Intents Classification for Neural Text Generation

Thomas FAVOLI
CentraleSupélec
thomas.favoli@student-cs.fr

Abstract

This article explores various text classification methods on the HCRC Map Task Corpus, including Bag-of-Words model with Naïve Bayes, Glove with Bidirectional LSTM, Glove with TextCNN, and BERT with MLP. The article highlights the strengths and limitations of each method and provides insights for future research in intent classification. GitHub Code

1 Introduction

The concept of dialogue act revolves around the idea that every utterance in a conversation has a purpose or intention. It is not just a sequence of words or sentences, but rather a deliberate attempt to convey a certain meaning or message. To better understand human conversations, it is important to be able to identify these underlying intentions or functions of the utterances. This is where the task of dialogue act classification comes into play. The classification of dialogue acts has numerous implications for natural language processing. It can be used to develop more advanced and intelligent dialogue systems that can facilitate various applications such as information retrieval, sentiment analysis, opinion mining, question answering, and automated summarization. The ability to identify different types of dialogue acts can also help better understand human emotions and social interactions.

The classification of dialogue acts can be approached using various techniques, including statistical, machine learning-based, and deep learning techniques. These techniques involve the use of various features such as lexical, syntactic, and semantic features, as well as contextual features such as the speaker, topic, and discourse structure.

In recent years, there has been a growing interest in deep learning techniques such as recurrent neural networks (RNNs) and convolutional neural networks (CNNs) for dialogue act classification. These techniques have shown promising results, and are expected to further improve the accuracy and efficiency of dialogue act classification (Duran et al., 2021).

2 Experiments Protocol

We will be using the Bag-of-words technique along with Naïve Bayes algorithm to classify dialogue acts. This approach involves extracting the features of the utterances and treating them as independent entities, which are then used to predict the class labels of the dialogue acts. The accuracy of this technique is expected to be moderate and may require further improvements.

Following this, we will be using the GloVe (Global Vectors for Word Representation) algorithm along with Bidirectional RNN (LSTM) and CNN (textCNN) models for dialogue act classification. The GloVe algorithm represents words as vectors in a high-dimensional space, enabling the models to capture the semantic relationships between words.

The Bidirectional RNN (LSTM) model is expected to capture the contextual dependencies of the utterances in both forward and backward directions, while the CNN (textCNN) model is expected to capture the local patterns of the utterances. Both models are expected to produce more accurate results compared to Naïve Bayes.

Finally, we will be using the BERT (Bidirectional Encoder Representations from Transformers) algorithm along with the MLP (Multilayer Perceptron) model for dialogue act classification.

The BERT algorithm is a state-of-the-art model that uses transformer-based architecture to capture the contextual dependencies of the utterances. The MLP model is expected to produce more accurate results by applying nonlinear transformations to the input features. This model is expected to achieve the highest accuracy and efficiency in dialogue act classification.

3 Data collection

The HCRC Map Task Corpus is a remarkable resource for researchers and students of linguistics, and computational linguistics alike. Comprising a set of 128 dialogues, this corpus has been meticulously recorded, transcribed, and annotated for a wide range of behaviors.

The dialogues within the HCRC Map Task Corpus were designed to simulate a navigation task, where one participant is required to navigate another through a maze using a map. The corpus captures the interaction between the two participants, including the directions given by the navigator, the questions asked by the traveler, and the responses of the navigator to those questions. There are twelve moves in the coding scheme:

Six initiating moves:

• instruct - commands the partner to carry out an action
• explain - states information which has not been elicited by the partner
• check - requests the partner to confirm information that the checker has some reason to believe, but is not entirely sure about
• align - checks the attention or agreement of the partner, or his/her readiness for the next move
• query-yn - asks the partner any question which takes a ”yes” or ”no” answer and does not fall into the previous two categories
• query-w - any query which is not covered by the other categories

Five response moves:

• acknowledge - a verbal response which minimally shows that the speaker has heard the move to which it responds
• reply-y - any reply to any query with a yes-no surface form which means ”yes”, however that is expressed
• reply-n - a reply to a a query with a yes/no surface form which means ”no”
• reply-w - any reply to any type of query which doesn’t simply mean ”yes” or ”no”
• clarify - a repetition of information which the speaker has already stated, often in response to a check move

One pre-initiating move:

• ready - a move which occurs after the close of a dialogue game and prepare the conversation for a new game to be initiated

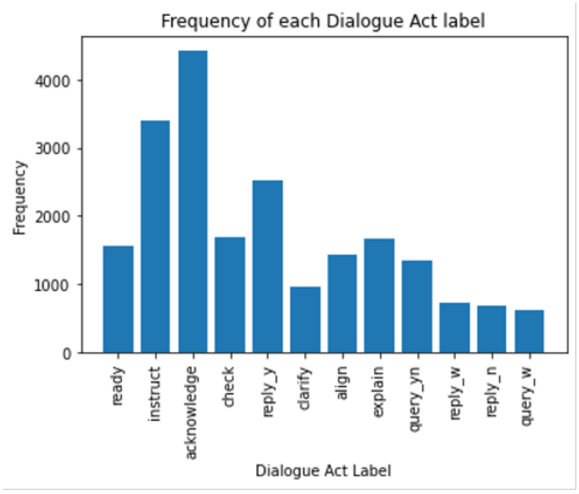


Figure 1 : Frequency of Dialogue Act in the Map Task Corpus

4 Bag-of-Words with Naïve Bayes classifier

We used the Bag-of-Words with a Naïve Bayes classifier to perform multi-class classification on the dialogue act labels in the MapTaskCorpus dataset. The data is first read from a text file and split into training and testing sets. Then, the CountVectorizer function is used to extract features from the training and testing data. These features are used to train the Naïve Bayes classifier, which is then used to predict the dialogue act labels of the test data.

One of the advantages of using the Bag-of-Words model is that it is simple to implement and can be effective for certain types of text classification tasks. Specifically this approach works well for datasets with a limited number of classes and small vocabulary sizes. However, the Bag-of-Words model has limitations, such as the inability to capture the context and order of the words in the document, resulting in sparse and high-dimensional feature vectors.

In fact Naïve Bayes assumes independence between the features, which is not always true in natural language, and may lead to suboptimal performance when the features are correlated.

In terms of accuracy, we report an average accuracy score of 0.566, which means that the classifier correctly predicted the dialogue act label for approximately 0.566 of the test data. While this accuracy score may seem relatively low, it is important to note that the performance of the classifier may vary depending on factors such as the specific dataset being used, the choice of feature extraction and the amount and quality of training data available.

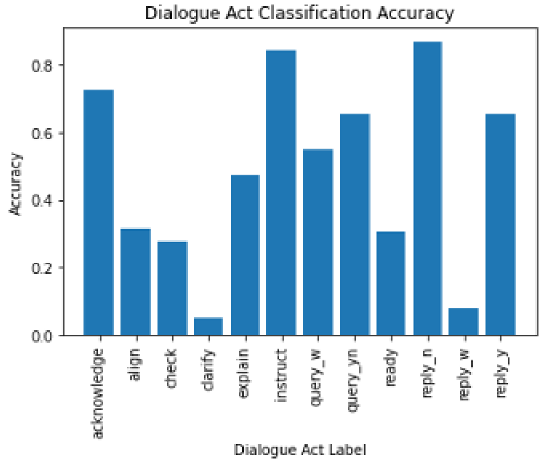


Figure 2 : Classification accuracy for the Map Task Corpus using Bag-of-Words model and Naïve Bayes classifier.

5 GloVe and Deep Learning Models

Instead of using Bag-of-Words with a Naïve Bayes, other feature extraction methods, such as word embeddings, could be used to capture the semantic and syntactic relationships between words in the document. Deep learning models, such as Recurrent Neural Networks (RNNs) or Convolutional Neural Networks (CNNs), can also be used to model the sequential and hierarchical structure of the text and achieve better performance on complex natural language tasks.

5.1 Global Vectors for Word Representation

GloVe (Global Vectors for Word Representation) (Pennington et al., 2014) is a popular pre-trained word embedding model that learns word representations based on co-occurrence statistics of words in a large corpus of text. These embeddings capture semantic and syntactic relationships between words and can be used to initialize the word embeddings layer in a neural network. We used the pre-trained GloVe embeddings of 100 dimensions and an embedding matrix is created for the words in the dataset.

5.2 Preprocessing the Dataset

Next, tokenization is the process of converting text into a sequence of tokens, this step is essential in natural language processing, as it reduces the dimensionality of the input data and makes it easier to process it. The Tokenizer class is also used to fit the vocabulary of the input text data and convert each sentence into a sequence of integers.

After tokenization, the sequences are padded to a fixed length. Padding is the process of adding zeros to the sequences that are shorter than the fixed length, which makes them all the same length. This is done because neural networks require inputs of the same size to process.

Finally, the labels are one-hot encoded. One-hot encoding is a process of converting categorical variables (i.e., labels) into numerical representations that can be used by the neural network.

5.3 Glove with Bidirectional LSTM

A Model using a Bidirectional LSTM is then used for text-classification (Maas et al., 2011). Bidirectional LSTM is a type of LSTM that processes the input sequence in both forward and backward directions, allowing the network to capture contextual information from both past and future inputs. More precisely the model architecture used is a sequential neural network with three layers: an embedding layer, a Bidirectional LSTM layer, and a dense output layer.

The embedding layer maps the input words to a high-dimensional vector space where the relationships between the words can be better understood by the model. In details, the embedding layer takes as input the number of words (i.e., the vocabulary size), the dimensionality of the embedding space (100), the pre-trained word embeddings, and the maximum length of the input sentences.

The Bidirectional LSTM layer is then used to capture the contextual relationships between words in a sentence. The LSTM layer has 100 units and uses a dropout rate of 0.2 to prevent overfitting. The recurrent dropout parameter is also set to 0.2 to further regularize the model.

The dense output layer is a fully connected layer that maps the output of the Bidirectional LSTM layer to the number of output classes. The dense layer uses a softmax activation function to convert the outputs to a probability distribution over the classes.

The model is compiled using the categorical-crossentropy loss function, which is suitable for multi-class classification problems. The adam optimizer is used to minimize the loss function during training, and the accuracy metric is used to monitor the performance of the model.

It's worth noting that the obtained accuracy of 0.6174 may be considered decent or not depending on the task and the dataset used. Therefore, it's important to compare the performance of the model with other baseline models and evaluate whether the accuracy is satisfactory or not.

5.4 Glove with TextCNN

In that end a CNN model is then used for multi-class text classification (Kim, 2014). The input layer takes as input the maximum length of the sentences. The embedding layer maps the input words to a 100-dimensional vector space using pre-trained word embeddings.

The model includes three convolutional layers with filter sizes of 3, 4, and 5. Each convolutional layer has a specific number of filters (i.e., 64), which scan the input sequences to extract relevant features. Also, the Convolution 1D layer applies a one-dimensional convolution operation to the input sequences, while the Max Pooling 1D layer performs a downsampling operation by taking the maximum value of each feature map.

The output of each convolutional layer is then flattened and concatenated into a single feature vector. The concatenated layer is fed into a dropout layer, which helps prevent overfitting by randomly dropping out some of the neurons during training. Finally, the output layer uses a softmax activation function to convert the outputs to a probability distribution over the classes.

The model is compiled using the adam optimizer and the categorical-crossentropy loss function, which is suitable for multi-class classification problems. The accuracy metric is used to monitor the performance. However we do not see any big improvement in the accuracy that still is of 0.60.

5.5 Bidirectional LSTM or TextCNN

One key advantage of using TextCNNs over LSTM (Long Short-Term Memory) models for text classification is that they are faster and require less memory to train. This is because LSTMs process sequential data in a way that is computationally expensive and requires a large amount of memory, which makes them slower and less scalable. In contrast, TextCNNs are simpler and more efficient, and can be trained on large datasets without the need for specialized hardware.

Additionally, TextCNNs are better suited for identifying local features within a text, such as n-grams, which can be important in text classification tasks. This is because CNNs apply filters to small windows of the input, allowing them to capture local features in a more effective manner compared to LSTMs, which process the input sequentially and may miss out on important local features.

6 BERT with MLP

BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained language model that has shown impressive performance on various NLP tasks (Devlin et al., 2018). It uses a transformer architecture, which allows it to capture contextual information and relationships between words in a sentence. This is especially useful in tasks such as text classification where understanding the meaning of the sentence is crucial for accurate classification.

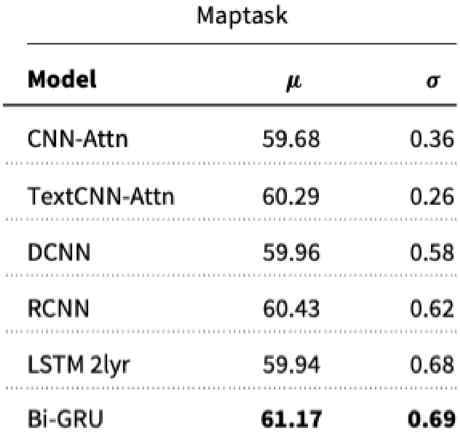
In contrast, CNN and RNN are both shallow architectures that are not specifically designed for capturing the contextual information and relationships between words in a sentence. While both CNN and RNN have been used for text classification tasks and have shown good performance in some cases (Duran et al., 2021), they have limitations in capturing the complex relationships between words that BERT can handle.

Therefore, by using BERT as a feature extractor and then feeding the features into an MLP (Multi-layer perceptron), we are leveraging the strengths of both BERT and MLP. More precisely BERT can capture the complex relationships between words in a sentence, while MLP can provide an effective way to combine and classify these features. The MLP consists of two fully connected layers and uses ReLU activation function in the first hidden layer and softmax activation function in the output layer. This approach is effective in various NLP tasks, including text classification (Duran et al., 2021), and achieves state-of-the-art results in many cases, however we still get an accuracy around 0.60.

7 Conclusion

In this article, we explored various text classification methods on the HCRC Map Task Corpus. Specifically, we used Bag-of-Words model with Naïve Bayes, Glove with Bidirectional LSTM, Glove with CNN, and BERT with MLP.

We found that all of the models achieved an accuracy of around 0.60, which is close to the benchmark accuracy (Duran et al., 2021). This indicates that these models are useful for classifying the text in this corpus, but there may be limitations to the data or the models themselves that prevent higher accuracy.



Maptask

Model	μ	σ
CNN-Attn	59.68	0.36
TextCNN-Attn	60.29	0.26
DCNN	59.96	0.58
RCNN	60.43	0.62
LSTM 2lyr	59.94	0.68
Bi-GRU	**61.17**	**0.69**

Figure 3 : Test set accuracy for each of the supervised models on the Maptask data (Duran et al., 2021)

The Bag-of-Words model with Naïve Bayes is a simple and efficient method, but it may not capture the semantic meaning of the words in the text.

The Glove with Bidirectional LSTM and Glove with CNN methods are more complex and can capture more complex relationships between words, but they may require more data and more computational power to achieve higher accuracy.

Finally, the BERT with MLP method is a state-of-the-art technique that can capture context and meaning very well. However, it may require more fine-tuning and training on specific data sets to achieve higher accuracy.

In conclusion, while the accuracy of these models may not be very high, they are still useful for classifying text in the HCRC Map Task Corpus. These results provide insights into the strengths and limitations of different intent classification methods and can be used to inform future research in this area.

8 References

[1] University of Edinburgh. HCRC Map Task Corpus LDC93S12. Web Download. Philadelphia: Linguistic Data Consortium, 1993.

[2] Pennington, J., Socher, R., Manning, C. (2014). Glove: global vectors for word representation. Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532–1543).

[3] Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., Potts, C. (2011). Learning word vectors for sentiment analysis. Proceedings of the 49th annual meeting of the associ- ation for computational linguistics: Human language technologies-volume 1 (pp. 142–150).

[4] Kim, Y. (2014). Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

[5] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K. (2018). Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[6] Duran, N., Battle, S., Smith, J. (2021). Sentence encoding for Dialogue Act classification. Natural Language Engineering, 1-30. doi:10.1017/S1351324921000310