Cezanne-ai[[1]](#footnote-1) project

This document gives details on implementing Cezanne-ai open-framework described in the **research paper “Cezanne-ai: a conversational AI open-framework for multi-domains, all the languages and limited data”**. We are not disclosing the actual coding from the beginning as we believe that every developer has tricks that can improve the outcome, but we are open to sharing codes and best practices. This project[[2]](#footnote-2) has the objective to build alternatives to current chatbots, operating systems, search engines and programming. If a direct path of communicating with the machine in the natural language is developed (inside a structured conversation) then coding, intermediaries and operating interfaces could become obsolete. This is not a new optic in conversational AI, but we believe that our framework has more arguments to become the new UI[[3]](#footnote-3).

Furthermore, the framework is designed to easily customize conversational bots for different domains/industries (also multi-domains) and the specificities of different business models. Guidance for language customization and database requirements are also provided every step of the way. As a general recommendation[[4]](#footnote-4) we encourage not to overuse MTurk or unlabeled data and to involve screenwriters and linguists in the processes as a fundamental principle.



Chapters/functional specifications:

1. Build a Natural Language Understanding algorithm based on the Pirkin 1&2 models proposed in the research paper. A back-up model with three possible models will be implemented.
2. Develop algorithms for conversational states and policies based on the Pirkin 3 model from the research paper.
3. Generate bot outputs algorithms based on the Pirkin 4&5&6 models presented in the research paper.
4. Adapting/customizing the framework in order to use it in other fields of activities/industry, other cities, or with other databases/ expert knowledge or answers. Books can be directly added and used by Cezanne-ai as knowledge in the conversation.

Due to our expertise, we are going to make constant referrals to conversational bots specialized in **core-inputs** (you can choose whatever domain you want, or even multi-domains), **forms** (example: reservations) that is capable to have also **deep conversations** through books queries and give answers using self-generative policies also (similar to the ones humans are learning/using, AGI)[[5]](#footnote-5).

Data prerequisites[[6]](#footnote-6)

One of the important benefits of the Pirkin models is the fact that they work with limited datasets/corpuses. This doesn’t mean that we don’t need important data, but the required data already exists:

1. for all the languages (as they are prerequisites in the learning curricula of any school),
2. at the company level (especially that we give alternatives),
3. or, it can be easily created as the models are working with sentence-intents that is simplifying the structuring/labeling of the data.

Code implementation

As we are proposing an open-framework also in terms of the code we will not detail the instructions/functions/classes…, but we will provide a dummy-code with step-by-step description of the code, and we will be proposing for different pipelines the following approaches to be used in terms of code:

* **Build code from scratch** (as we are not aware of existing codes that can help us with the pipeline)
* **Use existing codes** (as we believe that open-source codes exist for the specific pipeline)
* **Adapted code** (existing code that can be adapted to cover the pipeline specificities)

1. Implementation of Natural Input Understanding models (NIU-NLU)

The most important component of this project is the NLU/NIU layer, which incorporates contextual thinking based on fundamentals presented in the research paper (more like a labyrinth network than a NN).

In the 4th chapter we will give details on how to build a solution for different domains by using the framework. For example, in medicine or juridical domain you can add specialized books (laws and treaties) to back up the advisory session. NER prioritization could be the drugs and forms could be a scheduled visit to a doctor.

Our model will have the capability to do automatic training & labeling or scheduled database updates. It is not recommended to make daily changes in training specific situations. In order to use the framework for different languages/countries, we strongly recommend a fundamental review of your algorithms taking into account the language specificities (be aware that also the order can have implications and please see comments in the pipelines related to this topic as we want to guide you through the framework specificities).

* 1. Pirkin 1 model. Machine Education:
     1. Auto- Correct

**Objectives:**

* Inserting diacritics (if the case).
* Addressing abbreviations.
* Correcting UNK words with one/two letters deviation.

**Language specificities:** yes (see diacritics)

**Dependencies:** Emoji/Grammar-Semantics

**Database/ Vocabularies needed:** lexicon/ abbreviations/ specific vocabularies/Books& chapters titles

**To dos:**

1. A local input processing is needed. Only words are in scope of this algorithm.
2. Searching for UNK words in the input corpus.
3. See overlaps between specific vocabularies and general vocabularies.
4. Deploying algorithm for mapping UNK with vocabularies (one & two letter deviations will be addressed) – in that way, if the specific vocabulary words have articles, they will be brought to their base.
5. If in the mapping process more than one different vocabulary word will be found, all words will be marked for Grammar/Semantics analysis.
6. Abbreviations will be mapped with Abbreviation's vocabularies and if found will be marked for Emoji algorithm.

**Python Code:** Use existing codes. Special focus on diacritics and special language characters.

* + 1. Input Processing I.

**Objectives:**

* Linguistic evaluation. Addressing special characters of the language. (For example: removing hyphens by separating into 2 words;
* Removing words/ characters not needed;
* Addressing numerical data, punctuation, emoji;

**Language specificities:** yes (check language specificities);

**Dependencies:** DER/CVM/EER/Grammar-Semantics;

**Database/ Vocabularies needed:** Lexicon/ Dictionaries for numerical data and special dates & events /Emoji.

**To dos:**

1. Addressing special characters of the language in order for all words/characters to be included into a list that can be processed further (we will leave this task for linguists to assess).
2. All numerical characters found (including in a special vocabulary) are marked for DER.
3. Emoji are marked and sent to EER.
4. Mark nonexistent vocabulary words with UNK.
5. Mark words that exist more than once in vocabularies for Grammar/Semantics analysis.
6. Remove personal names, telephone numbers, visual inputs, links, other characters like $#@ and the related words.

**Python Code:** Use existing codes

* + 1. Composedwords

**Objectives:**

* Identifying composed words that can be treated as a single word;
* Identifying composed words that need to be addressed separately;
* Identifying adverbs & superlatives of specific adjectives and include them in categories;

**Language specificities:** yes (see expressions and ironies that are specific to each language);

**Dependencies:** Grammar-Semantics/Untrained NIU/ DER/ SPCA/ Embeddings/ Back up algorithm/ Reaction analysis/NER;

**Database/ Vocabularies needed:** Lexicon/Dictionaries for expressions, ironies, quotes, metaphors/Database for adverbs & superlatives for specific adjectives/ Specific vocabularies + Books/chapter titles– for the items composed of min 2 words.

**To dos:**

1. Find in the input words that appear in the NER specific vocabularies + Books/chapter titles and mark them for NER (special attention for the items composed of min 2 words).
2. Find expressions, metaphors, quotes and mark them for Untrained NIU. Remove them from the main model.
3. Find ironies and mark them for Reaction analysis. Remove them from the main model.
4. Find adverbs and include them in 5 categories: how often, when, how, how much and where. First 4 categories are marked for DER, the fifth in the SPCA pipeline.
5. Specific adjectives and their superlatives are included in 5 categories: plus-positive, positive, neutral, negative, minus-negative; in scope of Embeddings and DER relatively vector.

**Python code:** use existing codes

* + 1. NER

**Objectives:**

* Determining if the input is a core input or not. Make NER classification and GDPR clean-ups.

**Language specificities:** no;

**Dependencies:** SPCA/embeddings/ triggering words/CVM;

**Database/ Vocabularies needed**: NER & GER vocabularies /Books- NER vocabularies.

**To dos:**

1. Identify NER/GER by importance and classify them accordingly: NER1, NER2, NER3…. The same with books NER: NER01/NER02…. Additional NERs will be mapped through integration with existing slots available.
2. If we have other upper-case words inside the sentences that don’t appear in the database, it will be eliminated for GDPR reasons.
3. Identifications of the domains/industry based on the NER identifications. If NER from more domains/industries are identified, then the first counts for the domain identification.
4. If NER 1/NER1.0 is identified, then it will be marked as Subject. If we have NERs from the same domains then the prioritization counts and the 2nd,…NER will be marked as complements. If deep conversational intents do not have NERs, and the subject is CVM, 3rd row.

**Python code:** Adapted code (existing code that can be adapted to cover the pipeline specificities)

* + 1. Emoji/Abbreviations (EER)

**Objectives:**

* Gathering the emoji/abbreviations from auto-correct, processing and mark them for Untrained NIU/Reactions;

**Language specificities:** no;

**Dependencies:** Auto-correct /Processing I/Untrained NIU/ Reaction analysis;

**Database/ Vocabularies needed**: Emoji/Abbreviations vocabularies;

**To dos:**

1. Mark sentences with emoji/abbreviations for Untrained answers.
2. Classify all the emoji/abbreviations in 6 categories (sad, blink, kiss, smile, cool, laughing out loud).
3. Sad and Smile will be sent to the Reaction analysis pipeline.
4. The rest will be sent to Untrained NIU.

**Python code:** Adapted code.

* + 1. Grammar/Semantics

**Objectives:**

* Addressing words that are coming from the linguistic evaluation;
* Retrieving information for CVM (verbs);
* Addressing UNK words by finding their POS probabilities;
* Addressing words that has more than one POS in lexicons/vocabularies and estimate their right POS;

**Language specificities:** yes (first 2 objectives);

**Dependencies:** Input Processing l/CVM/ Composed words;

**Database/ Vocabularies needed:** Lexicon, verbs conjugations/forms, pronouns database, choose a database for POS training + Books database (that contains titles/first phrases for each chapter/sub-titles/ dialogues);

**To dos:**

1. Retrieve information by evaluating each verb and send it to CVM (future/present/past, firsts person/second/third, affirmative/negative forms)
2. Bring words that are coming from linguistic evaluation to their common base/stem/lemma.
3. Train database for semantic (some nouns are marked in composed words for this task).
4. For every word make a tuple of the word and his POS.

**Python code:** use existing codes.

* + 1. DER

**Objectives:**

* Making a vector with dates, holydays referrals and “when" adverbs;
* Making a budget vector;
* Making a number/hour vector;
* Making a relatively vector with adverbs (how often, how, how much) and specific adjectives;

**Language specificities:** no;

**Dependencies:** Input processing l/Composed words/CPL/NOG;

**Database/ Vocabularies needed:** DER are imported from Input Processing I & Composed word;

**To dos:**

1. Assess dates, holydays, ‘when’ adverbs and transform them into a custom date that is included in the date vector;
2. Assess numbers & hours and transform them into a custom number included in the number/hour vector (using available slots).
3. Assess budgets and transform them in budget vector (using available slots).
4. Asses “how often, how, how much" adverbs and specific adjectives from Composed words and transform them into a relative number that is kept in the relatively vector.
5. Actualize DER only if new specific data is provided (new or actualized) or we have a complete/trained SPCA; if not DER remains the same for the next turns.

**Python code:** adapted code

* + 1. Splitting sentences

**Objectives:**

* Splitting the input in sentences;
* Identifying consecutive words with the same POS (except verbs) that are separated by space, "or”, “and”, “comma“;

**Language specificities:** yes, for choosing which consecutive POS to use in the main model;

**Dependencies:** Grammar-Semantics/Input Processing II & Embeddings;

**Database/ Vocabularies needed:** specific vocabularies;

**To dos:**

1. Extract important verbs from specific vocabularies by using frequencies.
2. When the following punctuation is identified, separate in different sentences: “.”,”:”,”(“,”!”,”?”.
3. If 2 verbs are consecutive, we keep the one in the database "important verb” or the second one. The other is eliminated.
4. If the POSs are consecutive or separated by space, "or” “and” “comma” we will keep the first word in the string and the others will be sent to Input Processing 2 & Embeddings.
5. The sentence before “but” is eliminated.

**Python code:** existing code; Sentencepiece tokenizers are not effective for the objectives of this pipeline.

* + 1. CVM

**Objectives:**

Populating a 3\*3 matrix with 1 or 0 for the following:

* Question/Request or not, Negation or not, Affirmation or not;
* Future, present or past;
* First, second or third person of the verb.

Each sentence will have a CVM, if the sum of row 1>1 (priority: negation & question), if sum=0 than affirmation=1.

**Language Specificities:** yes (specific forms of negations, retrieving questions even if “?” doesn't exist);

**Dependencies:** Grammar-Semantics/Splitting sentences/CPL

**Database/ Vocabularies needed:** Vocabularies for negations& questions + involve linguists

**To dos:**

1. Retrieve info from Grammar-Semantics and populate the 2nd and 3rd row of the CVM.
2. Search each sentence for key words for negation
3. Searching for wording/punctuation for questions. For every language there are different ways of assessing negations.

**Python code:** Build code from scratch.

* + 1. IVM

**Objectives:**

Populating a matrix with the following information:

* Number of conversational turns, and what the bot is receiving each turn;
* The bot can receive the following: SPCA (5 types of SPCA presented in the specific pipeline), reactions, chitchats or back-up inputs;
* No. of short sentences, no. of complex sentences, no. of inputs from EER & Composed word.

**Dependencies:** EER/Composed word /Splitting sentences;

**To dos:**

Column 1. The number of the turn. Multiple inputs w/t a response from the bot is considered as 1.

Column 2. SPCA1 inputs (0 if non; 1.1,1.2 for the type of SPCA1, see specific pipeline).

Column 3. Reactions (0 for non; 1-9 for the type of reactions - see specific pipeline: Reaction analysis).

Column 4. Chitchats & SPCA3 (0 for non, 1-10 for specific chitchats, 9.1-9.4 the sub-categories for feedback and 10.1-1.7 sub-categories for elaborate),

Column 5. SPCA4 inputs (0 if non; 1,2,3 for the type of SPCA4, see specific pipeline).

Column 6. Short sentences. A sentence with 1 verb is classified as short. Count the number of sentences with 1 verb.

Column 7. Complex sentences. A sentence with more than 1 verb (the verbs that are marked for SPCA multiplication also counts).

Column 8. No. of inputs from EER & Composed words.

Column 9. back-up inputs (0 if non; 1.1,1.2 - see specific pipeline).

**Python code:** Build code from scratch.

* + 1. AVM

**Objectives:**

Populating a matrix with the following information:

* Number of conversational turns, and what the bot is answering each turn;
* Type of bot answers: NOG (NOG 1,2,3, 5 presented in the specific pipeline), secondary answer. “I do not know" answer;
* Type of bot questions/actions: additional, confirmation, review, change topic, disclaimer;
* 3rd flow empathetic interactions and bot queries;
* History of the answers/questions/empathy - not to risk repeating;

**Dependencies:** NOG/IVM

**To dos:**

Column 1. The number of the turns needs to be correlated with IVM.

Column 2. Answers (0 if non: 1,2,3,5 for the type of NOG (+core/deep/back up criteria), 7 for secondary and 8 for “I do not know” )

Column 3. Answer identification number - each answer will have a code.

Column 4. Questions/Actions (0 for non, 1 for additional, 2 for confirmation, 3 for review, 4 for change topic, 5 for disclaimer)

Column 5. Questions/Actions Identification number - each bot action/question will have a code.

Column 6. 3rd flow empathetic interactions and bot queries.

Column 7. 3rd flow identification number and bot queries - not to repeat the empathy.

**Python code:** Build code from scratch.

* 1. Pirkin 2 model. Machine/Deep Learning.
     1. Reply

**Objectives:**

* Evaluating the replies of the users’ utterance in a short sentence;
* The replies can be in a standalone input, or together with other types of inputs (core-inputs/deep conversational…);

**Language specificities:** no;

**Dependencies:** Splitting sentences/ Reaction analysis;

**Database/ Vocabularies needed:** 3 databases with 3 types of replies: confirmations / satisfactions/ understanding, divided equally between positive and negative and with 80-20 training vs testing.

**To dos:**

1. Apply an intent based algorithm on every short sentence to see if the user is giving the bot any replies (3 types).
2. Classify the replies in 6 categories (understanding, misunderstanding, confirmation, negation, satisfied, unsatisfied) and send them to the Reaction analysis pipeline.
3. Eliminating the sentence from Pirkin Model if a valid reply is identified. This sentence (in the case the input has more sentences) will be evaluated for chitchat purposes also.

**Python code:** Use existing codes – data augmentation through translation can be a solution if no datasets available

* + 1. Chitchat

**Objectives:**

* Evaluating the chitchat of the users utterance in short sentences;
* Chitchats can be in a standalone input, or together with other types of inputs (core-inputs/replies...);
* The bot answers will be given in the secondary flow (first displayed answer), and the bot can give a main answer at the same time (second displayed answer);

**Language specificities:** no;

**Dependencies:** CPL/Splitting sentences/Labyrinth model;

**Database/ Vocabularies needed:** databases with chitchats split in 8 categories + feedback (with 4 sub-categories) + elaboration database that is specific for queries (with 7 subcategories)– see Labyrinth model.

**To dos:**

1. Analyze CPL and splitting sentences.
2. Assess CVM of the Chitchat sentence that will be used in CPL for Diatribe.
3. Apply an intent based model on every short sentence to see if the user is chitchatting (some chitchats can lead to deep-conversational through labyrinth model).
4. Updating IVM/AVM depending upon which of the 10 databases was used to give an answer (or track the intent).
5. Eliminating the sentence from Pirkin Model if a valid chitchat is identified.

**Python code:** Use existing codes – data augmentation through translation can be a solution in terms of NLU if no datasets available

* + 1. Untrained NIU

**Objectives:**

* Giving answers to specific interactions that cannot be trained;
* The bot answers will be given in the secondary flow (first displayed answer), and the bot can give a main answer at the same time (second displayed answer);

**Language specificities:** yes;

**Dependencies:** Chitchat/EER/Composed word;

**Database/ Vocabularies needed:** database with specific answers (some random also for expressions);

**To dos:**

1. Receiving inputs from Composed word and Emoji/Abbreviation (these were already categorized).
2. Recategorized composed words that transmit a specific reaction (confirmation/satisfaction/understanding) and send this to reaction analysis.
3. Apply an algorithm, that is based on mapping and not a NLU model, only if no other chitchat valid answer was given. Only one answer is given based on LIFO
4. Updating AVM.
5. Eliminating inputs.

**Python code:** Build code from scratch.

* + 1. Input Processing II

**Objectives:**

* Cleaning up words that were used for other purposes;
* Preparing for Embeddings;

**Dependencies:** Splitting sentences/DER/CVM;

**Database/ Vocabularies needed:** Lexicon, vocabularies for verbs (conjugation forms);

**To dos:**

1. Remove numerical data, punctuation, uppercase and other words used in scope of CVM and DER;
2. Stem/lemma words and/or bring words to their base (for ex: verbs are transformed in their infinitive form), including words that are marked in Splitting sentences

**Python Code:** use existing algorithms

* + 1. Database processing

**Objectives:**

* Creating vocabulary with word frequencies for each database;
* Analyzing word POSs in inputs vs in databases;

**Dependencies:** Input processing I & II / Composed words/ Grammar Semantics;

**Database/ Vocabularies needed:** Database with chitchats/replies + NOG-NLG Labeled datasets + NIU-NLU Specific datasets

**To dos:**

1. Each database is processed in the same way the input is processed, without the creation of entities and matrices.
2. 4 steps in processing and the order: Input processing I, Composed words, Grammar-Semantics, Input processing II.
3. After processing, apply word frequencies (depending on the POS) and create vocabularies for each database.
4. Assign NER1, NER2…+ determine core-domain.

**Python code:** use existing algorithms + algorithms used in the pipelines Input processing I & II / Composed words/ Grammar Semantics.

* + 1. Books processing

**Objectives:**

* Creating vocabulary with word frequencies for each book;
* Transforming the content of the books into a structured database that can be trained;

**Dependencies:** NOG from books/Embeddings/ CVM/ Grammar/ Auto-correct;

**Database/ Vocabularies needed:**  Deep conversational datasets;

**To dos:**

1. All books will be divided in chapters/sub-chapters/paragraphs/sentences, maintaining at the same time their initial place by using a special origin notation for NOG chronological correlations.
2. Apply Machine Education on every sentence (Grammar, NER and CVM).
3. Delete paragraphs with NERs for novels and scripts.
4. CVM matrix will be computed for each sentence. If a paragraph has more sentences, the CVM of the last sentence will be considered the one corresponding to the paragraph.
5. All bold paragraphs, chapters and sub-chapters names will be considered Book Intent Database (BID), together with all paragraphs that are considered questions by CVM matrix.
6. All the CVMs affirmations and negations will be marked for the Book Output Database (BOD) keeping the chronology.
7. All the paragraphs in the BOD will be split in 9 for CVM’s possibilities (row 2 and 3) – persons and tenses.
8. Apply the same processes from Database processing for Book Intent Database.

**Python code:** use existing algorithms + algorithms used in the pipelines Input processing I & II / Composed words/ Grammar Semantics.

* + 1. Auto-Complete & Augmentation-through-conversation

**Objectives:**

* We will apply masking to find implicit words
* Using auto-complete in case a verb, an adjective or a noun is missing, and at the same time IVM/AVM indicates that there are no past intents interactions, in order to find the most probable words in the database, so as to complete the user input with implicit words.

**Dependencies:** IVM/AVM/Input Processing II;

**Database/ Vocabularies needed:** Database with chitchats/replies + NIU-NLU Specific datasets + NOG-NLG Labeled datasets + Deep conversational datasets;

**To dos:**

Auto-Complete if there aren’t any past utterances:

Step 1. Search for UNK words. If found, skip step 2. In case of more UNK words, mark the last one.

Step 2. If we don’t have past inputs from the user, search if we have all these three POSs in the input in this order: verb, noun, adjective. Mark the POS (only one) that doesn't exist.

Step 3. Auto-complete the marked position/UNK word using the limited databases.

\* Attention to the NER (ex: we will not do auto-complete for NER)

NLP masking task if there are past utterances:

1. We will mask the first token of the utterances that do not have a subject/NER to see if we find the implicit Subject. The resulted word will be searched in the past utterances and if found, will become the new subject.
2. We will mask the UNK words in terms of both vocabulary and POS and apply the same training flow.
3. We will mask the last token and apply the same flow. GPT3 can be more effective for this task and then we will apply the same training flow.
4. We will mask the last verb of the core-input. If after masking task we will have the same verb (or similar) we will consider this verb as the Predicate, if not we will also search in the past utterances applying the same training flow from the above tasks

**Python code:** use existing algorithms.

* + 1. Embeddings

**Objectives:**

* Analyzing words that are used to empower other words;
* Finding the most familiar word (for example a synonym to the word from the user that is more known) - we include here also words that are coming from Splitting sentences and composed words;
* Making a word embedding vector to cover all 4 dimensions of the word;

**Language specificities:** yes for words connectors (ex: ''at", “in”) and specific forms of the verbs (ex: “will be", “would”);

**Dependencies:** Splitting sentences/Composed words/ Input processing ll/Database processing/Books processing;

**Database/ Vocabularies needed:** vocabularies for synonyms, regional words, neologisms: vocabulary with connectors;

**To dos:**

1. Assess the syntactic functions of words using connectors (ex : "at”, “in”) + other types (like “will be”) + negations; then eliminate these words.
2. Built a matrix (we will name it k-word) that has on the first column the following:
   * Index no. of the word in the Lexicon/specific vocabularies/additional index for the composed words;
   * Domain identification;
   * NER identification – will be regarded as slot types in training;
   * The POS of the word;
   * Frequency of the word in the databases. NER will not have frequencies.
3. Include in the k-word additional columns: synonyms/ neologisms, regional words and words from splitting sentences.
4. The column with the highest frequency or with a slot type will become the k-word vector of the word.
5. Build the embedding vector of the word with the following data range between -3 and 3: frequency in the databases, frequency of the word in the input (if the words repeat in the same input), position of the word inside the input, position of the word vs the verbs/nouns/adjectives, category of the adjectives (from composed words).

**Python code:** Build code from scratch

* + 1. Triggering/ Common interest

**Objectives:**

* Assessing K-word (see Embeddings) for each word in the SPCA (sentence-intent that was outputted by the previous pipeline that is missing from the presentation[[7]](#footnote-7))
* Initiating clarification discussions with the user if we have incomplete information, or we don’t have a clear sentence-intent;

**Dependencies:** SPCA/ Embeddings/ IVM;

**Database/ Vocabularies needed:**

**To dos:**

Step 1. Verify IVM. Triggering applies only once (mainly for turn 1).

Step 2. See if we have specific k-words for all SPCA. If yes, step 3 is skipped.

Step 3. Address additional questions to the user for details. The question will be specific, by using the word with the highest frequency or the identified NER, or general, if we don't have k-word with high frequency.

**Python code:** Build code from scratch

* + 1. Domain validation

**Objectives:**

Verifying the type of SPCA, confronting also with CVM:

1. SPCA1 - or core (for example: restaurant recommendation or legal consultancy...);

2. SPCA2 – back up for replies (3 types depending on the databases: confirmation, satisfaction, understanding);

3. SPCA3 – back up for chitchats (this can be descriptions of a situations or discussions not intended for a conclusion);

4. SPCA4 – deep conversational discussion on a topic through book queries (without a back up model);

**Dependencies:** SPCA/ CVM/ Embeddings/ Reply/ Chitchat;

**Database/ Vocabularies needed:** all

**To dos:**

Step 1. Verifying all k-words of the sentence, if the user is talking about core domains and which one (restaurant, legal, investment...multi-domain).

Step 2. If core domain, confronting with CVM and checking the results of pipelines Reply and Chitchat.

Step 3. If SPCA1, check if P=“form verb/reservation” or S=“NER1”. If yes, classified as SPCA1.1 and SPCA1.2, respectively.

Step 3. If not core domain, check k-words and CVM to classify as SPCA2, SPCA3 or SPCA4.

Step 4. SPCA2 and SPCA3 will be also classified in SPCA2.1-2.3 and SPCA3.1-3.10 depending on the specific database used. 3.9 has 4 additional subtypes, 3.10 has 7 additional subtypes. If a secondary flow reply is active, SPCA2 will be eliminated. The same with SPCA3 and chitchats - the secondary flow has priority.

Step 5. SPCA4 will have 3 correspondences in NOG: novels/screenplays or specialized books.

Step 6. If no domain is identified, then we will consider it SPCA0 and w/t a SPCA memory update the bot will not have an answer.

**Python code:** Build code from scratch

* + 1. SPCA- memory update

**Objectives:**

* Making an update between present SPCA and past one that went through Reset pipeline from CPL layer;
* Making rules between present SPCA that can be SPCA1/SPCA3/SPCA4 and past ones with the same possibilities. SPCA2 is not in scope of memory update (it's treated in CPL);
* Implementing a summarization-augmentation algorithm for the E2E model – back-up model

**Dependencies:** NOG/Reset/Domain validation/ IVM/AVM/ CVM;

**Database/ Vocabularies needed:**

**To dos:**

Step 1. Assess if we have past SPCA and its type- Verify Reset pipeline & IVM & AVM. If not, SPCA final = SPCA & stop other steps.

Step 2. Verify if the present SPCA has the same type as the past SPCA. If not, skip Step 3.

Step 3. Prioritize the present SPCA. The Complement and Attribute can only be updated together (Or NER and Attribute). In case of no predicate in the current SPCA, the CVM final=CVM past (-1). In case of no subject in present SPCA, the final S = past S. The same with the Complement and the Attribute. Exceptions for specific additional questions. After update, the Domain validation pipeline will be repeated.

Step 4. If the present SPCA and past SPCA are of different types, no update will be made and final SPCA= present SPCA.

Step 5. If the present SPCA is SPCA0 and we have past SPCAs, the same rules from Step3 apply.

Step 6. If back-up model is active, we will implement a summarization-augmentation algorithm for long-term memory update (XU et al.,2021).

Python code: Build code from scratch

* + 1. Main flow/Back up

**Objectives:**

* Evaluating the input of the user utterance in a E2E model;
* Using this model as a backup in case the user has negative sentiments about the Pirkin model answers;
* Using also this model if the bot automatically determines that it is more efficient;

**Dependencies:** NOG/AVM/Auto-training/Splitting sentences/Memory update;

**Database/ Vocabularies needed:** NIU-NLU Specific datasets, NOG-NLG Labeled datasets, Deep conversational datasets + corpuses from pre-trained models (if the case)

**To Dos:**

1. Verify if there is a request for a back up answer.
2. Apply one of the 3 models (Encoders-decoders model for chatbot task (aka. T5), Open domain dialogue model with summarization memory-augmentation (aka. PARL.AI), Open-Source Language Understanding with Dialogue Management (aka. RASA)) on the user input (the same input used also for SPCA, and the 3 models will be implemented in the order that were presented).
3. Determine what kind of backup is used depending on the database (back-up 1 for core-domain, back up 4 for deep conversational through book queries).
4. Back up 1 will have 3 subdomains (Back up 1.1 for forms; for example: reservation, back up 1.2 if it is referring to NER1, and back-up 1 for the rest)
5. Updating IVM/AVM and go through the memory update pipeline for summarization-augmentation.

**Python code:** Use existing codes

1. Implementation of Conversational Policy Learning Pirkin 3 model.
   1. Reaction analysis

**Objectives:**

* Identifying if the user is having a reaction following the bot actions (by training/labeling the databases in 2 ways: using a DL model and Pirkin model (SPCA2) similar with chitchats – see the processes in NOG layer);
* If more than one reaction is coming from the current input then the priority is: Ironies, Reply, SPCA2, Emoji, Untrained;
* Actualizing IVM;
* Transforming these reactions into bot actions in the pipeline: Debate or bot actions from reactions.

|  |  |  |  |
| --- | --- | --- | --- |
| **NIU** | **CVM** | **Positive/Negative reaction** | **Classification/IVM** |
| ironies | n/a | negative | 8.Sarcastic |
| sad face | n/a | negative | 6.Sad |
| Smiley face | n/a | positive | 2. Happy |
| SPCA2/Reply/Untrained - confirmation | affirmation | positive | 3.Confirmation |
| SPCA2/Reply/ Untrained- confirmation | negation | negative | 5.Negation |
| SPCA2/Reply/Untrained - satisfaction | affirmation | positive | 1. Satisfied |
| SPCA2/Reply/Untrained - satisfaction | negation | negative | 9. Unsatisfied |
| SPCA2/Reply/ Untrained- understanding | affirmation | positive | 4. Understanding |
| SPCA2/Reply/ Untrained- understanding | negation | negative | 7. Misunderstanding |

* 1. Reset

**Objectives:**

* Resetting previous SPCA depending on AVM and current user reaction;
* Appending/resetting chitchats depending on AVM and current user reaction;
* Appending/resetting back-up answer depending on AVM and current user reaction;
* Resetting DER;
* Freeze means that no SPCA of inputs are updated until the end of the query/action.
* Append means that new input=past input (summarized) + present input = (memory update/summarization-augmentation)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **NOG/NIU (-1)** | **Pirkin**  **model** | **Pirkin +**  **negative** | **Chitchat**  **only** | **Chitchat +**  **negative** | **Back up** | **Backup +**  **negative** | **DER** |
| **NOG 1** | Reset PCA | Reset SPCA | n/a | n/a | reset | append | No reset |
| **NOG 2** | Reset PCA | Reset SPCA | n/a | n/a | reset | append | No reset |
| **NOG 3** | Reset SPCA | Reset SPCA | reset | append | n/a | n/a | reset |
| **NOG 4** | Reset SPCA | Reset P | n/a | n/a | n/a | n/a | reset |
| **NOG 5** | No reset | Reset CA | n/a | n/a | n/a | n/a | No reset |
| **NOG 6** | Freeze SPCA | Freeze SPCA | n/a | n/a | n/a | n/a | No reset |
| **Additional question** | No reset | No reset | append | n/a | append | n/a | No reset |
| **Confirmation question** | No reset | Reset PCA | n/a | n/a | n/a | n/a | No reset |
| **Change topic** | Reset SPCA | No reset | reset | Append | reset | append | reset |
| **Review request** | No reset | Reset PCA | n/a | n/a | append | reset | No reset |
| **“I do not know”** | Reset PCA | Reset PCA | reset | Append | append | append | No reset |
| **Disclaimer/ no action** | Freeze SPCA | No Reset | n/a | n/a | append | reset | No reset |

* 1. States

**Objectives:**

* Assessing the current state of the conversation;
* Additional/Confirmation questions are not influencing the states;
* In the current NIU we have main flow and secondary flow, the state is influenced only by the main flow, but Dialogues & Discourses will initiate a secondary flow (Diatribe) if the case;
* Queries states (as past states) will be treated differently in pipelines because of their specificities.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Current NIU/past states** | **N/A** | **Discourse** | **Dialogue** | **Diatribe** | **Debate** |
| **SPCA 0/back up 0** | IDNK (+Diatribe) | Discourse 0 (+Diatribe) | Dialogue 0 (+Diatribe) | Diatribe | Debate (+Diatribe) |
| **DER – relatively vector =10-90%** | Bot queries 2 | Bot queries 2 | Bot queries 2 | Bot queries 2 | Bot queries 2 |
| **SPCA1.0 short / back-up 1** | Dialogue 1(+Diatribe) | Dialogue 2 (+Diatribe) | Dialogue 2 (+Diatribe) | Dialogue 1 | Dialogue 1 (+Diatribe) |
| **SPCA 1.0 - complex** | Discourse 1 (+Diatribe) | Discourse 2 (+Diatribe) | Discourse 2 (+Diatribe) | Discourse 1 | Discourse 1 (+Diatribe) |
| **SPCA 1.1/back up 1.1 or DER (dates, number, hour)** | Bot Queries 1 | Bot Queries 1 | Bot Queries 1 | Bot Queries 1 | Bot Queries 1 |
| **SPCA 1.2/back up 1.2** | Dialogue 1 | User queries 1 | User queries 1 | Dialogue 1 | User queries 1 |
| **Only Reaction(SPCA2/Reply)** | Debate (+Diatribe) | Debate (+Diatribe) | Debate (+Diatribe) | Debate | Debate (+Diatribe) |
| **SPCA 3/Chitchat (w/t feedback/elaborate)** | Diatribe | Diatribe | Dialogue | Change topic 1 | Diatribe |
| **Chitchat (feedback/elaborate)** | Diatribe | User queries 1/2 | User queries ½ | Diatribe | Diatribe |
| **SPCA 4 short** | Dialogue 1 (+Diatribe) | User queries 2 | User queries 2 | Dialogue 1 | Dialogue 2 (+Diatribe) |
| **SPCA 4 complex** | Discourse 1 (+Diatribe) | User queries 2 | User queries 2 | Discourse 1 | Discourse 2 (+Diatribe) |

* 1. Discourse

**Objectives:**

* Applying policies depending on the evaluation of CVM and the type of the state;
* If more policies apply at the same time, the priority is: answer/review/confirmation/change topic/Disclaimer/”l do not know”
* Check if the state is accurate. If not, reclassify the state Review 1 purpose is to memorize current input.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CVM/Type of Discourse** | **CVM 3rd row** | **Discourse 0** | | | **Discourse 1** | | | **Discourse 2** | | |
| **CVM-first row** | **person** | **Affirmative** | **negative** | **question** | **affirmative** | **negative** | **question** | **affirmative** | **negative** | **question** |
| **CVM past** | i | Additional 3 | no action | IDNK | review1 | Debate 1 | IDNK | confirmation | Debate 1 | No action |
| ii | Additional 3 | Debate 1 | Additional 3 | confirmation | Debate 1 | NOG1 | NOG1 | Debate 1 | NOG1 |
| iii | no action | no action | IDNK | confirmation | Review1 | confirmation | review | Debate 1 | NOG1 |
| **CVM- present** | i | Additional 3 | Debate 2 | IDNK | review1 | Debate 2 | review1 | NOG1 | review1 | confirmation |
| ii | Additional 3 | Debate 2 | dialogue 0 | NOG1 | Debate 2 | NOG1 | NOG1 | Debate 2 | NOG1 |
| iii | Additional 3 | debate 2 | IDNK | confirmation | No action | confirmation | NOG1 | Debate 2 | NOG1 |
| **CVM- future** | i | Additional 3 | debate 3 | IDNK | confirmation | review1 | NOG1 | NOG1 | review1 | NOG1 |
| ii | Additional 3 | Debate 3 | Additional 3 | NOG1 | Debate 3 | NOG1 | NOG1 | Debate 3 | NOG1 |
| iii | Additional 3 | Debate 3 | Additional 3 | confirmation | Debate 3 | confirmation | NOG1 | Debate 3 | NOG1 |

* 1. Dialogue

**Objectives:**

* Applying policies depending on the evaluation of CVM and the type of the state;
* If more policies apply at the same time, the priority is: answer/review/confirmation/change topic/Disclaimer/ “l do not know”
  + - Check if the state is accurate. If not, reclassify the state
    - Review 2 is for asking the future reviews

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CVM/Type of Dialogue** | **CVM 3rd row** | **Discourse 0** | | | **Discourse 1** | | | **Discourse 2** | | |
| **CVM-first row** | **person** | **affirmative** | **negative** | **question** | **affirmative** | **negative** | **question** | **affirmative** | **negative** | **question** |
| **CVM - past** | I | IDNK | no action | IDNK | review2 | debate 1 | NOG2 | review2 | Debate 1 | NOG2 |
| II | IDNK | Debate 1 | NOG2 | NOG2 | Debate 1 | NOG2 | NOG2 | Debate 1 | NOG2 |
| III | no action | no action | IDNK | NOG2 | review2 | NOG2 | review2 | Debate 1 | NOG2 |
| **CVM- present** | I | IDNK | Debate 2 | IDNK | review2 | Debate 2 | NOG2 | NOG2 | review2 | NOG2 |
| II | IDNK | Debate 2 | NOG2 | NOG2 | Debate 2 | NOG2 | NOG2 | Debate 2 | NOG2 |
| III | IDNK | debate 2 | IDNK | NOG2 | No action | NOG2 | NOG2 | Debate 2 | NOG2 |
| **CVM- future** | I | IDNK | Debate 3 | IDNK | NOG2 | review2 | NOG2 | NOG2 | review2 | NOG2 |
| II | IDNK | Debate 3 | NOG2 | NOG2 | Debate 3 | NOG2 | NOG2 | Debate 3 | NOG2 |
| III | IDNK | Debate 3 | No action | NOG2 | Debate 3 | NOG2 | NOG2 | Debate 3 | NOG2 |

* 1. Diatribe

**Objectives:**

* Re-assessing the intention of the user to make sure we had an accurate understanding

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Type of Diatribe/ CVM 2nd row** | **CVM past** | | | **CVM present** | | | **CVM future** | | |
| **CVM - first row** | **affirmative** | **negative** | **question** | **affirmative** | **negative** | **question** | **affirmative** | **negative** | **question** |
| **What are you doing?** | No action | Debate 1 | NOG3 | No action | Debate 2 | NOG3 | No action | Debate 3 | NOG3 |
| **Who are you** | NOG3 | Debate 1 | NOG3 | NOG3 | Debate 2 | NOG3 | NOG3 | Debate 3 | NOG3 |
| **Personality** | NOG3 | Debate 1 | NOG3 | NOG3 | Debate 2 | NOG3 | NOG3 | Debate 3 | NOG3 |
| **Greetings** | NOG3 | | | | | | | | |
| **BFF** | NOG3 | Debate 1 | Debate 2 | NOG3 | Debate 2 | NOG3 | NOG3 | Debate 3 | NOG3 |
| **Specific info's** | No action | Debate 1 | NOG3 | No action | Debate 2 | NOG3 | NOG3 | Debate 3 | NOG3 |
| **General info's** | No action | Debate 1 | NOG3 | No action | Debate 2 | NOG3 | NOG3 | Debate 3 | NOG3 |
| **Redirect** | Debate 3 | Debate 1 | Review 2 | Review 2 | Debate 2 | Review 2 | Review 2 | Debate 3 | Review 2 |
| **Feedback (4 types)** | NOG3 + NOG5 (if user queries is active) | | | | | | | | |
| **Elaborate (7 types)** | NOG3+NOG5 | Debate 1 | No action | NOG3+NOG5 | Debate 2 | NOG3+NOG5 | No action | Debate 3 | NOG3+NOG5 |

* 1. Debate from other states

**Objectives:**

* Choosing the right policies when debate is initiated in the following states due to CVM assessment:
  + - * Discourse;
      * Dialogue;
      * Diatribe.

|  |  |
| --- | --- |
| **Debates initiated** | **Policies – 3rd flow – NOG 4** |
| Debate 1 (CVM - past) | Responding to bad news |
| Debate 2 (CVM - present) | Showing surprise |
| Debate 3 (CVM - future) | Getting the user to say more |

* 1. Debate or bot actions due to reactions

**Objectives:**

* If an input with reaction has also a secondary flow, the secondary flow will be addressed together with 3rd flow;
* Applying empathic actions/policies (3rd flow, with red) depending on the reactions of the user and the type of the state;
* If Backup answer (see conversational analysis, in some instances the back-up answer can be the answer from SPCA/Pirkin model and not from the back-up model) was already provided or doesn’t exist, then policies from the Secondary flow/NOG3 applies;
* If the current input has also intents, then the 3rd flow will not be initiated.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **User reactions/**  **bot action** | **Additional questions** | **Confirmation questions** | **Review/ Opinion 1** | **Review/ opinion 2** | **Disclaimers w/t IDNK** | **IDNK** | **Change topic** | **3rd flow/**  **bot queries** | **NOG 1/2/5** | **NOG 3** |
| **No reactions** | NOG1 | NOG1 | No action | Ch.Topic 2 | No action | Ch.Topic2 | No action | No action | Ch.topic2 | CH.Topic1 |
| **Understanding** | No action | NOG1 | Sp.policy1 | Sp.policy2 | No action | Backup NOG | Own story | Say more | Agreement | Good news |
| **Confirmation** | No action | NOG1 | Sp.policy1 | Sp.policy2 | No action | Backup NOG | Say more | Say more | Good news | Agreement |
| **Satisfied** | NOG1 | NOG1 | No action | Sp.policy2 | No action | Backup NOG | Good news | Say more | Agreement | Good news |
| **Happy** | NOG1 | NOG1 | No action | No action | No action | Backup NOG | Agreement | Say more | Good news | Good news |
| **Misunderstanding** | NOG1 | Own story | Own story | Own story | Surprised | Backup NOG | Own story | No action | Backup | Bad news |
| **Negation** | NOG1 | Backup NOG1 | Ch.Topic 2 | Ch.Topic 2 | Surprised | Backup NOG | Bad News | No action | Backup | Say more |
| **Unsatisfied** | Say more | Backup NOG1 | Ch.topic 2 | Bad news | Surprised | Backup NOG | Say more | No action | Backup | Own Story |
| **Sarcastic** | surprised | Backup NOG1 | Ch.topic 2 | Say more | Surprised | Backup NOG | Surprise | No action | Backup | Surprise |
| **Sad** | NOG1 | Backup NOG1 | Ch.topic2 | Say more | Surprised | Backup NOG | Say more | No action | Backup | Say more |

* 1. User queries

**Objectives:**

* Which policies apply depending on the current NIU?
* In which conditions the user queries state is initiating other states?
* If DER- relatively vector is between 10 and 90% then Bot queries 2 is initiated.

|  |  |  |
| --- | --- | --- |
| **Current NIU/past states** | **User queries 1/**  **initiation from other states** | **User queries 2/**  **Initiation from other states** |
| **SPCA 0/back up 0** | Additional question 4 | Additional question 5 |
| **SPCA1.0 short / back-up 1** | NOG 5 | Dialogue 1 |
| **SPCA 1.0 - complex** | NOG 5 | Dialogue 1 |
| **SPCA 1.1/back up 1.1** | Bot queries 1 | Bot queries 1 |
| **SPCA 1.2/back up 1.2** | NOG5 | Dialogue 1 |
| **Reaction(SPCA2/Reply) w/t the ones caused by bot questions** | Exit situations | Exit situations |
| **SPCA 3/**  **Chitchat (w/t feedback & elaborate)/ Untrained** | Change topic 1 | Change topic 3 |
| **SPCA 4** | Change topic 1 | NOG 5 (Labyrinth model) |
| **Chitchat (feedback + elaborate)** | N/A | NOG 5 (Labyrinth model) |

|  |  |  |
| --- | --- | --- |
| **Exit situations** | **User queries 1** | **User queries 2** |
| **Misunderstanding** | Exit/debate | Exit/debate |
| **Negation** | Exit/debate | No exit |
| **Unsatisfied** | Dialogue | Dialogue |
| **Sarcastic** | Exit/debate | No exit |
| **Sad** | Exit/debate | No exit |
| **Understanding** | No exit | No exit |
| **Confirmation** | No exit | No exit |
| **Satisfied** | Exit/Debate | No exit |
| **Happy** | No exit | No exit |

* 1. Labyrinth model (AGI)

|  |  |  |
| --- | --- | --- |
| **Situations inside user queries** | **Identification of the situation** | **NOG/Solution** |
| User doubts | Chitchat- feedback/+ current SPCA4 | NOG5 next + disclaimer 1 + negative feedback |
| User interest in writer opinion | Chitchat- feedback/ + current SPCA4 | NOG5 next + positive feedback |
| User interest in the writer | Chitchat- feedback/ + current SPCA4 | NOG5- writer presentation + positive feedback |
| The user is sharing his thoughts | Chitchat- feedback/ + current SPCA4 | NOG5 next + positive feedback |
| Bot intuition | DER – relatively vector = [10%-90%] | NOG6 - Bot queries 2 |
| Expositions | Chitchat elaborate | NOG5- beginning of the chapter reproduction |
| Plot | Chitchat elaborate | NOG5 – next phrase/section after beginning |
| Core action | Chitchat elaborate | NOG5 – the next section after the plot |
| Highlight | Chitchat elaborate | NOG5 – summarization – SPCA |
| Outcome | Chitchat elaborate | NOG5– last section of the chapter |
| Go back to the main topic | Chitchat elaborate | NOG5 – book introduction |
| Different approaches | Chitchat elaborate | NOG5- second answer as frequency |
| Change in topic | SPCA4 - new | Disclaimer 2 for changing chapter + NOG5 begin. |
| Bot initiatives | Positive reaction or no possible answers | Disclaimer 3 – promoting the book |

* 1. Bot queries (Prompts - AGI)

**Objectives:**

* + When to exit the bot queries?
  + What policies to apply in different circumstances?
  + Things are more straightforward for the Bot queries 1 and we don’t need supplementary analyses.
  + The initial intent is addressed only after DER becomes more than 90%.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Policy/DER relatively vector** | **DER<10%** | **DER=[10-29%]** | **DER =[30-49%]** | **DER =[50-69%]** | **DER =[70-89%]** | **DER>=90%** |
| Intuition policies | No policy | Share his thoughts | Interest in the user opinion | Interest in the user person | Doubts | No policy |

**Exit Bot queries 2**:

* DER relatively vector<10% or >90%;
* No DER relatively vector in the last 2 turns;
* New inputs: SPCA1;
* Repetitive types of SPCA’s or back up’s (ex: his asking 2 times for redirect);
* Reactions: sarcastic, misunderstanding, unsatisfied;

**Exit Bot queries 1:**

* All cases except SPCA1.2/1.1 and DER actualization,
  1. Special policies

**Objectives:**

* + Applying no action policy in defined cases, initiated either by other states, either by this state;
  + Applying a policy for instances when the user is not giving any reaction to the bot actions (answers/questions…);
  + Capturing reviews/opinions offered actively by the user (after positive reaction from the user);
  + Capturing reviews/opinions solicited by the bot (conditioned by the offering or the positive reaction)

|  |  |  |  |
| --- | --- | --- | --- |
| **Policies** | **States originated** | **NIU implications** | **To dos** |
| No action | All States | Freeze NIU | The bot will not have any actions |
| No reaction | NOG - all | Freeze NIU | Wait 1 minute and if no new input Debate assessment |
| Special policy 1 | Debate from reactions | Reset existing NIU | Capture past input and then Change topic 5 |
| Special policy 2 | Debate from reactions | Reset next inputted NIU | Capture new input and then Change topic 5 |

**Exit Review/Opinion policy 1**:

* After the user gave the acceptance the past review/opinion is captured;

**Exit Review/Opinion policy 2**:

* After the user gives the acceptance, the utterance from the same input as the reaction, or the following utterance will be considered a review/opinion and will be captured. The administrator can afterwards choose not to accept reviews/opinions that are not complaining with some rules. Different NLP solutions can be implemented on the review corpus, but this is not in scope of this project;

**Exit, no action or no reaction:**

* Any input by the user.

1. Implementation of Natural Output Generation models

* 1. Training and additional models
     1. Database training

**Objectives:**

* Training the database for Pirkin Model;
* Training the database for Chitchats, Replies and back up answer;

**Dependencies:** NOG from databases/NIU;

**Database/ Vocabularies/External needed:** all,

**To dos:**

**Pirkin model**

* The database with questions will go through NIU pipeline (the same as an input) except: CVM, IVM, AVM, Backup, Secondary, Books processing, Auto Complete, Triggering, Domain Validation, SPCA memory;
* Each resulting SPCA (that will have k-vector’s and an embedding vector’s) will be linked with the answer from the databases and chronological indices will be applied.

**E2E – DL model**

* It will be a standard training included into an E2E model. Also the answers will be marked not to duplicate. See also chapter 4

**Python code:** see NIU and Chitchat proposed algorithms.

* + 1. Books training

**Objectives:**

* Training books inputs;
* Using Books processing output;
* Making summarization using SPCA for all books output database;

**Dependencies:** NOG from books/NIU/ Books processing;

**Database/ Vocabularies/External needed:** Books Input Database, Books Output Database;

**To dos:**

**Inputs:**

* The Book Input Database will go through NIU – Machine Learning pipelines except: Backup, Secondary, Books processing, Auto Complete, Triggering, Domain Validation, SPCA memory – keep in mind that in Book processing we’ve also used Machine Education;
* The resulting SPCAs will be linked with immediate results from the Book Output Database, to establish the chapter/subchapter that will be queried. These results will be linked with the Labyrinth model and the principles from the deep conversational answers.
* The chapter will be blocked in the user queries states in order not to exist without considering the exit terms stipulated in the user queries pipeline.

**NOG5 Summarization**

* The Book Outputs Database will go through the same process as Database training for Pirkin Model.
* The resulted SPCA will be kept for Labyrinth model – NOG5 Summarization; the immediate answer will apply,

**Python code:** see NIU

* + 1. NLG enhancements (AGI)

**Objectives:**

* Determining if the speech of the user can be regarded as formal or informal;
* Calculating a scoring to determine the face reaction of the bot (avatar);

\* More enhancements will be presented in the next pipeline

**Dependencies:** Chitchats/Untrained NIU/Domain validation/ Reaction analysis/EER/AVM

**Database/ Vocabularies/External needed:**

**To dos:**

1. Each category of chitchat/SPCA3 (10) and of reaction (9) and Untrained (emoji/metaphors/abbreviation/expression/quotes) are allocated a scoring for informal/formal speech and face reaction between 0 and 1.
2. If scoring for informal/formal becomes more than 1 all the NOG’s for this user will be informal.
3. Displaying the avatar after calculating the scoring. We will have 6 avatars, consistent with EER categorization (smile, sad, laughing, cool, kiss and blink). If no other SPCA3/secondary flow inputs occur, the former avatar is still displayed.
4. Include in AVM the scoring for informal/formal and avatar in Column 8 and 9.

**Python code:** Build code from scratch

* + 1. Self-generative model (AGI)

**Objectives:**

* Assessing if the user is qualifying for receiving self-generative responses/reactions;
* Determining possible external impact on the bot reactions/responses/mood;
* Assessing how the bot behavior can change over time. The bot answers can be different in time and will be adapted to external factors and his changing behavior;

**Dependencies:** Bot queries/NLG enhancements/Domain classification/NOG Answers/ Reaction analysis/Auto-training/AVM;

**Database/ Vocabularies/External needed**: sites with news/weather/sports + 1 book per month (grounding principles applies);

**To dos:**

1. Assess criteria for Self-generative model (SPCA4+feedback+informal);
2. Choose the sites that the bot will be assessing daily (news/weather/sports) + books to include in database monthly;
3. Assess the daily news and monthly book by using reaction analysis. Also the weather/sports results will have negative or positive impact on the bot behavior/mood. All this will be quantified and will determine pessimist/optimist behavior. If the result is neutral, then the feedback will impact.
4. The pessimistic/optimistic indicator will be sent to NOG from books and will impact the answers (the core intents/ answers will not be impacted).
5. A bot behavioral changes indicator will be created by dividing the number of daily optimistic behaviors to the pessimistic ones. This percentage will also be sent to NOG and will have an impact on the answers (using disclaimers to show trust or doubts referring to his own answers).
6. Include in column 10 and 11 of the AVM the indicators for behavioral changes and pessimistic/optimistic.

**Python code:** Build code from scratch

* 1. Pirkin 4 model implementation. NOG from databases
     1. Chitchat + Untrained answer

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **NOG 3 initiated** | **States** | **E2E Secondary** | **SPCA3 mapping rules** | **Duplicate answer** | **NOG/Solution** |
| 1. | yes | Diatribe | yes | Yes/no | no | The bot will give the answer identified by Secondary flow E2E and update AVM. No core answers. |
| 2. | yes | Dialogue/ Discourse (+Diatribe) | yes | Yes/no | no | The bot will give the answer identified by Secondary flow E2E and update AVM. Also a core answer is pending from SPCA answer or back up answer. |
| 3. | no | all | Yes/no | Yes/no | Yes/no | No NOG for chitchat/ untrained |
| 4. | yes | Diatribe/ Discourse/ Dialogue | no | Yes (2 or 3 matches) | no | NOG for SPCA3 + Confidence Disclaimer |
| 5. | yes | Diatribe/ Discourse/ Dialogue | no | Yes (4 matches) | no | NOG for SPCA3 + w/t Confidence Disclaimer |
| 6. | yes | Diatribe/ Discourse/ Dialogue | yes | Yes (at least 2 matches) | yes | NOG for SPCA3 + Confidence Disclaimer if needed |
| 7. | yes | Diatribe/ Discourse/ Dialogue | no | Yes (3 or 4 matches) | yes | Search for NOG/ SPCA3 with less matches and different answer + Confidence disclaimer |
| 8. | yes | Diatribe/ Discourse/ Dialogue | yes | no | yes | No NOG for chitchat/ untrained |

1. In order for SPCA Rules to apply, we need minimum 2 matching at least for SPCA (both in terms of k-vector and embedding vector).
2. In order for an embedding vector to be considered a match, we will include a deviation of max. 1.5 points for each column with values between -3 and 3.
3. If NOG is a solution, then NLG enhancements will be assessed in order to establish if the answer will be formal or informal, or an avatar change is necessary.
4. If SPCA has more than one intent associated with an answer, then the priority is the number of matches or the Subject over Predicate over Complement.
5. AVM is being actualized.
   * 1. SPCA core answer (database)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **NOG 1 / 2 initiated** | **NER** | **DER** | **SPCA mapping rules** | **Duplicate answer** | **NOG/Solution** |
| 1. | yes | no | no | Yes (2-4 matches) | Yes/no | Deep conversational is activated |
| 2. | yes | no | budget | Yes (1-4 matches) | Yes/no | NOG from SPCA1 plus confidence disclaimer if needed |
| 3. | yes | no | no | no | yes/no | IDNK (I do not know disclaimer) |
| 4. | yes | core | no | Yes (2-4 matches) | no | NOG from SPCA1 plus confidence disclaimer if needed |
| 5. | yes | core | no | Yes (2-4 matches) | yes | Search for NOG/ SPCA1 with lesser matches/ different answer |
| 6. | yes | core | budget | Yes (1-4 matches) | Yes/no | User queries 1 initiated |
| 7. | yes | more cores\* | Yes/no | Yes (2-4 matches) | no | The prioritization of NER’s + NOG from SPCA1 |
| 8. | no | - | - | - | - | IDNK |

1. In order for SPCA Rules to apply, we need minimum 2 matching, at least for SPCA (both in terms of k-vector and embedding vector).
2. In order for an embedding vector to be considered a match, we will include a deviation of max. 1.5 points for each column with values between -3 and 3.
3. If NOG is a solution, then NLG enhancements will be assessed in order to establish if the answer will be formal or informal, or an avatar changing is necessary
4. If SPCA has more than one intent associated with an answer, then the priority is the number of matches or the Subject over Predicate over Complement.
5. At this stage, Dialogue and Discourse states will be treated the same (we are after the additional/confirmation questions possible phase).
6. IDNK- “I don’t know” reply – at least 20 types of answers that will be provided random and no duplicates.
7. SPCA contains the 5th category of adverbs (where) – an algorithm that matches the current position of the user (or the desired location) with the GPS locations defined in NER will be activated in order to find solutions (it is implemented in bucurieesti.ro bot).

\*The bot is multi-domain, and the user is using core-NER for more domains,

* + 1. Back up answer/E2E DL model (database)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **NOG 1 / 2 initiated** | **NER/DER dates,no** | **Type of backup** | **Training scale/efficiency** | **Duplicate answer** | **NOG/Solution** |
| 1. | yes | no | 1.0 | More than 80% | no | NOG generated by E2E model + AVM/NLG enhancement update |
| 2. | yes | no | 1.0 | More than 80% | Yes | Pirkin model initiation |
| 3. | yes | no | 1.0 | Less than 80% | yes/no | IDNK or Pirkin model initiation |
| 4. | yes | Main NER | 1.2 | More than 80% | No/yes | NOG generated by E2E model + AVM/NLG enhancement Update + user queries 1 state initiation |
| 5. | yes | Main NER | 1.2 | less than 80% | No/yes | IDNK or Pirkin model initiation + user queries 1 |
| 6. | yes | DER | 1.1 | 100% | No/yes | back-up answer & task oriented completion |
| 7. | yes | DER | 1.1 | Less than 100% | No/yes | Bot queries 1, no back up answer |
| 8. | yes | core | 1.0 | More than 80% | Yes/no | NOG generated by E2E model + AVM/NLG enhancement Update |
| 9. | yes | more cores\* | - | - | - | This solutions will probably not work for multi-domains/ Pirkin initialization |
| 10. | no | - | - | - | - | IDNK |

1. If NOG is a solution, then NLG enhancements will be assessed in order to establish if the answer will be formal or informal, or an avatar change is necessary.
2. At this stage, Dialogue and Discourse states will be treated the same (we are after the additional/confirmation questions possible phase).

\*the bot is multi-domain and the user is using core-NER for more domains,

* + 1. Conversational analysis (AGI)

**Objectives:**

* Determining if the Pirkin-SPCA model is better suited (as a principal model trained) for the bot configuration or one of the back-up models.
* Calculating a scoring for each interaction that the bot is having and a consolidated scoring that can show the results of the bot.

**Dependencies:** CPL/Auto-training/Back up model/Reactions/IVM/AVM

**To Dos:**

1. Each bot will have a Pirkin model as the initial, custom model for the core-intents.
2. Calculating a scoring for each interaction. Two indicators will be evaluated: 1. no. of turns 2. User reactions.
3. Calculating the average of the scoring for interactions only with a custom model.
4. Calculating the average of scoring for interaction where back-up answers were given.
5. At each x interactions compare 3 with 4.
6. If results from point 4 are better than 3, the backup model will become a custom model.
7. Repeat the process.
   * 1. Questions

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Type of questions | When are they  used? | Database | Parameter | Meaning of the questions | Bot actions |
| Additional 1 | Triggering | 5 random | NER or  highest frequency | Specific additional questions related to the parameter | Debate assessment |
| Additional 2 | Triggering | 5 random | no | General additional question to provide more details | Debate assessment |
| Additional 3 | Discourse state policies/ User queries state | 10 random | no | Clarification of the intent | Debate assessment |
| Confirmation | Discourse state policies | 10 random | no | Asking the user if he is interested in the bot view on the matter | Debate assessment |
| Review/Opinion 1 | Discourse state policies | 5 random | no | Asking the user if he can capture the past user discourse as an opinion/review | Debate/Special policies |
| Review/Opinion 2 | Dialogue state policies/ Diatribe | 5 random | no | Asking the user if he wants to give as a review, informing the user of GDPR implications if he chooses to sign the review/opinion | Debate/Special policies |

Attention not to duplicate Questions – Check AVM before answering

* + 1. Change Topic

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Type of change topic** | **When is it used?** | **Database** | **Parameter** | **Meaning of the topic change** | **Bot actions** |
| Change Topic 1 | Consecutive Diatribe states | 10 random | no | The user needs to understand that this is not a chitchat and redirect to core-intents. | Debate assessment |
| Change topic 2 | Debate- Review | 5 random | no | The user doesn’t want to give reviews. Redirect to proposed core-intents. | Debate assessment |
| Change Topic 3 | User queries 2 | 5 random | no | Redirect to existential topics. | Debate assessment |
| Change Topic 4 | Debate/NOG 4 – 3rd flow | 5 random | no | The debate risks becoming boring. | Debate assessment |
| Change Topic 5 | Special policies | 5 random | no | Thank the user for the review/opinion and then initiate a new discussion. | Debate assessment |

Attention not to duplicate – Check AVM before answering

* + 1. Disclaimers

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Type of disclaimers** | **When are they used?** | **Database** | **Parameter** | **Meaning of the disclaimers** | **Bot actions** |
| Disclaimer 1 | Labyrinth – User doubts | 5 random | no | Explaining that we will follow the book chronology | no |
| Disclaimer 2 | Labyrinth – change chapter | 5 random | no | Let the user understand that the bot is changing the chapter of the book | no |
| Disclaimer 3 | Labyrinth – promoting books | 5 random | no | Arguing that it is better to read the entire book for a better understanding | no |
| Confidence disclaimer 50% | Matching SPCA | 5 random | no | The following answer is with 50% confidence | no |
| Confidence disclaimer 75% | Matching SPCA | 5 random | no | The following answer is with 75% confidence | no |
| Disclaimer - doubts | Self Generative model | 5 random | no | The bot has a bad period | no |
| Disclaimer - trust | Self generative model | 5 random | no | The bot has a good period | no |
| Disclaimer – no answer 1 | NOG 6- bot queries 2 | 5 random | no | In these cases, the bot is constraining from an answer | Debate assessment |
| Disclaimer – no answer 2 | NOG 6- bot queries 2 | 5 random | no | The bot prefers not to give an answer that can be wrong | Debate assessment |
| Disclaimer 4 | NOG 6- bot queries 2 | 5 random | no | The bot is giving an answer anyway | no |
| IDNK | Many pipeline | 30 random | no | Either it doesn’t know the answer or doesn’t have enough data | Debate assessment |

Attention not to duplicate – Check AVM before answering

* 1. Pirkin 5 model implementation. NOG from books queries or from specific/commercial bots
     1. Deep conversational answers

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Type of NOG** | **States** | **Self-generative model implication** | **Paragraphs/ Sentences** | **CVM row 2&3 correlation** | **NOG solutions** |
| NOG 1 from SPCA4 | Discourse | Yes (first +/- BOD and disclaimer) | Paragraph | Don’t apply | The first BOD paragraph linked to the matched BID SPCA that is optimistic or pessimistic + discl. |
| NOG 2 from SPCA4 | Dialogue | Yes (first +/- BOD, w/t disclaimer) | Last Sentence | apply | The last sentence from the BOD paragraph linked to the matched SPCA that is +/-, and match CVM |
| NOG 5 next | User queries 2 | Yes (first +/- BOD and disclaimer) | Last Sentence | apply | The last sentence from the BOD paragraph linked to the matched SPCA that is +/-, match CVM + discl. |
| NOG 5 beginning | User queries 2 | Yes (disclaimer) | Paragraph | Don’t apply | The first BOD paragraph linked to the matched BID SPCA + disclaimer |
| NOG 5 next beginning | User queries 2 | Yes (first +/- BOD and disclaimer) | Paragraph | Don’t apply | The second BOD paragraph linked to the matched BID SPCA that is optimistic or pessimistic + discl. |
| NOG 5 next plot | User queries 2 | Yes (first +/- BOD, w/t disclaimer) | Paragraph | apply | The third BOD paragraph linked to the matched BID SPCA that is +/-, and match CVM |
| NOG 5 summarization | User queries 2 | Yes (first +/- BOD and disclaimer) | SPCA | Don’t apply | The first SPCA summarization of the BOD paragraph is linked to the matched BID SPCA that is +/-, +discl. |
| NOG 5 last section | User queries 2 | Yes (disclaimer) | paragraph | Don’t apply | The last BOD paragraph linked to the matched BID SPCA + disclaimer |
| NOG 5 introduction | User queries 2 | no | paragraph | Don’t apply | The first paragraph of the book/ not linked |
| NOG 5 second answer | User queries 2 | Yes (first +/- BOD, w/t disclaimer) | paragraph | apply | The second BOD paragraph linked to the matched SPCA that is +/-, and match CVM |
| NOG 5 writer present. | User queries 2 | no | paragraph | Don’t apply | The paragraph from chapter 3 writer presentation |

1. The CVM of the SPCA will be mapped to the answers CVM from the Book Output Database. Confidence disclaimer will be replaced by Behavioral disclaimer.
2. The matching will be done similarly with SPCA core answers without DER or NER implications; AVM will be actualized.
3. If the chapter doesn’t have enough paragraphs available for labyrinth model – IDNK disclaimer is provided.
   * 1. Commercial bots answers

**Objectives:**

* Integrating a commercial bot in Cezanne-ai framework and in the bot database.

**Dependencies:** Pirkin model;

**Database/ Vocabularies/External needed**: Q&A labeled data specific for the tasks

**To Dos:**

1. The Subject of SPCA will be replaced by the NER of the commercial bot when user-queries 1 is initiated.
2. The back-up and secondary will be inactivated.
3. CPL will be inactivated except Reaction analysis that will determine exit conditions.
4. From NOG only SPCA core answers will be active.
5. All the utterances-answers will be included also in the main database (with the NER included even if redundant) for cases when the user is also inputting NER1 and the user-queries 1 hasn’t been initiated.
6. In the Cezanne-ai framework only the integrated database needs to be uploaded with specification of the NER1.

**Python code:** adapted Pirkin model

* 1. Pirkin 6 model implementation. NOG from intuition
     1. NOG 6 – Bot queries 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Type of task** | **NER1** | **DER date** | **DER number** | **DER hour** | **NOG solution** |
| Make reservations/ Schedule a meeting | yes | yes | yes | yes | Ask for confirmation |
| no | Ask for DER hour |
| no | n/a | Ask for DER number (example: the persons that will attend) |
| no | n/a | n/a | Ask for DER date |
| no | n/a | n/a | n/a | Ask for NER1 |

1. NER1 is the top entity that should refer to the names of restaurants, doctors, lawyers, bankers…
2. When initiated NOG6 – queries 1 is giving a disclaimer of the user intention that can contain an external link for selling/task-oriented purposes or the reservation or schedules can be done by bot queries 1 or additional interfaces
3. The bot needs to perform 4 steps in order (NER1, DER date, DER number, DER hour) but first will check NER1 and DER if this information is already provided. After each step he will evaluate NER1 and DER and go to the next step
4. After all the steps it will ask for a confirmation.
5. The bot will have databases with questions for reservation and meeting schedule
   * 1. NOG 6 -Bot queries 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **DER\*** | **Bot Queries 2.0** | **DER 1** | **Bot queries 2.1** | **DER 2** | **NOG solution** |
| 70-89% | Has doubts regarding some ideas/topics | None\*\* | n/a | n/a | Initial NOG 1,2,3,4,5 |
| 10-90% | Interest in the person | 10-90% | Disclaimer no answer 1 |
| 50-69% | Has interest in the user as a person | none | n/a | n/a | Initial NOG 1,2,3,4,5 |
| 10-90% | Interest in his opinion | 10-90% | Disclaimer no answer 2 |
| 30-49% | Has interest in the user opinion | none | no | n/a | Initial NOG 1,2,3,4,5 |
| 10-90% | Shares its thoughts | 10-90% | Initial NOG 1,2,3,4,5 + disclaimer 4 |
| 10-29% | Wants to share its thoughts | none | no | n/a | Initial NOG 1,2,3,4,5 |
| 10-90% | no | n/a | Initial NOG 1,2,3,4,5 |

1. We will have a database with 10 bot utterances for each of the 4 policies of the bot queries
2. The utterance will contain the adjective/superlative/adverb who’s meaning the bot wants to clarify.
3. The utterances will be generated in chronological order, checking AVM to see the last bot utterance used and not to duplicate.
4. Only 2 maximum consecutive bot queries 2 will be performed before answering, if the user doesn’t exit before (exit condition in the CPL states).

\*refers at DER relatively vector

\*\*none refers also to the cases DER is less than 10% or higher than 90%

* + 1. NOG 4 – 3rd flow

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **3rd flow** | **First 3rd flow** | **Current SPCA1/4** | **NOG/Solution** |
| 1. | Responding to good news/ Showing agreement/ Linking to your own story | yes | n/a | NOG 4 from specific 3rd flow database |
| no | n/a | Change topic 4 |
| 2. | Responding to bad news/ Getting the user to say more/ Showing surprise | Yes/no | complete | NOG 4 + rephrase + last 2 words of the SPCA classified utterance |
| yes | incomplete | NOG 4 + rephrase + missing SPCA |
| no | incomplete | NOG 4 + rephrase + last 2 words of the SPCA classified utterance |

1. We will have a database with 10 bot utterances for each of the 6 policies of the 3rd flow.
2. The utterances will be generated in chronological order, checking AVM to see the last used bot utterance and avoiding duplicates.
3. The policies where the bot utterance will contain a rephrase database are marked with red.
4. Even if the bot uses the E2E model to answer, the SPCA will be used to elaborate on NOG4.
5. Rephrase database: 10 bot utterances that will contain either missing S/P/C/A, either the last two words of the user’s utterance classified as SPCA1 or SPCA4.
6. Cezanne-ai framework configuration

One of the most important benefits of the model is that it is designed to accommodate multi-domain bots, limited databases and complex business objectives that include advisory sessions, socializing or creative discussions through the art of conversation. Furthermore, commercial chatbots and task-oriented frameworks can be integrated for enhancing the business objectives.

The prototype that was conceptualized and detailed in the previous chapter has a custom configuration, but by implementing Cezanne-ai in a UI framework it will be easy to customize the bot depending on different needs.

Framework configuration

|  |  |  |
| --- | --- | --- |
| **Topic** | **Possibilities** | **What needs to be done for configuration** |
| Grounding | all | Provide databases adapted to the grounding chosen and the location for implication over self-generating model |
| Advisory Bot | All in scope of the paper | Provide databases with NER and training (see below) |
| Commercial bot | all | Provide commercial database |
| Deep conversational | all | Choose the topic (the same with the core or from the books uploaded) |
| Questions/Change topics/ Disclaimers | all | Provide databases with minimum 5 choices for each type |
| Multi-domain bot | All in scope of the paper | Upload databases for each domain; be careful at generalities and prioritization |
| Books training | (Novels, scientific and scripts) | Upload books, specifying the type and a description of the author |
| Database training | All in scope of the paper | Upload databases with utterances and answers; utterances can be simple sentences, as SPCA can accommodate this type of labeling |
| NER | All | Provide databases with NER and their priority. For NER location provide also GPS coordinates for the additional algorithms |
| Task-oriented | Reservation/meeting schedules | Define the type of task-oriented and questions (database) for bot queries 1 or input a link in the bot queries 1 to redirect the conversation to a selling framework, for example |

1. Cezanne-ai after the famous painter of “The conversation”; There are similarities with Paul Cezanne work also in terms of the hybrid NLP/AGI approach to conversation. [↑](#footnote-ref-1)
2. it incorporates elements of a functional specification document and a detailed architecture & solution of the proposed framework. [↑](#footnote-ref-2)
3. But first we need to evaluate the effectiveness of the framework [↑](#footnote-ref-3)
4. In Fundamentals file on github we are arguing more on this topic. [↑](#footnote-ref-4)
5. For a comprehensive understanding over the terminology, we will refer to the research paper [↑](#footnote-ref-5)
6. Check the data-prerequisites file on the github repository [↑](#footnote-ref-6)
7. See the research paper, especially Pirkin 2 model presentation [↑](#footnote-ref-7)