IOP Conf. Series: Journal of Physics: Conf. Series **1487** (2020) 012014 doi:10.1088/1742-6596/1487/1/012014

An Intelligent Chatbot System Based on Entity Extraction Using RASA NLU and Neural Network

Anran Jiao

Nankai University, Tianjin, 300071, China

E-mail: anranjiao@mail.nankai.edu.cn

Abstract. Intelligent chatbot systems are popular issues in the application fields of robot system and natural language processing. As the development of natural language processing and neural network algorithms, the application of artificial intelligence is increasing in Chatbot systems, which are typically used in dialog systems for various practical purposes including customer service or information acquisition. This paper designs the functional framework and introduces the principle of RASA NLU for the Chatbot system, then it integrates RASA NLU and neural network (NN) methods and implements the system based on entity extraction after intent recognition. With the experimental comparison and validation, our developed system can realize automatic learning and answering the collected questions about finance. The system analysis of two methods also validate that RASA NLU outperforms NN in accuracy for a single experiment, but NN has better integrity to classify entities from segmented words.

1. Introduction

Natural language processing (NLP) is one of theoretically advanced techniques for the automatic understand human beings and representation of their languages [1]. It is one of the major areas of artificial intelligence, and thus is used in various situations like machine translation, text mining, speech recognition, and so on. There are several basic phases in NLP, including phonetics, morphology, syntax, semantics, and pragmatics. To understand human language, the machine needs to divide the whole chunk of text into paragraphs, sentences, and words. Besides, it should learn to recognize the relationships between the different words, draw the exact meaning from the text, understand sentences in different situations, and consider the prior discourse context [2].

In the early 21st century, a feed-forward neural network language model was proposed. The use of word embedding with word2vec implementation made it efficient to get a certain relation between words. More recently, feed-forward neural networks have been replaced with recurrent neural networks (RNNs) and long short-term memory networks (LSTMs) for language modelling [3]. And current research in NLP is shown that those successfully applied deep learning methods (e.g., convolutional neural networks (CNNs), recurrent neural networks, and recursive neural networks) are replacing traditional hand-crafted models (e.g., SVM and logistic regression) [4]. Collobert et al. [5] argued that a simple deep learning framework performs better in several NLP tasks, especially in named-entity recognition (NER). As RNNs have more “memory” information than other previous computational cells in current processing [4], it is increasingly popular to use an RNN language model for NLP applications in recent years. Conditional Random Fields (CRFs), a sequence labelling, is also influential in NER tasks. With these models, texts are trained to understand the structure and meaning.

Compared with computer work like mathematical operations which is direct and accurate, however, human language is generally ambiguous and concealed in semantic. Thus it is difficult to get computers closer to a human-level understanding of language. As an essential application of NLP, chatbot makes it possible for computers to understand the intent of natural language and make reasonable responses. These applications have recently become increasingly popular on different mobile and online platforms. Among several kinds of chatbots, the main type called "virtual assistants" is served to support requirements from users in widely various domains and sectors. There are three main parts in a chatbot system: a Natural Language Understanding (NLU) part that gets user's intents, a Dialog Management part that monitors the current system and conversation state, and a Natural Language Generation part that responds to the user. NLU part plays a crucial role in the whole system and there are several methods to achieve the goal to understand and respond.

RNNs are shown to be a great way to build a language model, which can also be used for tagging, sentence classification, generating text, and so on. RASA NLU is often used as a tool to build conversational systems, which is an open source natural language understanding module. It comprises loosely coupled modules combining a number of natural language processing and machine learning libraries in a consistent API [6]. Rasa predict a set of slot-labels and slot-values associated with different segments of the input rather than a sequence of slots for each input word [7]. In this work, it is applied in a conversational system compared with popular RNN method in accuracy.

This paper presents a chatbot to search the price, the cap and the volume of stocks based on RASA NLU using iex-finance API. The principle of this method is explained in detail in section 2. Then, another method based on RNN is presented in section 3. In section 4, the comparison of two methods is introduced and accuracy rates are calculated. The conclusion is presented in Section 5.

2. Framework

2.1. Description of Chatbot pipeline

Based on RASA NLU, the entity search module, a chatbot for stock information query could be built. The financial chatbot is connected to WeChat using iex-finance API. Consequently, all the interactions between the bot and users are through WeChat windows. Once users intend to look for the price, the cap or the volume of stocks, messages are sent to the bot, and RASA NLU abstracts entities from the messages. Then the bot gets users' intents and responds properly according to the state machine.

First, since the training examples are less than 1000, we choose a spaCy model for a language called "en_core_web_md" and the RASA NLU pipeline "spacy_sklearn", which uses pre-trained word vectors from GloVe. The two components are written into a file "config_spacy.yml" as machine learning model. Second, there should be a file "rasa_stock.md" to store NLU training data. The data is just a list of messages that users expect to receive, and it has also annotation with the intent and entities that RASA NLU learn to extract. In this case, it includes intents like "affirm", "greet", "goodbye", and "stock_search". Also, the data is structured into several parts: examples, synonyms, and regex features. The examples in "stock_search" are very essential and are expected to be brief and representative, which includes various expressions and even some mistakes. In this work, the following intents are included in the examples of collected dataset:

(1) "I would like to know about the stock";
(2) "show me the stock of [Tesla Inc.](name)";
(3) "I am looking for price of this stock";
(4) "please give me the [turnover](volume) of [Apple Inc.](name)";
(5) "i need the [capitalization](cap)".

Then we use a trainer model to train these data. After receiving a message, RASA NLU returns information on all the entities that have been identified. After obtaining the entities in the message, we select "symbol" from the entities, which refers to names of stocks. Using iex-finance API, it is easy to get relevant information such as price, market cap, and volume.

The main function of the chatbot is to provide answers to several rounds of inquiries about stocks. For example, users can ask for the bot's function, and it will answer, "I'm a bot to help you with stocks". If users want to know something about stocks, the bot will provide information about several stocks step by step as shown in figure 1.

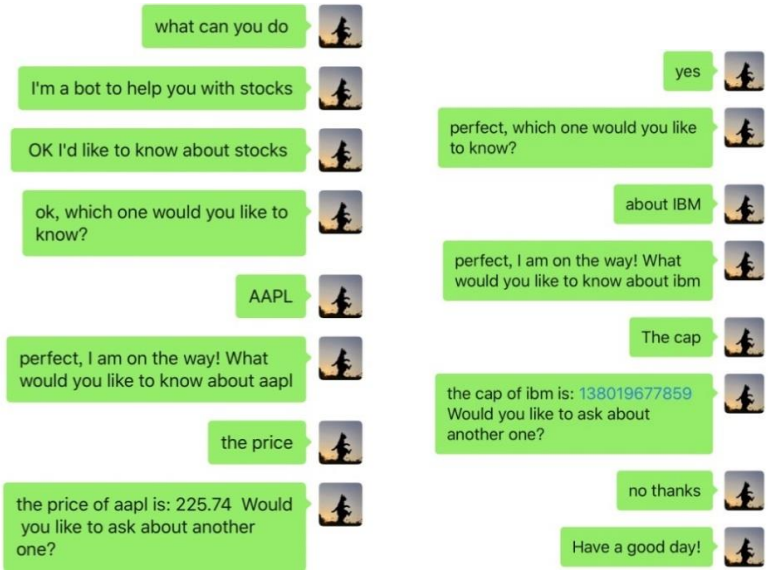

Figure 1. The example conversation of financial chatbot.

The general steps are summarized as following:

(1)Send a message to the financial chatbot;

(2)RASA NLU analyses the sentence and returns entities;

(3)Get information about stocks from iex-finance API;

(4)Return the possible intent of the message chatbot received by regular expression and keywords;

(5)Respond to the message according to the intents and current states based on the state machine.

Overall, step 2 "RASA NLU analyses the sentence and return entities" is obviously the vital part of NLP in this financial Chatbot, so we will describe its principle in the next section in detail.

2.2. Principle of RASA NLU

As presented before, RASA NLU is used to understand language for chatbots and AI assistants, especially focus on intent classification and entity extraction. Pipeline "spacy_sklearn" is composed of different components using some NLP libraries such as spaCy, scikit-learn, and sklearn-crfsuite. This pipeline is equal to the following full list of components in figure 2 referring to the documentation at https://nlu.rasa.ai.

```
pipeline:
- name: "nlp_spacy"
- name: "tokenizer_spacy"
- name: "intent_entity_featurizer_regex"
- name: "intent_featurizer_spacy"
- name: "ner_crf"
- name: "ner_synonyms"
- name: "intent_classifier_sklearn"
```

Figure 2. The list of components which are equal to pipeline "spacy_sklearn".

With these components, RASA NLU is able to analyse messages. The three basic principles in the process are described as follows.

First, the "tokenizer_spacy" component promotes a tokenization process using spaCy, a library for advanced natural language processing written in Python and Cython [8]. And parts of speech (POS) are annotated using the library. In spaCy, strings, word vectors and lexical attributes are stored in containers. In order to grant efficiency compared with the standard BiLSTM solution, special neural models are designed and developed from scratch specifically for spaCy in which convolutional layers are deployed with residual connections, batch normalization and maxout non-linearity. From the work of Joakim Nivre [9] about the transition-based framework, the arc-eager transition system, and the imitation learning objective, the foundation of the parser is built.

Then, the GloVe vectors [10] extracted from each token are concatenated to form the feature vector of a whole sentence. This representation of a sentence is used to train a multiclass support vector machine (SVM) and recognize user intents by "intent_classifier_sklearn" component.

Finally, "ner_crf" component provides a conditional random field (CRF) classifier to train on the sentence tokens and POS tags to extract entities in the training data. CRFs use words as the time steps and entity classes as the states if it is considered to be an undirected Markov chain. By recognized features (capitalization, POS tagging, etc.) of the words, the most likely set of tags is then calculated and returned. The intent classifier and entity extractor are trained using the scikit-learn library and the sklearn-crfsuite library, respectively [11].

3. Methods for Entity Extraction

3.1. RASA NLU method

In order to predict whether a given word, in context, represents one of five categories: name (stock company name), symbol (stock symbol), price (stock price), cap (stock capitalization), and volume (stock market value), we could treat it as a 6-class classification problem.

Using the pipeline of RASA NLU, we prepared a training data set to recognize the intent and extract entity. The training data includes several intents: affirm, greet, goodbye, and stock_search. In stock search part, we use about 500 sentences with marked entities to train the RASA NLU trainer. In the training data set, there are different kinds of stocks, expressions, and sentence patterns. And in a single sentence, there may be none entity or several entities. Since the chatbot is mainly used to search stocks in daily life, sentences in the training data contain stock related words and colloquial expressions, including both interrogative sentences and declarative sentences.

More specifically, there are 464 sentences in stock searching part. 49 sentences (10.6%) without named entity are presented to search for “stocks” or “shares”, which includes 27 distinct sentence patterns. 31 sentences contain named entity “symbol”, among which there are 24 sentences with 11 kinds of stock symbols like AAPL. In the rest 7 sentences, “ticker symbol”, “symbol”, and “stock symbol” all refer to named entity “symbol”. 35 sentences contain named entity “name”, including 11 different kinds of companies. Similarly, 7 sentences use “company” and “company name” to represent named entity “name”. 32 sentences with “price”, “stock price”, and number like “19.82” refer to named entity “price”. 33 sentences with “capitalization”, “cap”, and “market cap” refer to named entity “cap”. Also, 33 sentences with “volume”, “turnover”, “daily volume” and “average volume” refer to named entity “volume”. Totally, 163 (35.1%) in 464 sentences contain only one entity, and following 214 sentences (46.1%) contain two entities. 36 sentences contain both “symbol” and “price” named entities. 35 sentences contain both “symbol” and “cap” named entities. And 37 sentences have both “symbol” and “volume” entities. 35 sentences have both “name” and “price” entities. 35 sentences have both “name” and “cap” entities. 36 sentences have both “name” and “volume” entities. The other 38 sentences (8.2%) have three entities.

With our designed pipeline and collected training data, it is easy to get predicted intents and entities. We import trainer from rasa_nlu package to train the data and then parse the interpreter. A list of entity information will be returned, including value and entity dictionary. To judge the accuracy, we

have a test data set with 160 sentences in it. Intentionally, there are some spelling mistakes, such as volum, capitalizaton, vilue, and so on. Compare the predicted intents and entities with correct ones, we could get the accuracy. The intent accuracy is defined as correct recognized intents divided by the total number of intents.

$$\text{intent\_accuracy} = \frac{\text{number of correct recognized intents}}{\text{total number of intents}} \tag{1}$$

The entity extract accuracy is defined as a number of correct extracted entities divided by the total number of entities. Besides, the integrity of entity means a number of recognized entities divided by the total number of entities, while the integrity of sentence means the total number of lines divided by the number of a sentence which entities are extracted completely.

$$\text{entity\_accuracy} = \frac{\text{number of correct extracted entities}}{\text{total number of entities}} \tag{2}$$

$$\text{integrity of entity} = \frac{\text{number of recognized entities}}{\text{total number of entities}} \tag{3}$$

$$\text{integrity of sentence} = \frac{\text{number of completely extracted lines}}{\text{total number of lines}} \tag{4}$$

3.2. Neural Network method

The NER model can be a one-hidden layer neural network with an additional representation layer similar to word2vec, using Tensorflow system. TensorFlow is a machine learning system that operates at large scale and in heterogeneous environments, which uses dataflow graphs to represent computation, shared state, and the operations that mutate that state [12]. First, there are prepared word vectors, training set, dev set for tuning hyperparameters, and the test set. Words are converted to one-hot vectors $\boldsymbol{x}^{(t)}$ into an embedding matrix L, and labels of named entities are also converted into digits. Second, we choose the tanh activation function for the hidden layer and use the neural network to train the weight W and the bias b. We could get the prediction $\hat{\boldsymbol{y}}$ as:

$$\boldsymbol{h} = \tanh\left(\boldsymbol{x}^{(t)}\boldsymbol{W} + \boldsymbol{b_1}\right) \tag{5}$$

$$softmax\left(x\right) = \frac{e^x}{\sum e^x} \tag{6}$$

$$\hat{\boldsymbol{y}} = softmax\left(\boldsymbol{hU} + \boldsymbol{b_2}\right) \tag{7}$$

In equation (5)-(7), $\boldsymbol{U}$ denotes the input of softmax layer, $\boldsymbol{h}$ denotes the output of tanh layer, $\boldsymbol{b_1}$ and $\boldsymbol{b_2}$ are the biases, respectively. Then we evaluate the cross-entropy loss as:

$$\boldsymbol{J_1} = -\sum_{i=1}^{6} y_i log(\hat{y}_i) \tag{8}$$

In equation (8), the true value y_i and the prediction value $\hat{y}_i$ of six kinds of samples are used to calculate the total loss value $\boldsymbol{J_1}$. To avoid parameters from overfitting, we also use the extra L2 regularization term of loss function $\boldsymbol{J_2}$, and get the combined loss function $\boldsymbol{J}$ as:

$$\boldsymbol{J} = \boldsymbol{J_1} + \boldsymbol{J_2} \tag{9}$$

$$J_2 = \frac{\lambda}{2}\left[\sum_{i,j} W_{ij}^2 + \sum_{i',j'} U_{i'j'}^2\right] \tag{10}$$

During the training process, we use the gradient descent algorithm to find the minimum of the loss function. We could also find the confusion matrix for the dev set. Finally, since we get the prediction of test set, we have the result also include entity extract accuracy and integrity.

$$\text{entity\_accuracy} = \frac{\text{number of correct extracted entities}}{\text{total number of words}} \tag{11}$$

$$\text{integrity of entity} = \frac{\text{1-number of words with O label in prediction}}{\text{total number of words with named entity labels}} \tag{12}$$

$$\text{integrity of sentence} = \frac{\text{number of completely extracted sentences}}{\text{total number of sentences}} \tag{13}$$

The sum of the training set, dev set, and test set are similar to the data set we used in section 3.1. The only difference is that sentences are divided into words with labels. In total, there are 3252 words for training and 1064 words for testing in our collected dataset.

4. System analysis

4.1. Analysis of intent recognition

As for RASA NLU method, the result in figure 3 shows that intent accuracy is 0.99375. Basing on the training set, there is only one mistake that intent of "how are you" are recognized as "stock_search", which is shown in table 1. Since there is no similar expression in training data, it is plausible that there is a mistake. As for "have a good time", "time" is a new word for the computer which replaces the word "one" and "day", and there is no mistake in this experiment.

```
intent_accuracy: 0.99375
--USER--: how are you
intent error!
pre_intent: stock_search intent: greet
```

Figure 3. The result of intent recognition using RASA NLU method.

Table 1. Confusion matrix of intent recognition using RASA NLU method.

	Affirm	Greet	Goodbye	Stock_search	Total
Affirm	3	0	0	0	3
Greet	0	2	0	1	3
Goodbye	0	0	4	0	4
Stock_search	0	0	0	150	150
Total	3	2	3	150	160

4.2. Analysis of entity extraction

First, we evaluate the RASA NLU method. When it comes to entities, the integrity is 0.92 and the entity extract accuracy is 1.0 with the particular training set.

As presented in figure 4, if there are some spelling mistakes like "turnove" or incomplete expressions like "Apple" (correct expression should be "Apple Inc.") in the sentence, errors may occur. Besides, the result shows that sometimes the dignity cannot be recognized as "price" and company name "Tesla Inc." or "Facebook Inc." placed at the end of the sentence cannot be extracted.

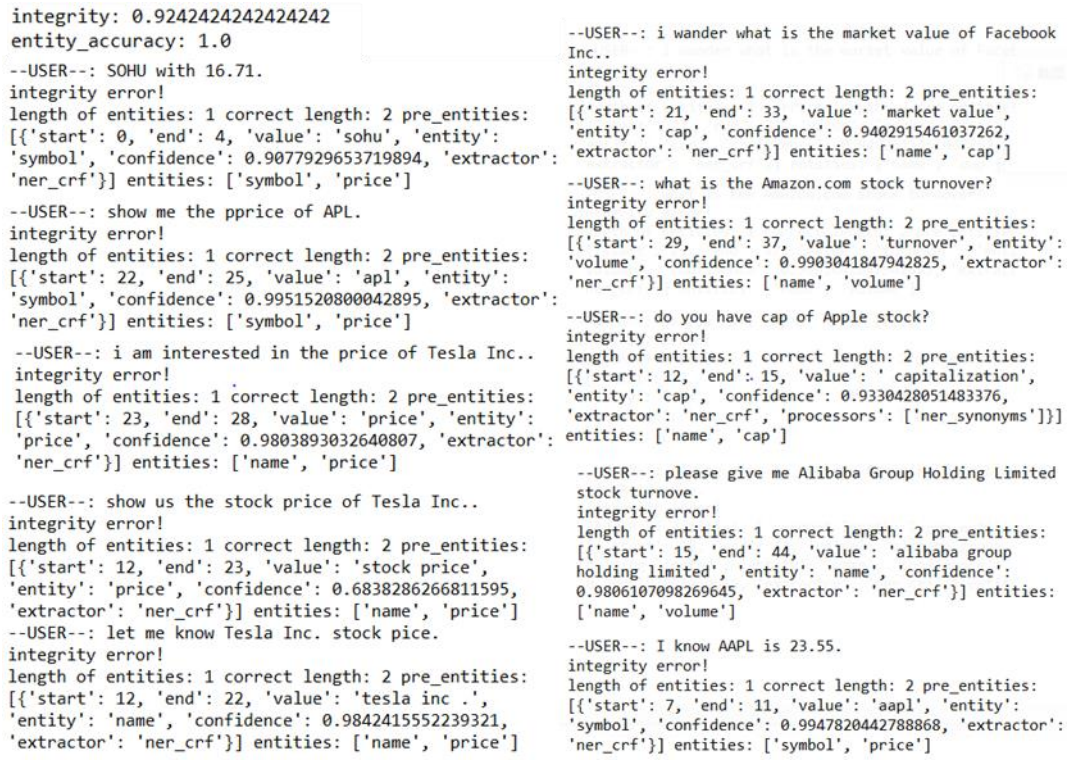

Figure 4. The result of entity extraction using RASA NLU method.

During the process using particular training data and test data, there is no mistake in entity extraction, which shows that it is of great accuracy to extract named entity for RASA NLU. 10 entities in 209 entities are missing as shown in table 2.

Table 2. Confusion matrix of entity extraction using RASA NLU method.

	Symbol	Name	Price	Cap	Volume	None	Total
Symbol	51	0	0	0	0	0	51
Name	0	43	0	0	0	5	48
Price	0	0	36	0	0	4	40
Cap	0	0	0	36	0	0	36
Volume	0	0	0	0	33	1	34
None	0	0	0	0	0	0	0
Total	51	43	36	36	33	10	209

As for the neural network method, accuracy is increased step by step in the training process. In a single test, the entity extract accuracy is 0.95 and the integrity of entity is 0.99 as shown in figure 5.

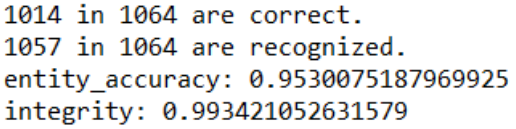

Figure 5. The result of entity extraction using the NN method.

From the confusion matrix in table 3, we could see that several mistakes are made when deciding the entities. 7 entities are missing in the process.

Table 3. Confusion matrix of entity extraction using the NN method.

	Symbol	Name	Price	Cap	Volume	None	Total
Symbol	34	10	0	0	3	5	52
Name	7	41	0	1	0	1	50
Price	1	2	36	0	1	1	41
Cap	1	0	0	23	13	0	37
Volume	1	1	0	1	32	0	35
None	0	0	1	0	0	848	849
Total	44	54	37	25	49	855	1064

4.3. Comparison of two methods in entity extraction

In a single experiment, RASA NLU method has higher accuracy compared with NN method. Also, when considering one sentence as a whole, the RASA NLU method is superior to extract all the entities. However, the NN method has better integrity to classify entities from segmented words as presented in table 4.

Table 4. Comparison of RASA NLU method and NN method in entity extraction (round the number to two significant figures).

	accuracy	integrity of entity	integrity of a sentence
RASA NLU method	1.0	0.95	0.92
RNN method	0.95	0.99	0.70

5. Conclusion

Natural language processing is a vital component of intelligent Chatbot systems. In this paper, a function framework is designed and the principle of RASA NLU is introduced for the Chatbot system. The designed system integrates RASA NLU and neural network (NN) methods and implements the system based on entity extraction after intent recognition. This paper has compared our methods in recognition accuracy and integrities of entity or sentence, and has also validated the developed system in realistic situation. In the future, this system will be further improved the recognition accuracy of entity extraction for longer sentences and more complicated entities. The methods in this paper are only used for academic research and not for commercial purposes.

References

[1] Cambria E, White B 2014 Jumping NLP curves: a review of natural language processing research *J. IEEE Computational Intelligence Magazine* **9(2)** p 48-57.
[2] Tembhekar S and Kanojiya M 2017 A Survey Paper on Approaches of Natural Language Processing (NLP) *J. International Journal of Advance Research, Ideas and Innovations in Technology* **3(3)** p 1496-98.
[3] Kamper H and Ruder 2018, October 1 A Review of the Neural History of Natural Language Processing *Aylien* Retrieved March 18, 2019, from recent-history-of-natural-language-processing/
[4] Young T, Hazarika D, Poria S and Cambria E 2017 Recent Trends in Deep Learning Based Natural Language Processing *arXiv: Computation and Language*
[5] Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K and Kuksa P P 2011 Natural Language Proc. (Almost) from Scratch. *J. Journal of Machine Learning Research* p 2493-2537.
[6] Bocklisch T, Faulkner J, Pawlowski N and Nichol A 2017 Rasa: Open Source Language Understanding and Dialogue Management *arXiv: Computation and Language*

[7] Desot T, Raimondo S, Mishakova A, Portet F and Vacher M 2018, September Towards a French Smart-Home Voice Command Corpus: Design and NLU Experiments *C. text speech and dialog*
[8] Honnibal M and Montani I 2017 spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing *To appear*
[9] Goldberg Y and Nivre J 2012 A Dynamic Oracle for Arc-Eager Dependency Parsing *C. international conf. on computational linguistics*
[10] Pennington J, Socher R and Manning C D 2014 Glove: Global Vectors for Word Representation *C. empirical methods in natural language proc.*
[11] Segura C, Palau A, Luque J, Costa-jussa M and Banchs R 2018 Chatbol, a chatbot for the Spanish "La Liga"
[12] Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, ... and Zheng X 2016 TensorFlow: a system for large-scale machine learning *J. operating systems design and implementation* p265-283