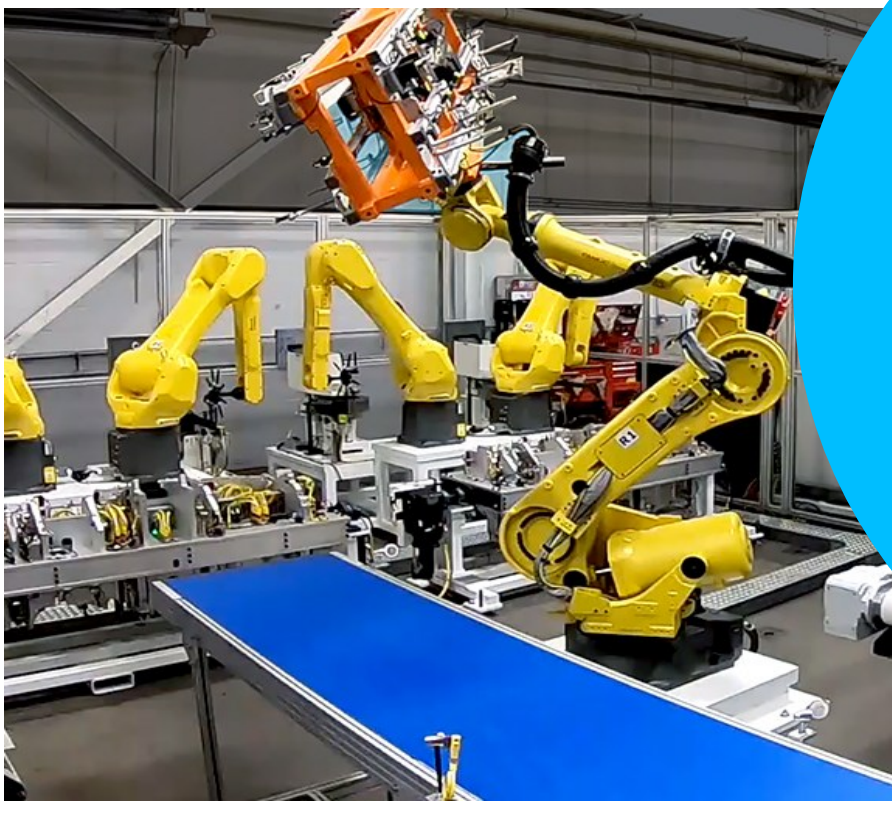
University of Stuttgart
Institute of Industrial Automation
and Software Engineering



Experimental evaluation of neural language models for semantic matching between user requests and executable functions of an automation system

Final Presentation

Research Thesis No. : 3497

Supervisor: Yuchen Xia

Gopal Chitrasen Panigrahi

Master in Information Technology

Contents


Motivation
Conception
Implementation
Results and Evaluation
Conclusion

Motivation

Motivation

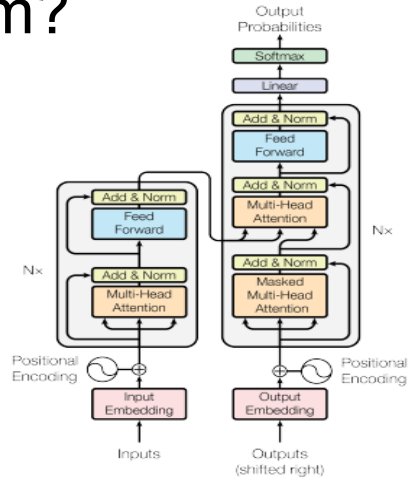
CP Factory

- Technical expert required in the field of automation for any query related to the system
- Complete information on how the system is programmed is not available
- Additional software required for interaction with the user e.g. SCADA, CMMS or an HMI
- The entire system is not so friendly for new users

Thesis Problem Statement

How can we use NLP models for the maintenance and diagnosis of an automation system?

{1} Typical SCADA/HMI system



{2} Standard Transformer model

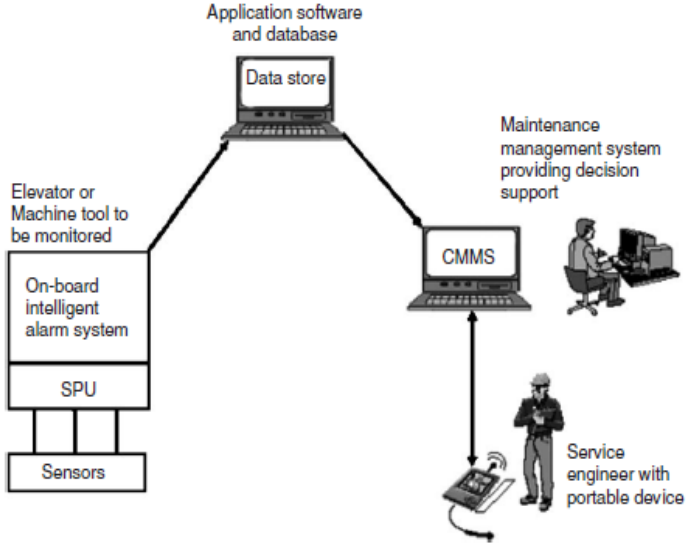
Why NLP ?

- Understanding of different types of code and the context of an automation system
- Adaptability and Scalability of Large Language Models

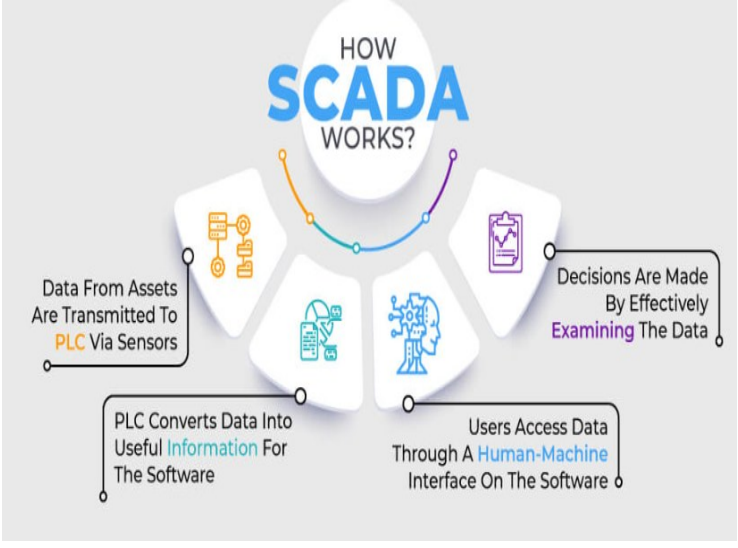
Background

Background

Methods used for diagnosis and maintenance in an Automation system



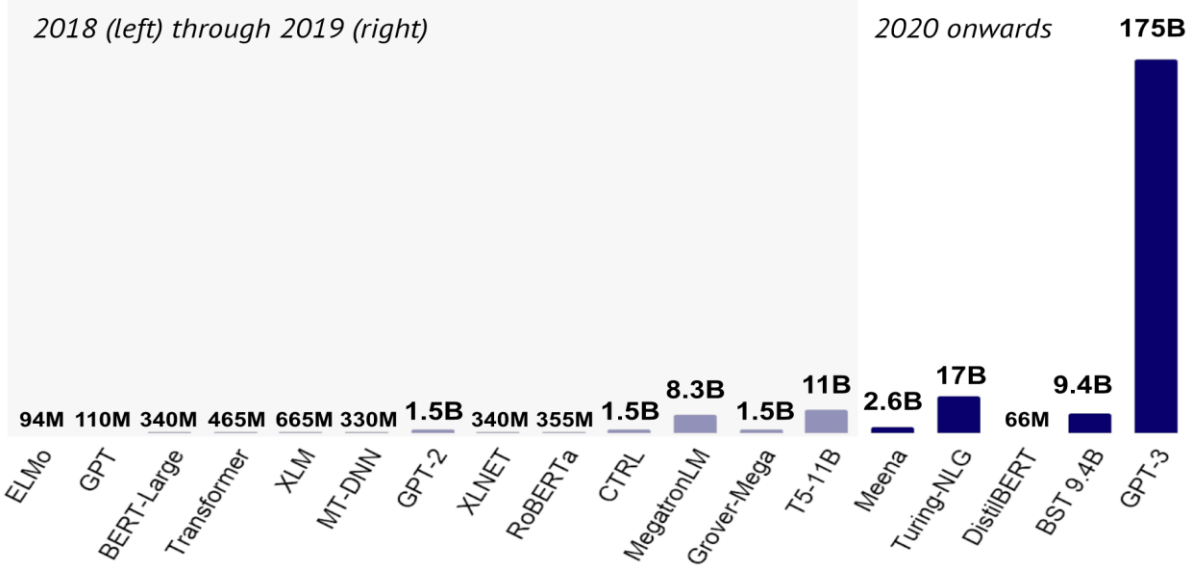
{3} CMMS system



{4} SCADA system

Background

Evolution of Large Language Models



{5} Evolution of LLM's

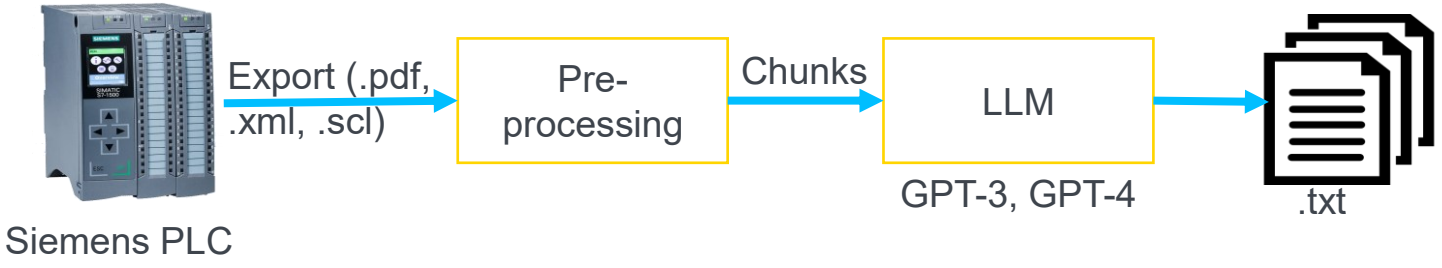
GPT-3 model consists of

- 175B parameters
- 96 attention layers
- 3.2M batch size

Conception

Conception

Phase1: Feature Extraction

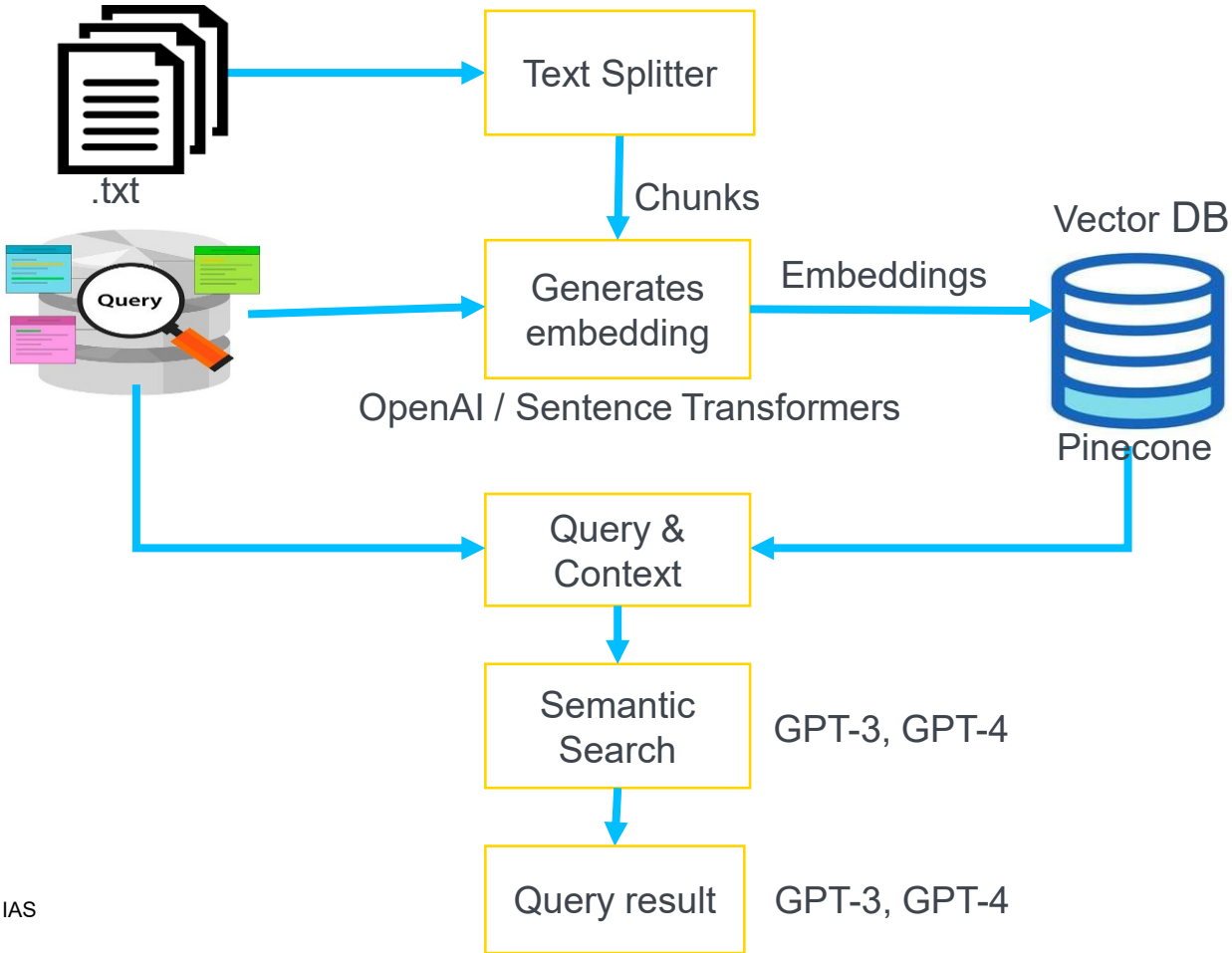


Prompts: Large language models (LLMs) have made it possible for everyone to interact with them, using prompts in natural language.

```
def generate_explanation(code_snippet):
    prompt = "read the SCL file exported from the Siemens TIA Portal and explain it:\n```python\n" + code_snippet + "\n```"
    response = openai.Completion.create(
        engine="text-davinci-003",
        prompt=prompt,
```

Conception

Phase2: Semantic Search



Implementation

Dataset

Siemens TIA portal Dataset

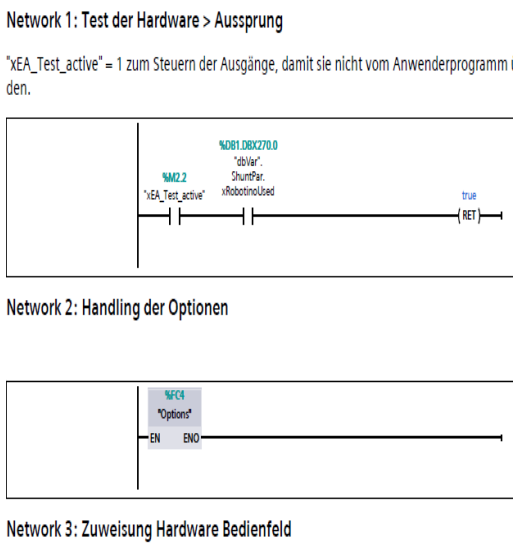
Siemens TIA Portal (V16) → Export raw data from PLC in .xml, .scl and .pdf formats

```
<Parts>
  <Access Scope="GlobalVariable" UId="21">
    <Symbol>
      <Component Name="xEA_Test_active" />
    </Symbol>
  </Access>
  <Access Scope="GlobalVariable" UId="22">
    <Symbol>
      <Component Name="dbVar" />
      <Component Name="ShuntPar" />
      <Component Name="xRobotinoUsed" />
    </Symbol>
  </Access>
  <Access Scope="LiteralConstant" UId="23">
    <Constant>
      <ConstantType>Bool</ConstantType>
      <ConstantValue>true</ConstantValue>
    </Constant>
  </Access>
  <Part Name="Contact" UId="24" />
  <Part Name="Contact" UId="25" />
  <Part Name="ReturnValue" UId="26" />
</Parts>
```

Exported in XML format

```
#xInitEn := #xInit OR #iMode <> 1;

IF #xInitEn OR NOT #xInitDone THEN

  #iZero := 0;
  IF #RcvOrder.diONo <> 0 AND #RcvOrder.iOPos <> 0 THEN
    "OpReset"(diONo := #RcvOrder.diONo,
              iOPos := #RcvOrder.iOPos,
              pRcvHeader := #RcvHeader);
    IF ENO THEN
      #iRetVal := FILL(BVAL := #iZero, BLK => #RcvOrder);
      #iRetVal := FILL(BVAL := #iZero, BLK => #RcvHeader);
    END_IF;
  END_IF;

  "dbVar".Hmi.ISTB.xErrMes := FALSE;

  #iRetVal := 0;
  #xInitDone := TRUE;
  #iStep := 1;
  #xSubSecOccupied := False;
  #xBusy := FALSE;    // 16.06.2016 TBWL
  #xTagError := FALSE;
  #xOrderWarning := FALSE;
```

Exported in SCL format



Exported in PDF format

Dataset - Exploratory Data Analysis

3 Parts Data exploratory Analysis

1 **Pre-processing data**

2 **Generating explanations**

3 **Information in a text file**

- Creating chunks:

```
<Parts>
  <Access Scope="GlobalVariable" UId="21">
    <Symbol>
      <Component Name="xEA_Test_active" />
    </Symbol>
  </Access>
  <Access Scope="GlobalVariable" UId="22">
    <Symbol>
      <Component Name="dbVar" />
      <Component Name="ShuntPar" />
      <Component Name="xRobotinoUsed" />
    </Symbol>
  </Access>
  <Access Scope="LiteralConstant" UId="23">
    <Constant>
      <ConstantType>Bool</ConstantType>
      <ConstantValue>true</ConstantValue>
    </Constant>
  </Access>
  <Part Name="Contact" UId="24" />
  <Part Name="Contact" UId="25" />
  <Part Name="ReturnValue" UId="26" />
</Parts>
```

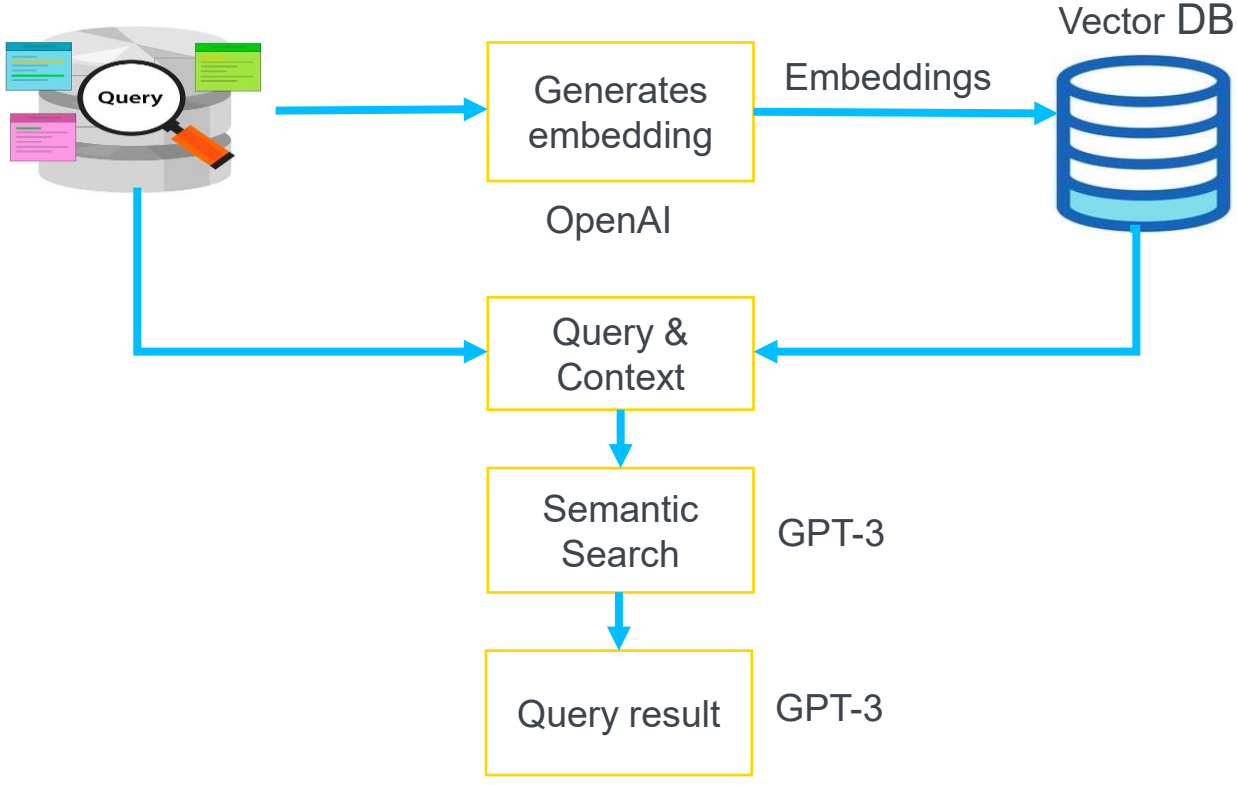
```
print(docs[0].page_content)

The above code snippet is written in Structured Control Language (SCL) which is us
is used to define the functionality of a function block called "Branch_DefaultMes"
also defines the variables used within the function block It also contains some co
is used to control a robotic arm used in a manufacturing process It reads data fro
timers to delay certain actions It also initializes variables and sets up a state
of a program written in the Siemens SCL language It reads an RFID tag and extracts
```

```
This PLC code is a logic statement which sets a Global Vari
This XML code shows a Programmable Logic Controller (PLC) c
This PLC code is a program that is designed to control a tr
This XML is a PLC code that sets up a title for the program
This PLC code creates a system of logic gates that control
This PLC code is setting up the connections between differe
The PLC code in the XML is setting up a call to a function
```

Conclusion: Remove all unuseful information from all files and generate explanations using an LLM

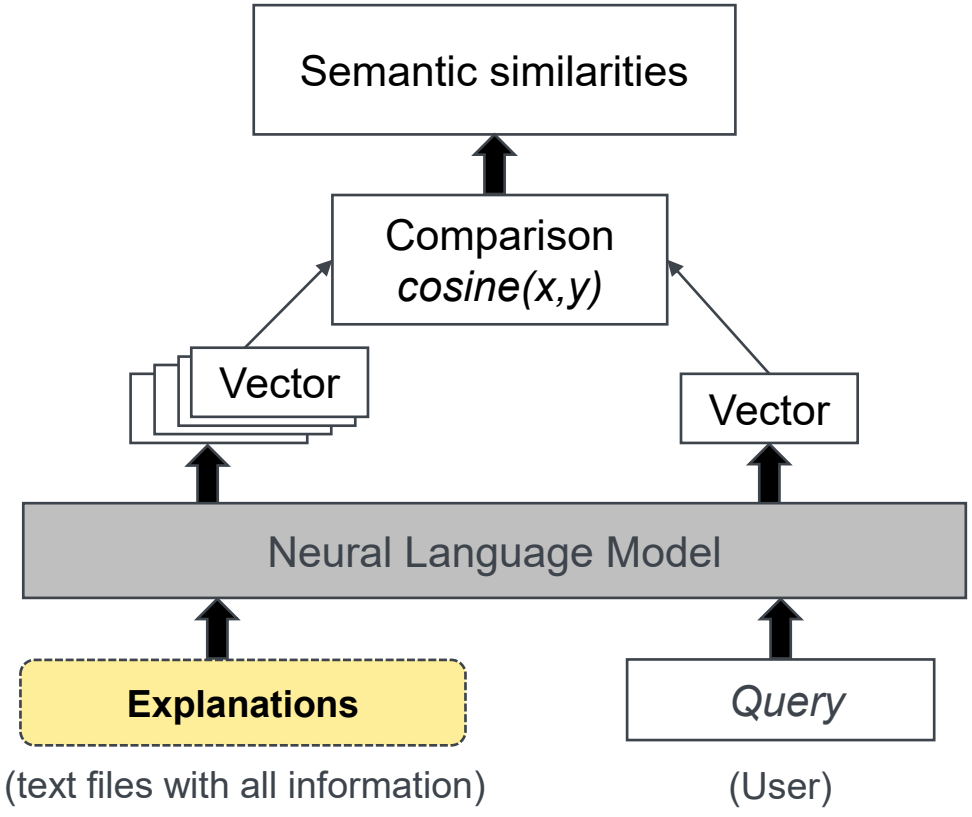
Semantic Matching

Search algorithm using OpenAI embeddings and vector database



Semantic Matching

How does the matching work?



Results and Evaluation

Results

Evaluation Metric: User feedback

```
query = "explain the main section of a branch"
get_answer(query)
```

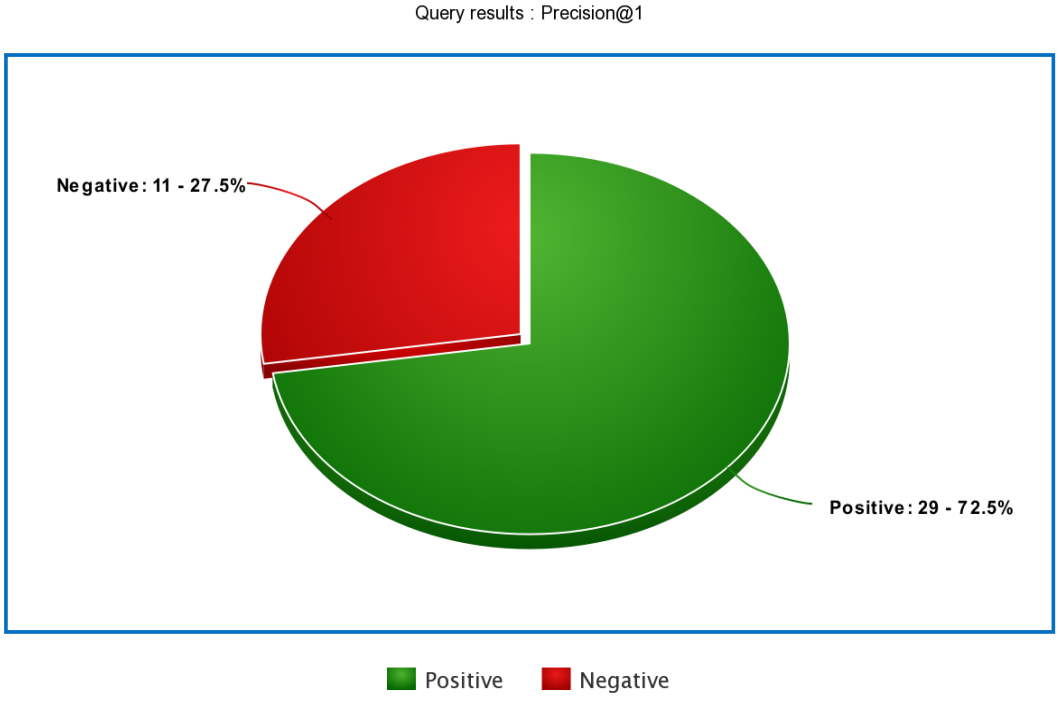
```
[Document(page_content='The above code snippet is written in Structured Control Language (SCL) which is used to program and co
' The main section of a branch is responsible for reading the inputs and outputs of the function block and defining the variab
used within the function block. It also contains comments to explain the purpose of the function block. Additionally, it sets
imers to delay certain actions, initializes variables, and sets up a state machine to control the robot's behavior.'
```

```
[ ] query = "when does the LED gets activated"
get_answer(query)
```

```
[Document(page_content='to a function block called "OutLED". This function block has an input parameter named "xInit" and an o
' The LED gets activated when the "Initial_Call" local variable is set to true.'
```

Multiple queries were resolved via the user feedback evaluation (extracted from the TIA portal in PDF format) for all explanations stored in a vector database

Results

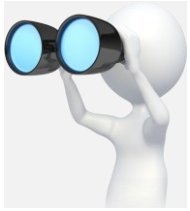


User feedback evaluation method was applied for 40 different queries

Conclusion and Future Scope

Conclusion and Future Scope

- Successfully linked different user queries with the proper executable functions by achieving high accuracy in semantic matching
- This evaluation sheds light on whether implementing the system in nearly real-time circumstances is feasible. Evaluating and analysing the outcomes of the assessment metrics enabled to evaluate the effectiveness, advantages, and drawbacks of the suggested solution



- Explore and evaluate the complete dataset exported from the CP factory
- Integrate with the TIA portal for live data and resolve user queries in a more dynamic and efficient way

References

Images Sources

{1} https://www.smart-energy.com/regional-news/asia/scada-india/

{2} Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. https://doi.org/10.48550/arXiv.1706.03762

{3} Gilabert, E., & Arnaiz, A. (2006). Intelligent automation systems for predictive maintenance: A case study. *Robotics and Computer-Integrated Manufacturing*, *22*(5-6), 543-549. https://doi.org/10.1016/j.rcim.2005.12.010

{4} https://www.biz4intellia.com/blog/importance-of-updating-existing-plcsc/

{5} https://www.theaidream.com/post/openai-gpt-3-understanding-the-architecture

University of Stuttgart
Germany

Thank you!

Gopal Chitrasen Panigrahi

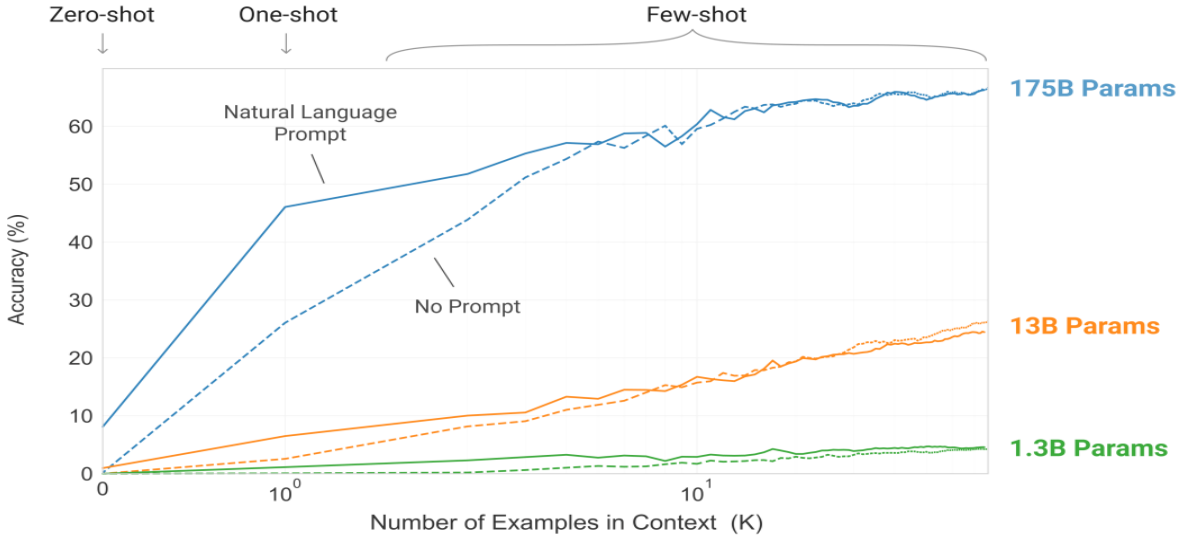
e-mail st179549@stud.uni-stuttgart.de

University of Stuttgart

Institute for Industrial Automation and
Software Engineering
Prof. Dr.-Ing. Dr. h.c. Michael Weyrich

Allmandring Vahingen Stuttgart

Large Language Models



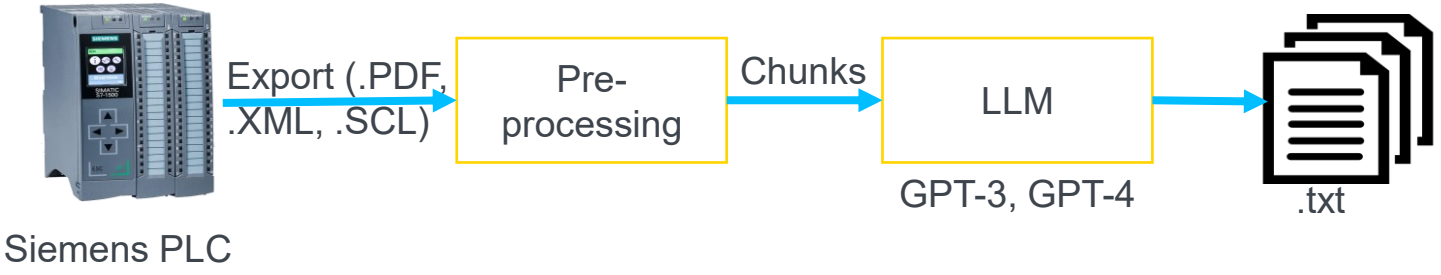
GPT-3 model consists of:

- 175B parameters
- 96 attention layers
- 3.2M batch size

{4} LLM uses in-context information efficiently

Conception

Phase1: Feature Extraction



- All necessary data is exported from the Siemens TIA portal in the required format
- In the pre-processing step, each ladder network from the XML or SCL file is created into chunks
- Using LLM such as GPT-3 the explanations are created for the following input chunk provided

Dataset - Exploratory Data Analysis

3 Parts Data exploratory Analysis

1

Pre-processing data

- Creating chunks:
  - Create chunks depending upon networks in the PLC ladder logic for XML file types
  - As the LLM cannot process the whole XML file at once, chunks are created for each network from the ladder code
  - Pre-processing step for SCL files are not required

2

Generating explanations

- Takes the chunks as input and generates explanations using GPT-3 model
- Eliminates all the unnecessary information from the dataset and provides necessary results in a human-readable format
- In-context learning and prompts helps the LLM to perform better

3

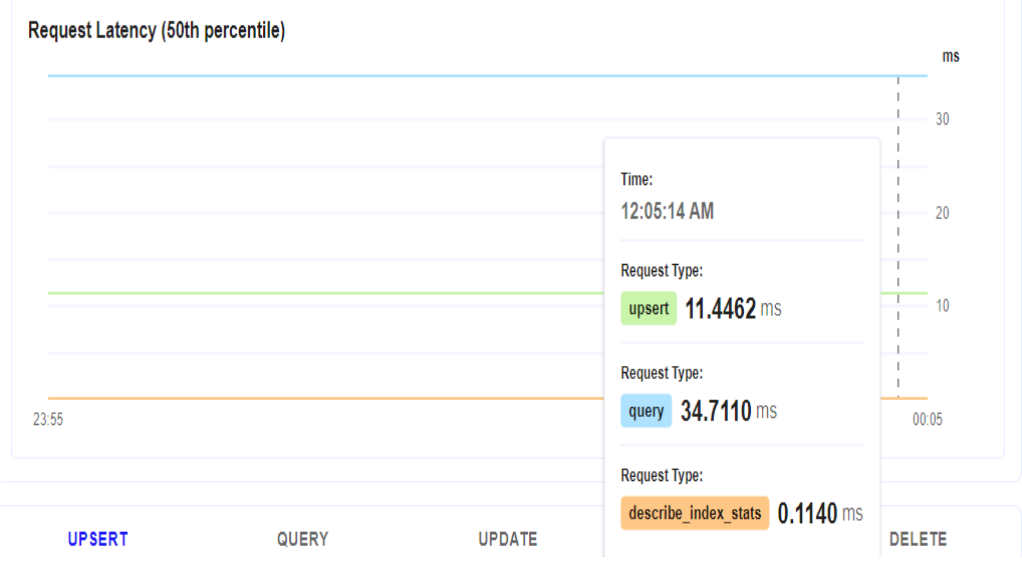
Information in a text file

- After generating all the results, necessary explanations are stored in a text file for a ladder or structure language code

Conclusion: Remove all unuseful information from all files and generate explanations using an LLM

Semantic Matching

Pinecone Vector DB



Five text files with explanations which correspond to 33 vectors and a single query :

- Uploading time for 33 vectors: 11.45ms
- Single query resolved in 34.7ms