Dialogue Act Detection Using Contextual Knowledge

Jan PICHL

Dept. of Cybernetics, Czech Technical University, Technická 2, 166 27 Praha, Czech Republic
pichljan@fel.cvut.cz

Abstract. Dialogue act detection is an important part of the conversational systems. It describes a user’s intention which is one of the features the system uses to select the next action. In the comparison to the intent detection, dialogue acts are more general, and they do not depend on the current topic. We compare several model architectures which use a single utterance as an input, multiple concatenated utterances or utterance window in which the single utterances are processed sequentially. The experiments are conducted on a Switchboard dataset which consists of transcribed telephone conversations. The resulted system is used as part of the Alquist bot which is one of the competitors of the Alexa Prize 2018.

Keywords
Dialogue system, conversational AI, dialogue acts, intent detection

1. Introduction
Thanks to the growing number of smart assistants like Amazon Echo or Google Home in the market, the popularity of conversational systems grows as well. These conversational systems also called bots are commonly used for various types of tasks such as hotel reservation, ordering pizza or a simple chat. All the types of dialogues share a lot of common aspects. The part of the dialogue system which is shared among various use-cases is the natural language understanding (NLU). NLU combines several tasks such as intent recognition, dialogue act detection, entity recognition, entity linking, or entity disambiguation. The results of this task are further used to manage the dialogue flow and to select appropriate responses.

The task-oriented systems[1] have typically more restricted domain in comparison to chit-chat systems. This restriction allows the developers to take advantage of intent recognition since all the possible intents can be enumerated and we can build a classifier with a fixed number of classes. The open domain systems or chit-chat focused dialogues, on the other hand, can have an enormous number of intents where each intent corresponds to the particular topic or a specific part of the topic.

Since the intents cannot be enumerated, it is usually more suitable to use a type of a mentioned entity to determine the topic or a particular sub-dialogue to be triggered. In addition to the entity type, it is necessary to know the type of the utterance. We need to classify whether the sentence is a statement, question, confirmation, etc. which is usually called a dialogue act [2]. Based on the combination of these two features (entity type and dialogue act), the dialogue system is able to determine which part of the system should be executed.

The additional decision is made by component named dialogue manager (DM) which can be a machine learning model taking NLU features as an input or simple rules made on top of the same features.

2. Related Work
Since the dialogue acts detection task is a specific case of text classification, various approaches from other related tasks can be used. One of the popular models is a convolutional neural network[3]. This architecture originally applied to image related tasks has shown successful results on text classification [4]. Recurrent neural networks[5, 6] designed to deal with series of data are used [7] for text classification to address variability in length of sentences.

In the task-oriented dialogues systems[1], all the intents depend on the domain and the restrictions of the system. The intents are a more specific case of dialogue act and can be detected using the same methods[8, 9, 10].

For all the models, a sequence of sentence words needs to passes as an input. The most straightforward approach is to pass the word as a one-hot encoded index to a previously created vocabulary. This approach results in a large sparse matrix. To reduce the dimension of the vectors in natural language processing (NLP) tasks, the word vectors such as word2vec[11] or GloVe[12] are used.

In this work we apply current text classification models on dialogue act classification task[13, 14, 15] with the extension of incorporating contextual data. The models are tested on commonly used Switchboard dataset[16].
3. Task description

Dialogue acts detection is a text classification task. Each sentence is represented as a sequence of words, and it needs to be classified into one of the predefined classes. The basic classes describe whether the user’s utterance in a statement, question or confirmation. Additional classes are often included based on the needs of the dialogue systems. We describe the classes in detail in Experiments section.

3.1. Dataset description

We use Switchboard dataset for our experiments. It contains two-sided telephone conversations which are transcribed into the text. Most importantly, each sentence is labeled with one dialogue act class. The original dataset contains over 200 classes. The Switchboard Coder’s Manual, however, contains rules to cluster those classes into a total number of 43. The corresponding names and examples for each dialogue act class are shown in the Table 1.

<table>
<thead>
<tr>
<th>Name</th>
<th>Tag</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statement-non-opinion</td>
<td>sd</td>
<td>Me, I’m in the legal department.</td>
</tr>
<tr>
<td>Acknowledge (Backchannel)</td>
<td>b</td>
<td>Uh-huh.</td>
</tr>
<tr>
<td>Statement-opinion</td>
<td>sv</td>
<td>I think it’s great</td>
</tr>
<tr>
<td>Abandoned or Turn-Exit</td>
<td>%</td>
<td>So, -</td>
</tr>
<tr>
<td>Agree/Accept</td>
<td>aa</td>
<td>That’s exactly it.</td>
</tr>
<tr>
<td>Appreciation</td>
<td>ba</td>
<td>I can imagine.</td>
</tr>
<tr>
<td>Yes-No-Question</td>
<td>qy</td>
<td>Do you have to have any special training?</td>
</tr>
<tr>
<td>Non-verbal</td>
<td>x</td>
<td>[Laughter], [Throat_clearing]</td>
</tr>
<tr>
<td>Yes answers</td>
<td>ny</td>
<td>Yes.</td>
</tr>
<tr>
<td>Conventional-closing</td>
<td>fc</td>
<td>Well, it’s been nice talking to you.</td>
</tr>
</tbody>
</table>

The distribution of the top 10 classes is shown in the Figure 1. Most of the sentences from the dataset are statements followed by acknowledgments and opinions. Some of the classes such as question are divided into several subclasses (Yes-No-Question, Wh-Question, Declarative Yes-No-Question, Open-Question, Rhetorical-Questions). This fact should be considered before building a dialogue system since multiple similar classes can reduce the overall performance and they could be unnecessary for the particular system.

Fig. 1. Counts of the 10 most frequent classes in Switchboard dataset (after clustering).

Overall, the dataset contains 213,543 training samples and 4,514 testing samples. The average length of the sentence is 8 tokens.

4. Model architectures

We propose several model architectures for solving the task. Most of them are inspired by other text classification tasks. The models can be divided into two basic groups depending on whether they use contextual information or not. The contextual information includes an utterance window which takes \( N \) sentences as the input. Another approach to include the context is to use the model which accepts only one utterance as the input and form the sentence in a way to contain multiple concatenated utterances. The idea behind the contextual window is that some of the dialogue act classes difficult to be predicted based on the sentence words only.

All the models use pre-trained word vectors as input. We use GloVe with 300 dimensions trained on 6 billion tokens. The coefficients of the word vectors are not tuned during the network training process.

4.1. Multi-channel convolution

This model is inspired by [4]. First, each sentence has to be padded to obtain sentences with the same length. The model uses five convolutional channels of filters with lengths from one to five. This allows the model to catch the semantic meaning of groups of adjacent words. The output dimensionality of the filters of each length is set to \( N/2 \) where \( N \) is the dimensionality of word vector. The activation function is \( \tanh \). Each convolutional layer is followed by max pooling layer with window length equal to the output length of the previous layer which means that only the one top feature for each filter length is selected. Now we have five vec-
tors (one for each filter length) which we concatenate into a single vector. This vector intuitively introduces a sentence representation. The concatenated vector is then fed into a fully connected layer with softmax activation and the output dimension equal to the number of all classes.

4.2. Recurrent neural network

The recurrent neural network model eliminates the requirement to have each sentence of the same length. Each word vector from input sentence is sequentially fed into the LSTM\([6]\) layer. We use a bidirectional recurrent layer which means that the words are also fed into the recurrent layer in a reversed direction. The outputs from the forward and backward processing are then concatenated. The concatenated vector is then fed into the same fully connected layer as in the convolutional model which classifies the input into the desired number of classes.

4.3. Contextual CNN-RNN

The previous models take one utterance as an input. Since we are dealing with dialogues, the sentences can be grouped into the sets corresponding to the individual dialogues. The data need to be formed as follows. Each sample consists of \( C \) utterances where \( C \) is the length of the contextual window. This means that each sample has one additional dimension (the time series dimension). The convolutional part of the network is the same as described in Section 4.1. This part of the network is then applied sequentially over the time series dimension for each utterance separately. Intuitively, it creates a sentence representation for all the utterances in the context window. This set of sentence representations is then sequentially processed by the LSTM layer. This layer processes the sentences only in the forward direction. In comparison with the recurrent layer in the previous subsection, we do not use bidirectional layer since the dialogue act does not depend on the future context because it is unknown at the time of classification.

The sequentially processed utterances are then fed into the fully connected layer which classifies the sample into the desired number of classes.

5. Experiments

5.1. Data preprocessing

The Switchboard dataset [16] contains the annotated dialogues with dialogue acts. Additionally, it contains the part-of-speech tag (POS) for each word and disfluency markers [17]. The disfluency markers are the specials symbols which highlight repeats, self-repairs or filler words. Since we want to classify the dialogue acts purely based on the sentence words, we need to remove those marks from the text.

We use the python toolkit available on Github\(^2\) to process the dataset. This allows us to iterate over the utterances spread among a large number of files. Additionally, it allows us to select only the POS tagged words and remove the disfluency tags automatically. However, approximately one-third of the testing dataset does not contain POS tagged words thus this approach returns empty sentences for these samples.

There are two approaches how to clean the data. The first is to simply remove only the disfluency tags and keep all the tagged word in the sentence. The second is to remove all the tags along with all the correction or repeating words. This approach results in clearer sentences with no repetitive words. We choose the former approach as it is more similar to real-world dialogues since users are correcting themselves and repeating words as they speak.

5.2. Results

We tested all the models proposed in Section 4 and compared them with state-of-the-art attention RNN described in [13] (the accuracy of the model is marked by ‘*’ in Table 3).

The parameters of our models were set as follows. The common parameters for all the models are: size of the dictionary: 10,000, max sentence length: 40, embedding dimension: 300, training/validating split ratio: 0.7/0.3, loss: categorical cross-entropy, optimizer: adam, number of epochs: 8 and batch size: 32. The model-specific parameters are listed below:

- **CNN**: 5 different kernel sizes (from 1 to 5), output dimension for each kernel: 150, activation: tanh, max-pooling windows size: \( 40 - ks + 1 \) where \( ks \) is a kernel size.
- **RNN**: bidirectional, number of units: 300
- **CNN-RNN**: the same CNN layer as described above followed wrapped by time distributed wrapper, followed by LSTM with 500 units.

The accuracy of each model compared with the current state-of-the-art model is shown in Table 3. The accuracy is measured on the original Switchboard test split (4,514 samples). It is important to note that we were not able to reproduce the accuracy of the state-of-the-art model with the information provided in their paper. Since they do not provide information about dataset preprocessing, we have to apply a custom solution which might be different. We implemented

\(^2\)https://github.com/cgpotts/swda
Tab. 2. Selected misclassified sentences.

<table>
<thead>
<tr>
<th>Utterance</th>
<th>Predicted act</th>
<th>Annotated act</th>
</tr>
</thead>
<tbody>
<tr>
<td>I don’t think so either.</td>
<td>Statement-opinion</td>
<td>Agree</td>
</tr>
<tr>
<td>And, uh, oh, what’s that real, oh, not real old, my wife she is old too.</td>
<td>Wh-Question</td>
<td>Statement-non-opinion</td>
</tr>
<tr>
<td>And there’s, there’s no waiting period on that.</td>
<td>Declarative Y/N-Question</td>
<td>Statement-non-opinion</td>
</tr>
<tr>
<td>That’s really great.</td>
<td>Statement-opinion</td>
<td>Appreciation</td>
</tr>
<tr>
<td>I, I agree with you.</td>
<td>Statement-non-opinion</td>
<td>Agree</td>
</tr>
<tr>
<td>You mean like from the coal.</td>
<td>Summarize</td>
<td>Statement-opinion</td>
</tr>
<tr>
<td>And so, the best bet, any, anymo-, these days are compact disk</td>
<td>Statement-opinion</td>
<td>Statement-non-opinion</td>
</tr>
<tr>
<td>But, uh, that gets old too in a very short order</td>
<td>Statement-non-opinion</td>
<td>Statement-opinion</td>
</tr>
<tr>
<td>Right.</td>
<td>Agree</td>
<td>Acknowledge</td>
</tr>
<tr>
<td>What’s that going to be like.</td>
<td>Rhetorical-Questions</td>
<td>Wh-Question</td>
</tr>
</tbody>
</table>

proposed attention model with smoothing and contextual information which resulted in 66.4 % accuracy on the preprocessed dataset.

5.3. Error analysis

In Table 3, we can see that our models have lower accuracy than the presented state-of-the-art attention model described in [13]. However, it is comparable to the accuracy of the SOTA model which we reproduced given the information in their paper. We can also see that adding a contextual information slightly improves the accuracy.

However, our primary focus is to integrate the dialogue act classifier into our open-domain chatbot. We randomly selected 100 of utterances from the validation split to see the errors made by the model. In the selected samples, the distribution of the errors is as follows:

- 33 % - The predicted class is Statement-non-opinion while the ground truth is Statement-opinion or vice-versa.

- 30 % - Actual misclassifications including No class instead of Yes, Statement instead of a question, Statements instead of Affirmative non-yes answers and Statement instead of Appreciation

- 19 % - Wrong type of yes answer. There are similar classes in the dataset: Acknowledge, Agree, Yes answers and Response Acknowledgement

- 9 % - Predicted statement (or another class) for uninterpretable utterance

- 5 % - Questionable annotation. These utterances are shown in the first 5 rows of Table 2.

- 4 % - Wrong type of question: Rhetorical-Questions instead of Wh-question

6. Conclusion

We evaluated the performance of several models on dialogue acts prediction task. Although our performance of the reconstructed state-of-the-art model is lower than the accuracy presented in their paper, we believe this is mainly caused by insufficient information about dataset preprocessing. We compared the results of CNN, RNN and combined models with and without contextual information. The contextual information causes a small increase in accuracy since some dialogue acts also depend on the previous utterances.

The information described in Error analysis section is crucial for the final usage in the dialogue system. There are a lot of classes which are very similar to each other such as various types of questions, or agreements. Typically, dialogue system does not need several classes for yes or agreements to make a decision about the next step. Merging those classes into one increases the accuracy of the model while the dialogue system maintains its ability to drive the dialogue.

Acknowledgements

The research described in the paper was supervised by Ing. Jan Šedivý, CSc., CIIRC CTU in Prague and it was
supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS18/103/OHK3/1T/37. Furthermore, it was supported by Amazon company by stipends and provided infrastructure.

References


About Authors...

Jan Pichl finished his bachelor’s degree in 2014 in Software Engineering at Faculty of Informatics, Czech Technical University. The topic of the bachelor thesis was Optical music sheet recognition module. After that, he started his master’s degree in Artificial Intelligence and finished in 2016 at Faculty of Electrical Engineering, CTU. The topic of the master thesis was Enhancing Question Answering with Structured Database Queries. Now, he is a Ph.D. student, and he is interested in Natural Language Processing. He also works as a researcher in eClub Prague on related projects. Last year, he was a team leader of Alquist team in Alexa Prize competition.