

Task

Implement the core functionalities of a conversational agent (CA)

- Domain identification/ Dialog act prediction
- Content extraction from User utterances (semantic frame slot filling)
- Agent move prediction

More about each subtask in the *Evaluation notebook* (link below).

Use a variety of techniques different for each subtask or in combination with each other, e.g.:

- Sentence detection, lemmatization, part-of-speech tagging, syntactic parsing
- Pre-trained word embeddings
- Basic neural network architectures (training)
- Transformer-based models (fine-tuning)
- Entity extraction with embedding-based similarity search to define the categorical values
- Entity extraction as a sequence-labeling task
- Entity extraction as a summarization task
- Classification as k-nearest neighbors with sentence-embedding-based similarity search
- Linguistically motivated rule-based post-processing to adjust the model outcome

Dataset

Data: https://huggingface.co/datasets/multi_woz_v22

Paper: <https://aclanthology.org/2020.nlp4convai-1.13.pdf>

Domain: hotel and restaurant reservation dialogues

For each component, separate training and evaluation datasets in a specific format for the task shall be derived from the corresponding subsets (train/validation/test).

Use correct ground truth inputs for each component rather than outputs of the previous component to avoid accumulating the errors in the pipeline.

Evaluation

Evaluation notebook shows how to load and evaluate the performance within each task (complete missing parts of the code by analogy):

<https://colab.research.google.com/drive/1ymtXg787eJtwaKI7Exu6ckdUFifyGqkd?usp=sharing>

Other expected evaluations:

- Provide classification reports for every model that show the scores for individual classes
- What ratio of the dialogues were successful, i.e. values for the most possible number of the compulsory slots were extracted correctly?
- What ratio of the ground truth slot names are known by the end of the dialogue (at least some values have been assigned even if not correct)? What ratio of the slots contain up-to-date values by the end of the dialogue, i.e. correctly updated within the dialogue?
- What is the average execution time per dialogue? Calculate it separately for individual stages if all the models cannot be loaded in the memory at once.

Improvements of the components

In case some components perform poorly, try to change the technique, adjust training parameters, shuffle the examples, augment the examples, or/and fine-tune models pre-trained on a similar task (transfer learning). For each component, find the most relevant information

from the dialogue that shall be used in the input to the component (e.g., concatenation of the current utterance and values from the dialogue history in a single input).

Demonstrate the results

- submission of the code
- pptx presentation of how the tasks have been realized including evaluations
- live demonstration of the CA
 - with examples from the dev/test datasets
 - with new prepared utterances for the user (or spontaneously typed in within the demo) beyond those from the dataset (agent's moves from the dataset's dialogue can be used, no need to simulate them).

Useful links

Basic language analysis techniques:

<https://spacy.io/usage/linguistic-features#sbd>

<https://stanfordnlp.github.io/stanza/> – can be slow

Sequence labeling

<https://github.com/glample/tagger> – operating through text files, not convenient

<https://github.com/kamalkraj/BERT-NER> – operating through text files, not convenient

<https://github.com/huggingface/transformers/tree/main/examples/pytorch/token-classification>

Extraction as summarization

<https://github.com/huggingface/transformers/tree/main/examples/pytorch/summarization>

Text classification with transformers

<https://github.com/huggingface/transformers/tree/main/examples/pytorch>

<https://github.com/huggingface/transformers/tree/main/examples/pytorch/text-classification>

Notebook for text classification with simpler networks and pre-trained fixed embeddings

<https://colab.research.google.com/drive/1H30CrbKzIBdJ9owjKzngcFpaGr8HdgYv?usp=sharing>

text classification as sentence similarity with k-nearest neighbors:

https://huggingface.co/models?pipeline_tag=sentence-similarity&sort=trending

Hugging Face model cards to check the outcomes and the way to load the model, e.g.:

<https://huggingface.co/nlptown/bert-base-multilingual-uncased-sentiment>

<https://huggingface.co/sentence-transformers/all-mpnet-base-v2>