

if the determination is made that the intent in the first predicted user intent set does not belong to the current scenario, the intent is removed, and

if the determination is made that the intent in the first predicted user intent set does belong to the current scenario, the intent is retained, and

generating a current scenario intent set of intents in the first predicted user intent set that belong to the current scenario.

18. The method of claim 15, wherein the step of identifying a first intent statement and a second intent statement based on the first determination result further comprises carrying out a judgement of dependencies between intents in the current scene intent set, wherein, if the first judgment result indicates that there are dependencies among the intents in the current scenario intent set, the method further comprises processing and storing the first intent statement according to the order of dependencies, where the first intent statement is the response statement for the last intent.

19. The method of claim 15, wherein the step of identifying a first intent statement and a second intent statement based on the first determination result further comprises carrying out a judgement of dependencies between intents in the current scene intent set, wherein, if the first judgment result indicates that there are no dependencies among the intents in the current scenario intent set, the method further comprises obtaining and storing the second intent statement(s), where the second intent statement(s) are the response statements for each intent.

20. The method of claim 15, wherein the step of merging the first intent statement and the second intent statement to obtain a response statement comprises:

combining the first intent statement and the second intent statement to obtain a third intent statement,

obtaining a reply priority based on structured information in a knowledge base, the knowledge base comprising a structured repository of knowledge information from a source, and

sorting the third intent statement according to the reply priority of each intent to obtain a reply statement.

21. A system for multi intent recognition of user utterances, the system comprising:

a capsule neural network module configured to extract features from one or more input sentence vectors corresponding to the user utterances, the capsule neural network module comprising:

a convolutional layer configured to extract low-level features of images or text corresponding the one or more input sentence vectors to produce a feature map output,

a primary capsule layer configured to convert the feature map output into one or more vector representations,

a convolutional capsule layer configured to process the vector representation outputs, and

a fully connected capsule layer configured to output predicted vectors for each category of a set of categories corresponding to the one or more sentence vectors, and

a dynamic routing module to learn the mapping relationship of the capsule neural network model from low-level features to high-level features to obtain at least a first output capsule.

22. The system of claim 21, wherein the capsule neural network model is configured to extract features from the input sentence vector(s) by sequentially passing the input sentence vectors through each of the layers of the capsule neural network model.

23. The system of claim 22, wherein the dynamic routing module is configured to simultaneously utilize a dynamic routing mechanism to learn a mapping relationship of the capsule neural network model from low-level features to high-level features to obtain at least the first output capsule.