Challenge: A deep dive into Dialogue Act Recognition (2024-2025)

By: *Ward Van den Bossche*, **Student ID**: *0605083*

Introduction

This project was developed as part of the Current Trends in Artificial Intelligence course at VUB, under the supervision of Dr. Leticia Arco Garcia. The objective of this work is to train a deep learning model to predict dialogue acts from sequences of utterances within a dialogue.

The dataset used for this project is best summarized by a statement from the official repository: "*The Schema-Guided Dialogue (SGD) dataset consists of over 20k annotated multi-domain, task-oriented conversations between a human and a virtual assistant*" [1].

These conversations span various domains such as banking, travel, and weather. Of particular relevance to this project are the dialogue act annotations, which are provided for both the system and the user. Some example dialogue acts include `INFORM`, `REQUEST`, and `CONFIRM`. A complete list of possible values can be found in the dataset's GitHub README.md. Each dialogue contains multiple turns, and each turn is characterized by it's speaker, the spoken utterance and the dialogue act.

We will be working on a subdataset of SGD, provided by VUB's AI Lab in the files TrainSet.json and TestSet.json *. The statistics can be observed below:

Indicators	Train	Test
Total dialogues	5403	1331
Total utterances	82588	16850
Total domains	14	17
Total USER dialogue acts	11	11
Total SYSTEM dialogue acts	10	10
Average dialogues per domain	385,92	78,29
Average utterances per dialogue	15,28	12,66

We will consider RNN approaches in particular. In the process we will report on performance and try to analyse the models from different viewpoints. We will start from a basic model in the first implementation, inspired by the Deep Learning Course I followed at VUB, where an RNN based implementation was used in a sentiment analysis exercise. This implementation uses only the utterance as context, and is not pretrained on any other data.

The second implementation follows the methodology proposed by Kumar et al. [2]. As cited there, the feature extractor is based on a pretrained mLSTM model used in Radford et al. [3], and its PyTorch port [4] was adopted for acquiring rich utterance embeddings.

* For debugging purposes, and in order to take a proper look at the documents, formatted versions can be found in TrainSet_formatted.json and TestSet_formatted.json.

How To Run

To run the code the reader must: - install the required libraries (see requirements.txt).

- download the **mlstm_ns.pt** from here. Place this file in the implementation_2 folder.
- Follow the two notebooks implementation_1.ipynb and implementation_2.ipynb respectively. The notebooks rely on different python scripts, which will be referenced at the appropriate times when following the notebooks. Simply run the cells in order, or observe the precomputed cells to view the results, most code cells are preceded by a description of what happens.

Results

Implementation 1:

```
Evaluation Results:
Top-1 Accuracy: 83.98%
Top-5 Accuracy: 97.79%
Balanced Accuracy: 69.87%
Avg. Time per epoch: 75.70 seconds

Classification Report:
                                                 precision    recall  f1-score   support

                                  SYSTEM_CONFIRM       0.92      0.90      0.91      1113
                                  SYSTEM_GOODBYE       1.00      1.00      1.00      1331
                                   SYSTEM_INFORM       0.83      0.81      0.82       719
           SYSTEM_INFORM|SYSTEM_NOTIFY_SUCCESS       0.78      0.76      0.77       310
  SYSTEM_INFORM|SYSTEM_OFFER|SYSTEM_NOTIFY_FAILURE       0.50      0.29      0.37        17
         SYSTEM_NOTIFY_FAILURE|SYSTEM_REQ_MORE       0.94      0.95      0.94        79
                           SYSTEM_NOTIFY_SUCCESS       0.88      0.94      0.91       499
                                    SYSTEM_OFFER       0.82      0.77      0.80       806
                             SYSTEM_OFFER_INTENT       0.78      0.97      0.87       383
              SYSTEM_OFFER|SYSTEM_INFORM_COUNT       0.93      0.95      0.94       608
            SYSTEM_OFFER|SYSTEM_NOTIFY_FAILURE       0.60      0.43      0.50        14
                                  SYSTEM_REQUEST       0.96      0.95      0.95      1961
                                 SYSTEM_REQ_MORE       0.99      0.97      0.98       585
```

```
                                     USER_AFFIRM       0.81      0.73      0.77       572
                              USER_AFFIRM_INTENT       0.68      0.78      0.73       139
                  USER_AFFIRM_INTENT|USER_INFORM       0.66      0.65      0.65       103
                                     USER_INFORM       0.83      0.79      0.81      1961
                              USER_INFORM_INTENT       0.85      0.80      0.82       658
           USER_INFORM_INTENT|USER_NEGATE_INTENT       0.00      0.00      0.00         0
                  USER_INFORM_INTENT|USER_SELECT       0.60      0.80      0.69       167
               USER_INFORM_INTENT|USER_THANK_YOU       0.00      0.00      0.00        12
                  USER_INFORM|USER_INFORM_INTENT       0.66      0.65      0.65       772
USER_INFORM|USER_INFORM_INTENT|USER_NEGATE_INTENT       0.00      0.00      0.00         0
      USER_INFORM|USER_INFORM_INTENT|USER_SELECT       0.57      0.81      0.67       117
   USER_INFORM|USER_INFORM_INTENT|USER_THANK_YOU       0.00      0.00      0.00        14
                         USER_INFORM|USER_NEGATE       0.80      0.75      0.77       216
                   USER_INFORM|USER_REQUEST_ALTS       0.69      0.61      0.65       184
                                     USER_NEGATE       0.00      0.00      0.00         7
                              USER_NEGATE_INTENT       0.39      0.81      0.53        69
                  USER_NEGATE_INTENT|USER_GOODBYE       0.66      0.64      0.65        72
                         USER_NEGATE|USER_GOODBYE       0.00      0.00      0.00         9
                       USER_NEGATE|USER_THANK_YOU       0.88      0.82      0.85       587
                                    USER_REQUEST       0.90      0.86      0.88       719
                               USER_REQUEST_ALTS       0.81      0.86      0.83       152
                         USER_REQUEST|USER_AFFIRM       0.85      0.83      0.84       340
                                     USER_SELECT       0.63      0.79      0.70       515
                         USER_SELECT|USER_GOODBYE       0.67      0.67      0.67       279
                                  USER_THANK_YOU       0.72      0.84      0.77       377
                      USER_THANK_YOU|USER_GOODBYE       0.76      0.70      0.72       384

                                        accuracy                           0.84     16850
                                       macro avg       0.65      0.66      0.65     16850
                                    weighted avg       0.84      0.84      0.84     16850
```

Implementation 2

```
Evaluation Results:
Top-1 Accuracy: 94.49%
Top-5 Accuracy: 99.49%
Balanced Accuracy: 85.61%
Avg. Time per epoch: 5.22 seconds

Classification Report:
                                                precision    recall  f1-score   support

                                  SYSTEM_CONFIRM       0.98      0.95      0.96      1113
                                  SYSTEM_GOODBYE       0.98      0.98      0.98      1331
                                   SYSTEM_INFORM       0.99      0.95      0.97       719
            SYSTEM_INFORM|SYSTEM_NOTIFY_SUCCESS       0.99      0.96      0.98       310
```

ablation study: window size vs performance

Figure 1: ablation study: window size vs performance

```
              SYSTEM_INFORM|SYSTEM_OFFER|SYSTEM_NOTIFY_FAILURE       0.79      0.88      0.83        17
                   SYSTEM_NOTIFY_FAILURE|SYSTEM_REQ_MORE       0.99      0.92      0.95        79
                                   SYSTEM_NOTIFY_SUCCESS       0.97      0.99      0.98       499
                                            SYSTEM_OFFER       0.91      0.97      0.94       806
                                     SYSTEM_OFFER_INTENT       0.92      0.98      0.95       383
                       SYSTEM_OFFER|SYSTEM_INFORM_COUNT       0.97      0.96      0.97       608
                     SYSTEM_OFFER|SYSTEM_NOTIFY_FAILURE       1.00      0.93      0.96        14
                                          SYSTEM_REQUEST       1.00      0.99      0.99      1961
                                         SYSTEM_REQ_MORE       0.96      0.98      0.97       585
                                             USER_AFFIRM       0.98      0.97      0.97       572
                                      USER_AFFIRM_INTENT       0.90      0.93      0.91       139
                          USER_AFFIRM_INTENT|USER_INFORM       0.89      0.85      0.87       103
                                             USER_INFORM       0.99      0.98      0.98      1961
                                      USER_INFORM_INTENT       0.96      0.93      0.94       658
                   USER_INFORM_INTENT|USER_NEGATE_INTENT       0.00      0.00      0.00         0
                          USER_INFORM_INTENT|USER_SELECT       0.81      0.90      0.85       167
                       USER_INFORM_INTENT|USER_THANK_YOU       0.44      0.33      0.38        12
                          USER_INFORM|USER_INFORM_INTENT       0.94      0.95      0.95       772
       USER_INFORM|USER_INFORM_INTENT|USER_NEGATE_INTENT       0.00      0.00      0.00         0
              USER_INFORM|USER_INFORM_INTENT|USER_SELECT       0.84      0.84      0.84       117
           USER_INFORM|USER_INFORM_INTENT|USER_THANK_YOU       0.60      0.21      0.32        14
                                 USER_INFORM|USER_NEGATE       0.95      0.90      0.93       216
                           USER_INFORM|USER_REQUEST_ALTS       0.84      0.86      0.85       184
                                             USER_NEGATE       0.75      0.43      0.55         7
                                      USER_NEGATE_INTENT       0.60      0.72      0.66        69
                         USER_NEGATE_INTENT|USER_GOODBYE       0.71      0.76      0.73        72
                                USER_NEGATE|USER_GOODBYE       0.60      0.67      0.63         9
                              USER_NEGATE|USER_THANK_YOU       0.99      0.99      0.99       587
                                            USER_REQUEST       0.97      0.94      0.95       719
                                       USER_REQUEST_ALTS       0.91      0.93      0.92       152
                                USER_REQUEST|USER_AFFIRM       0.95      0.96      0.95       340
                                             USER_SELECT       0.85      0.89      0.87       515
                                USER_SELECT|USER_GOODBYE       0.79      0.68      0.73       279
                                          USER_THANK_YOU       0.80      0.77      0.79       377
                             USER_THANK_YOU|USER_GOODBYE       0.71      0.83      0.77       384

                                                accuracy                           0.94     16850
                                               macro avg       0.83      0.81      0.81     16850
                                            weighted avg       0.95      0.94      0.95     16850
```

Ablation Study: Window Size vs Performance (implementation 2 only)

System Specs

Experiments were run on:

- CPU: AMD RYZEN 5 3600
- GPU: NVIDIA GeForce RTX 2070 SUPER
- RAM: 16 GB

Note: Training times may vary significantly on different hardware.

Discussion

In order to get the full context of this discussion, the reader is urged to go through the notebooks first, as there they will find more results and visualisations about the experiments.

Both models show a strong overall performance, but the performance lowers w.r.t. rare dialogue acts. This is reflected in a performance drop of about 10%-14% when comparing regular with balanced accuracy. In a way this makes sense, since some combinations of intents simply are very rare, the 0 score in `USER_NEGATE|USER_GOODBYE`, for example, shows that it suffers from data scarcity (only 9 samples). The second implementation, however does well even in the absence of data, this suggests that the richer feature embeddings from the pretrained model, and context of previous utterances played a key role in improving performance. Intuitively, this makes sense, since a dialogue act like `USER_NEGATE` only makes sense after a system offer. The richer feature embeddings might help capture more nuanced differences in words.

A time per epoch of 75.70 seconds for the first implementation seems relatively low, but considering the small size of the subdataset, this was expected. At 50 epochs this training process took just over 1 hour to run. What's remarkable is the very low running time of the second implementation at an average time per epoch of 5.22 seconds. The fact that we precomputed the embeddings using the pretrained model for the utterances plays a big factor here. It must be noted that this process itself took quite some time as well, running for 15 minutes to process the test set alone.

The visualisations for both implementations show that a lot of the misclassifications happen between acts with multiple labels and their sub-label counterparts, for example `USER_THANK_YOU` is sometimes confused with `USER_THANK_YOU|USER_GOODBYE`. This is a byproduct of how the dataset was preprocessed, and for this reason it would be very interesting to see a multi-label variant of this project. We see that the t-SNE plots show proper clustering of dialogue acts, especially in the last implementation.

The ablation study for the second implementation shows that for this particular task and dataset, a context of 1 previous utterance does best. Increasing the context window results in a drop in performance, likely due to utterances from early states adding too much noise to our final hidden representation. As it stands, we use the mean hidden state, but it would be interesting to study the results on the final hidden state version as well. We also note the vast increase in training time as the window size increases, this could also be due to memory constraints, since the experiments were run on a pc with one GPU.

In conclusion, the first implementation shows strong performance in spite of it's lack of optimizations (less context, no pretrained model). The second implementation solidifies the idea that more context matters when it comes to dialogue act classification, but only to a certain extent, and depending on the task at hand. Finally, we can leverage pretrained embeddings trained on large corpora to transfer semantic knowledge.

Future work

When it comes to dialogue act recognition, and this particular dataset, there is still a lot to explore in the future. Hopefully these models can serve as a benchmark for future implementations.

A list of possible additions includes: - Hyperparameter tuning for even better performance (hidden layer size, embedding size, etc.) - Consider a Multi-Label variant of the dataset, where instead of grouping labels together like `USER_THANK_YOU|USER_GOODBYE`, we keep them separate and allow an utterance to have multiple labels - Run an ablation study where either of these has been implemented: - stopwords have been filtered out in the precomputing step - where the first implementation has access to multiple utterances as context - the last hidden state is used in both implementations instead of the average - Test the performance of classical ML algorithms like decision trees, support vector machines, etc. - Consider different representation strategies for (word) embeddings, e.g. Word2Vec or GloVe. - Consider other Deep Learning Models such as transformers: BERT, GPT, etc. - Test performance on the full dataset, and the sgd-x version to test generalization.

References

[1]
The Schema-Guided Dialogue Dataset
Rastogi, A., Zang, X., Sunkara, S., Gupta, R., & Khaitan, P. (2020).
Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset, in *Proc. AAAI Conf.*, pp. 8689–8696

[2]
A Context-based Approach for Dialogue Act Recognition using Simple Recurrent Neural Networks
Bothe, C., Weber, C., Magg, S., & Wermter, S. (2018).
In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

[3]
Learning to Generate Reviews and Discovering Sentiment
Radford, A., Jozefowicz, R., & Sutskever, I. (2017). *arXiv preprint arXiv:1704.01444*.

[4]
PyTorch Sentiment Neuron (Port)
GitHub repository by guillitte.
Linked from the official OpenAI repository as a third-party PyTorch implementation.