An Approach to Natural Language Intention Understanding of Civil Aviation Passengers Based on DIET Architecture

Ning He
R&D Center, The Second Research Institute of CAAC, Chengdu, China
hening@caacsri.com

Weijia Ye*
R&D Center, The Second Research Institute of CAAC, Chengdu, China
yeweijia@caacsri.com

Pan Zhu
R&D Center, The Second Research Institute of CAAC, Chengdu, China
zhupan@caacsri.com

ABSTRACT

With the development of science and technology, more and more products appeared in the airport terminal to provide services for the passenger in civil aviation. However, due to the lack of understanding of natural language by scientific and technological equipment, the repeated consultation of the same question often fails to give satisfactory answers to the passengers, which reduces the service efficiency and friendliness. In order to provide better services for passengers in civil aviation, we aim at the common problems of passengers in the airport environment, simulate the possible service demands of passengers, analyze the key feature data in the demands, and combine the joint identification model of DIET to realize the accurate understanding of the intention in the natural language of passengers. In this paper, Python tools are used for making simulation experiments of natural language and intention understanding to verify the feasibility of the implementation method in this paper.

CCS CONCEPTS

• **Computing methodologies**; • **Machine learning**; • **Machine learning algorithms**;

KEYWORDS

Civil aviation, Intention understanding, Natural language, DIET

ACM Reference Format:
Ning He, Weijia Ye*, and Pan Zhu. 2021. An Approach to Natural Language Intention Understanding of Civil Aviation Passengers Based on DIET Architecture. In *The 5th International Conference on Computer Science and Application Engineering (CSAE 2021), October 19–21, 2021, Sanya, China.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3487075.3487101

1 INTRODUCTION

In Civil aviation travel, more and more technology products appeared in the airport terminal to provide services for the passenger, but due to the technological equipments lack of understanding of natural language, which often appearing the repeatly consultation of the same problem , and cannot give passenger satisfaction answer. The science and technology equipment utilization rate is not high, and feeling bad of using.

Intention recognition can be regarded as a classification problem, which can be solved by classification methods, mainly including rule-based learning, machine learning and other traditional classification methods and deep learning classification methods. This paper first sorts out the possible problems related to the demand in civil aviation travel, analyzes the key words in each problem, and calculates the most likely intention of the problem through natural language understanding and comparison of the key words, so as to achieve the purpose of accurately understanding the passenger intention. Therefore, the requirements of efficient, accurate and adaptable implementation of natural language understanding have become the core of the method in this paper. Compared with the traditional natural language understanding method, this paper uses the DIET framework to realize natural language understanding, which solves the problems of the traditional method of sample training with low efficiency and low accuracy[1][2].

2 EXTRACT KEY INFORMATION FROM SERVICE REQUIREMENTS

By sorting out and classifying the common problems of passengers in air travel, the intention category database is constructed, such as ticket purchase intention, inquiry intention, check-in intention, payment intention, shipping intention, asking the way intention, service intention, security inspection intention, ticket refund intention, connecting intention, flight change intention and so on.

We can design a variety of possible service requirement corpus based on the keywords in the intention recognition database to provide basic training data for the next step of natural language learning. In order to have enough corpus samples for algorithm training, the DSL language of Chatito tool is adopted in this paper to achieve sample generation, and its syntax format is shown in Figure 1

%[consign_for_shipment] indicates the intention is consign_for_shipment.

@[package] indicates that this is an entity identity, and the entity is package.

~[action], ~[start], ~[question] are expressed as various short sentences.

*[90%] ~[start]~[action]@ [Package] indicates that the current combined statement represents 90% of the total corpus file.

*[10%] ~[question] indicates that the statement "QUESTION" represents 10% of the corpus file.

CSAE 2021, October 19–21, 2021, Sanya, China

ACM ISBN 978-1-4503-8985-3/21/10…$15.00
https://doi.org/10.1145/3487075.3487101

```
%[consign_for_shipment]('training':'10','testing':'10')
*[90%] ~[start]~[action]@[package]
*[10%] ~[quesion]
@[package]
  trunk
  baggage
  luggage
~[action]
  consign
~[start]
  ask for where to go
  I want
~[quesion]
  checked luggage requirment
  check the computer
  consignment charge
```

Figure 1: DSL Syntax.

Figure 2: Word Segmentation Effect.

The intention, statement, and corresponding entity values can be automatically marked in the corpus generated by the Chatito tool.

3 PREPROCESSING OF NATURAL LANGUAGE RECOGNITION

Before sentence comprehension of natural language is realized, the sentence is preprocessed by word segmentation and word vector coding, and the processed result goes through corresponding mathematical model to realize entity extraction and intention recognition. In this paper, the word segmentation of natural language statements is realized by JIEBA, and the results of word segmentation of the statement are calculated in Python computing environment, as shown in Figure 2

In this paper, BERT is used to realize the word vector coding work of passenger' sentences. BERT (Bidirectional Encoder Representations from Transformers) is a Word2Vec (Word Translated into Word Vector) model, which set new records in accuracy in 11 directions in the field of NLPs (Natural Language Processing). Statement entered by the passenger is regarded as tag sequence, which can be either word or sentence. The tag sequence needs to be converted into the corresponding word vector before being processed[3]. The schematic diagram of the BERT word coding process is shown in Figure 3

The word coding is implemented in the Python environment, such as the word vector coding of the word "query", as shown in Figure 4

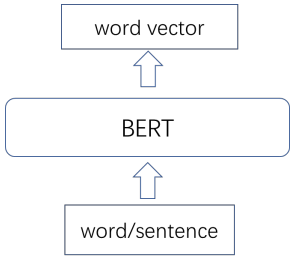

Figure 3: Word Coding Flow Chart of BERT.

```
(python36) [root@htsp ~]# python
Python 3.6.10 |Anaconda, Inc.| (default, May  8 2020, 02:54:21)
[GCC 7.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from bert_serving.client import BertClient
>>> bc = BertClient(port=5555, port_out=5556)
>>> bc.encode(["查询"])[0]
array([-1.98270693e-01,  1.21324778e+00, -4.13309455e-01, -4.30119634e-01,
        2.86693096e-01, -1.24297726e+00,  1.27630308e-01, -5.63713729e-01,
        3.88150997e-02,  6.63490713e-01, -9.90367867e-03,  2.20330670e-01,
        1.24660897e+00, -1.99182540e-01,  1.05195439e+00, -2.72162437e-01,
        2.77773768e-01, -4.65492785e-01,  3.15554976e-03,  7.46627450e-01,
       -2.28456452e-01,  2.62229532e-01, -1.60709307e-01,  7.49890208e-02,
        2.50302970e-01,  2.99810141e-01,  1.06604539e-01,  1.50157481e-01,
```

Figure 4: Implement of Word Vector Coding in Python Environment.

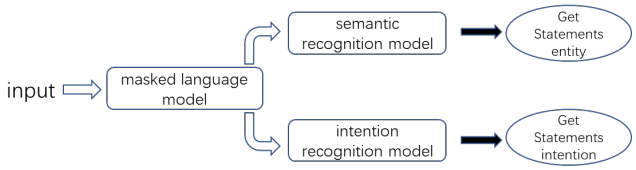

Figure 5: Training Processes of the Three Models for Semantic Understanding.

4 ENTITY EXTRACTION AND INTENTION RECOGNITION OF PASSENGERS' INTENTION

In this paper, Dual Intent and Entity Transformer (DIET) is selected to study the key algorithms of Entity extraction and Intention recognition. The architecture of DIET joint recognition model is composed of three algorithm models, Mask Language Model (MLM), Conditional Random Field (CRF) semantic slot recognition model, and Transformer based intention recognition model, which are independent of each other to realize semantic understanding while users input statements[4]. The training process of the three models for semantic understanding is shown in Figure 5

4.1 Mask Language Model (MLM)

Mask language model adds an additional training target in training to predict the value of the random mask's input token (15% of the token in the random mask input sequence).

Here's an example. If the input statement is "My dog is Hairy", 15% of the tokens will be randomly selected to mask. Suppose the fourth token is chosen and the Hairy is masked, then the masking process is as follows:

```
def _mask_loss(
    self,
    outputs: tf.Tensor,
    inputs: tf.Tensor,
    seq_ids: tf.Tensor,
    lm_mask_bool: tf.Tensor,
    name: Text,
) -> tf.Tensor:
    # make sure there is at least one element in the mask
    lm_mask_bool = tf.cond(
        tf.reduce_any(lm_mask_bool),
        lambda: lm_mask_bool,
        lambda: tf.scatter_nd([[0, 0, 0]], [True], tf.shape(lm_mask_bool)),
    )

    lm_mask_bool = tf.squeeze(lm_mask_bool, -1)
    # pick elements that were masked
    outputs = tf.boolean_mask(outputs, lm_mask_bool)
    inputs = tf.boolean_mask(inputs, lm_mask_bool)
    ids = tf.boolean_mask(seq_ids, lm_mask_bool)

    outputs_embed = self._tf_layers[f"embed.{name}_lm_mask"](outputs)
    inputs_embed = self._tf_layers[f"embed.{name}_golden_token"](inputs)

    return self._tf_layers[f"loss.{name}_mask"](
        outputs_embed, inputs_embed, ids, inputs_embed, ids
    )
```

Figure 6: Calculation of Loss Value in Mask Language Model.

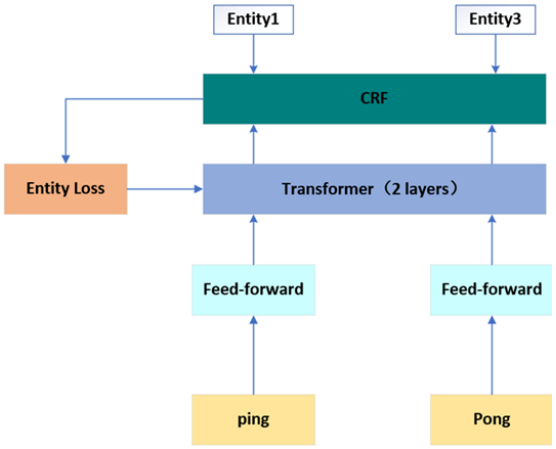

Figure 7: Semantic Slot Filling Recognition Model Training Flow Chart.

1. 70% probability, replace the target token with the vector __MASK__, namely
"my dog is hairy" –> "my dog is __MASK__"
2. 10% probability, replace the target token with the vector __MASK__, namely
"my dog is hairy" –> "my dog is apple"
3. 20% probability, replace the target token with the vector __MASK__, namely
"my dog is hairy" –> "my dog is hairy"

With a mask, Transformer does not know which word it will predict or which word will be replaced by a random word, so it has to maintain the distribution of contextual representations for each input token. That is to say, if the model learns which word is the one to be predicted, and then the learning of contextual information will be lost; and if the model in the process of training can't learn which word is to be predicted, it must learn by contextual information to determine the word to be predicted, and such model has the ability to express the features of sentences. In addition, since the probability of randomly replacing all tokens in the relative sentence is only 1.5% (that is 10% of 15%), it will not affect the language comprehension ability of the model.

The calculation function of the loss value generated in the training of this model is defined as follows:

$$L_M = -\left[S_M^+ - \log\left(e^{s_M^+} + \sum \Omega_M^- e^{s_M^-}\right)\right] \quad (1)$$

All mask terms y_{token} will get a_{MASK} after the processing by the Transformer.

Among them: $S_M^+ = h_{MASK}^T h_{token}^+, S_M^- = h_{MASK}^T h_{token}^-$; while $h_{MASK} = E(a_{MASK})$, h_{MASK} represents the vector space obtained by word embedding a_{MASK}; $h_{token} = E(y_{token})$, Similarly, h_{token} represents the vector space obtained after the embedding of the mask vocabulary, Among them $h \in IR^{20}$; Ω_M^- indicates a group of negative sample.

This paper adopts the Python development environment and adopts the loss function method defined in TensorFlow platform. Figure 6 shows the calculation of loss value in the mask language model.

4.2 Conditional Random Field (CRF) Semantic Slot Recognition Model

Conditional random field (CRF), proposed by Lafferty et al. in 2001, is a probabilistic graph model following Markov property. CRF adds some observed values (features) on the basis of Markov random field, and models the conditional probability distribution of output variables under the given set of input variables. CRF has achieved good results in Chinese word segmentation, named entity recognition, sequence labeling and other problems[5].

Conditional random field (CRF) has achieved good results in the sequential labeling problem. The semantic slot filling task of natural language can be regarded as the sequential labeling problem, that is, the word or words in each input statement is marked with a label related to the slot filling task, and the corresponding slot value is inferred from the label. Here's a simple example of asking for directions at the airport:

Question: Can you tell me the way to the toilet
Tags: O O O B

In the example above, the user wants to ask for directions to the toilet. First, the sentences are segmented based on words, and then the words are marked with BIO marking method. The destination in the question is extracted as slot information, and the extracted destination in this sentence is "toilet".

The training process of the entire semantic slot filling recognition model is shown in Figure 7

The model training adopts English input example, that is, when the phrase ping pong is input, the word segmentation will be split into two tokens, ping and pong, and then the two tokens will be encoded by word vector to obtain word vector, and then the size of Transformer can be adapted after feed-forward processing and then be input to CRF after the processing of Transformer; CRF conducts prediction training according to the entity and the corresponding tag sequence and obtains the loss value of the entity. Then, the loss value is added to Transformer for several times of training, and the model parameters with the best match are obtained after gradient

```python
def _calculate_entity_loss(
    self,
    outputs: tf.Tensor,
    tag_ids: tf.Tensor,
    mask: tf.Tensor,
    sequence_lengths: tf.Tensor,
) -> Tuple[tf.Tensor, tf.Tensor]:

    sequence_lengths = sequence_lengths - 1  # remove cls token
    tag_ids = tf.cast(tag_ids[:, :, 0], tf.int32)
    logits = self._tf_layers["embed.logits"](outputs)

    # should call first to build weights
    pred_ids = self._tf_layers["crf"](logits, sequence_lengths)
    # pytype cannot infer that 'self._tf_layers["crf"]' has the method '.loss'
    # pytype: disable=attribute-error
    loss = self._tf_layers["crf"].loss(logits, tag_ids, sequence_lengths)
    # pytype: enable=attribute-error

    f1 = self._f1_score_from_ids(tag_ids, pred_ids, mask)

    return loss, f1
```

Figure 8: Calculation of Loss Value in Semantic Slot Recognition Model.

descent (if and only if the loss value is the least, the model is the best match).
The calculation function of the loss value generated in the training of this model is defined as follows:

$$L_E = L_{CRF}\left(a, y_{entity}\right) \tag{2}$$

Where a represents the output sequence of the input vocabulary gained through the Transformer, and Y_{entity} is the entity sequence predicted by CRF according to the mark layer in the output sequence a. $L_{CRF}(.)$ represents the loss function of CRF.
This paper adopts the Python development environment and adopts the loss function method defined in the TensorFlow platform. As shown in Figure 8, the loss value in the semantic slot recognition model is calculated.

4.3 Intention Recognition Model Based on Transformer

This model is an intention classifier that identifies the intention of a user statement. First of all, the input content is processed by a feed-forward layer, and following procedures are going through Transformer and word embedding processing. The similarity between the processed results and the intended word embedding output is calculated, and the loss value is obtained. Then, the loss value is added to the Transformer for several times of training. When and only if the loss value is the smallest, the model matching is completed[6].
The calculation function of the loss value generated in the training of this model is defined as follows:

$$L_1 = -\left[S_1^+ - \log\left(e^{s_1^+} + \sum \Omega_1 - e^{s_1^-}\right)\right] \tag{3}$$

The __CLS__ word gets the output a_{CLS} after going through Transformer.
Among them: $S_I^+ = h_{CLS}^T h_{intent}^+$, $S_I^- = h_{CLS}^T h_{intent}^-$; while $h_{CLS} = E(a_{CLS})$, h_{CLS} represents the vector space obtained by a_{CLS} after word embedding;

```python
def _calculate_label_loss(
    self, a: tf.Tensor, b: tf.Tensor, label_ids: tf.Tensor
) -> tf.Tensor:
    all_label_ids, all_labels_embed = self._create_all_labels()

    a_embed = self._tf_layers[f"embed.{TEXT}"](a)
    b_embed = self._tf_layers[f"embed.{LABEL}"](b)

    return self._tf_layers[f"loss.{LABEL}"](
        a_embed, b_embed, label_ids, all_labels_embed, all_label_ids
    )
```

Figure 9: Calculation of Loss Value in Intention Recognition Model.

```python
def _total_batch_loss(
    self, batch_in: Union[Tuple[tf.Tensor], Tuple[np.ndarray]]
) -> tf.Tensor:
    """Calculate total loss"""

    prediction_loss = self.batch_loss(batch_in)
    regularization_loss = tf.math.add_n(self.losses)
    total_loss = prediction_loss + regularization_loss
    self.total_loss.update_state(total_loss)

    return total_loss
```

Figure 10: Calculation of Loss Value in Joint Identification Model.

In the same way,$h_{intent} = E(y_{intent})$, h_{intent} represents the vector space of intention y_{intent} after word embedding, in which $h \in IR^{20}$;Ω_I^- indicates a group of negative sample set.
This paper adopts Python development environment and adopts the loss function method defined in TensorFlow platform. Figure 8 shows the calculation of loss value in the intention recognition model[7].

4.4 Calculation of Loss Function of DIET Joint Identification Model

In the process of semantic understanding realized by DIET joint recognition model, the total loss function is equal to the sum of the three model loss functions, namely:

$$L_{total} = L_1 + L_E + L_M \tag{4}$$

L_I represents the loss function of the intention recognition model, L_E represents the loss function of the semantic slot recognition model, and L_M represents the loss function of the lexical mask model.
This paper adopts Python development environment and adopts the loss function method defined in TensorFlow platform. As shown in Figure 9, it is the calculation of loss value in the joint identification model.

5 EXPERIMENTAL TEST

In the experiment, we simulated a total of 11 intentions including asking for directions, checking, check-in, transfer, refund, etc., and semantic slots including boarding gate, suitcase, toilet, etc., and used DSL language and regular expression to generate test data.

```
"entities": [
    {
        "entity": "package",
        "start": 7,
        "end": 10,
        "extractor": "DIETClassifier",
        "value": "行李箱"
    }
],
"intent_ranking": [
    {
        "name": "consign_for_shipment",
        "confidence": 0.9142504334
    },
    {
        "name": "refund",
        "confidence": 0.0160996318
    },
    {
        "name": "service",
        "confidence": 0.016081458
    },
    {
        "name": "pay",
        "confidence": 0.0126496116
    },
```


Figure 11: Corpus Data Entity Extraction and Intention Recognition Results.

After the calculation of the model algorithm, the results are as follows: the entity extraction and intention recognition results of " Could you tell me where to check the suitcase?" are shown in Figure 10

As can be seen from Figure 10, the semantic slot in the corpus "Could you tell me where to check the suitcase?" can be identified as "suitcase", and the intention identification with the highest credibility is " consign_for_shipment ".

6 CONCLUSIONS

By combining the pre-arranged key words in civil aviation travel with the combined identification model of DIET, the intention of passengers in civil aviation travel can be quickly understood, the real needs of passengers can be obtained, and services can be provided to passengers accurately, which increases the service efficiency and friendliness of passengers.

We believe that this method will gradually enrich the language feature database when it is used in the actual airport environment, leading to higher and higher accuracy in understanding the real intention of natural language in passenger civil aviation travel.

ACKNOWLEDGMENTS

This work is supported by the National Key R&D Program of China: Construction and demonstration on accessible, convenient and intelligent life service system. Under Grant No.2019YFF0303300, Subject 3, Research and application demonstration on airport intelligent accessible service support technology under Grant No.2019YFF0303303.the Sichuan S&T Cooperation Program: Research and technical verification of accessible intelligent interactive system for airport special population under Grant No.2020YFG0110. Sichuan Province Science and Technology Planning Project under Grant No.2019YFH004.

REFERENCES

[1] Zhang Wei (2020).Research on Recommendation Algorithm Combined with Temporal and Spatial Data Features [D]. Shandong University.

[2] Qi Xijing, Tang Liang, Kang Weixin, Qin Jiaojiao (2020). Multi-objective Decision-making Method for Maintenance Scheme of Highway Bridge Beam and Bridge Deck [J]. Journal of Northeastern University (Natural Science Edition),41(07):1033-1040.

[3] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and ChrisDyer (2016). Neural architectures for named entity recognition. CoRR, abs/1603.01360.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). Attention is all you need. In Advances in neural information processing systems, pages 5998–6008.

[5] Yifan Xia and Baosheng Liang (2020). Gaze Estimation Based on Deep Learning Method. In Emrouznejad, A. (eds) Proceedings of the 4th International Conference on Computer Science and Application Engineering (CSAE 2020). Association for Computing Machinery, New York, NY, USA, Article 25, 1–6. https://doi.org/10.1145/3424978.3425003.

[6] Xinghui Wu, Zaifeng Shi, Yuping Zhou, and Haihua Xing (2020). The Application of Three Machine Learning Algorithms in Student Performance Evaluation. In Emrouznejad, A. (eds) Proceedings of the 4th International Conference on Computer Science and Application Engineering (CSAE 2020). Association for Computing Machinery, New York, NY, USA, Article 93, 1-5. https://doi.org/10.1145/3424978.3425072.

[7] Jian Zhu, Yingyu Hou, and Ziqiang Liu (2020). Simulation of a Slender Fish Swimming Based on Machine Learning Method. In Emrouznejad, A. (eds) Proceedings of the 4th International Conference on Computer Science and Application Engineering (CSAE 2020). Association for Computing Machinery, New York, NY, USA, Article 83, 1-4.