2024
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
UNITED STATES
SAN JOSE, CA, USA
MARCH 4-7, 2024

Requirements Recognition for Verification IP Design Using Large Language Models

Siarhei Zalivaka

SK hynix NAND Product Solutions Poland

accellera
SYSTEMS INITIATIVE

Agenda

- Introduction
- Proposed Approach
- Model Training
- Results
- Conclusion

2024
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
UNITED STATES
SAN JOSE, CA, USA
MARCH 4-7, 2024

Introduction

accellera
SYSTEMS INITIATIVE

Background

M-PHY Version 5.0 – 232 pages

UFS Version 4.0 – 489 pages

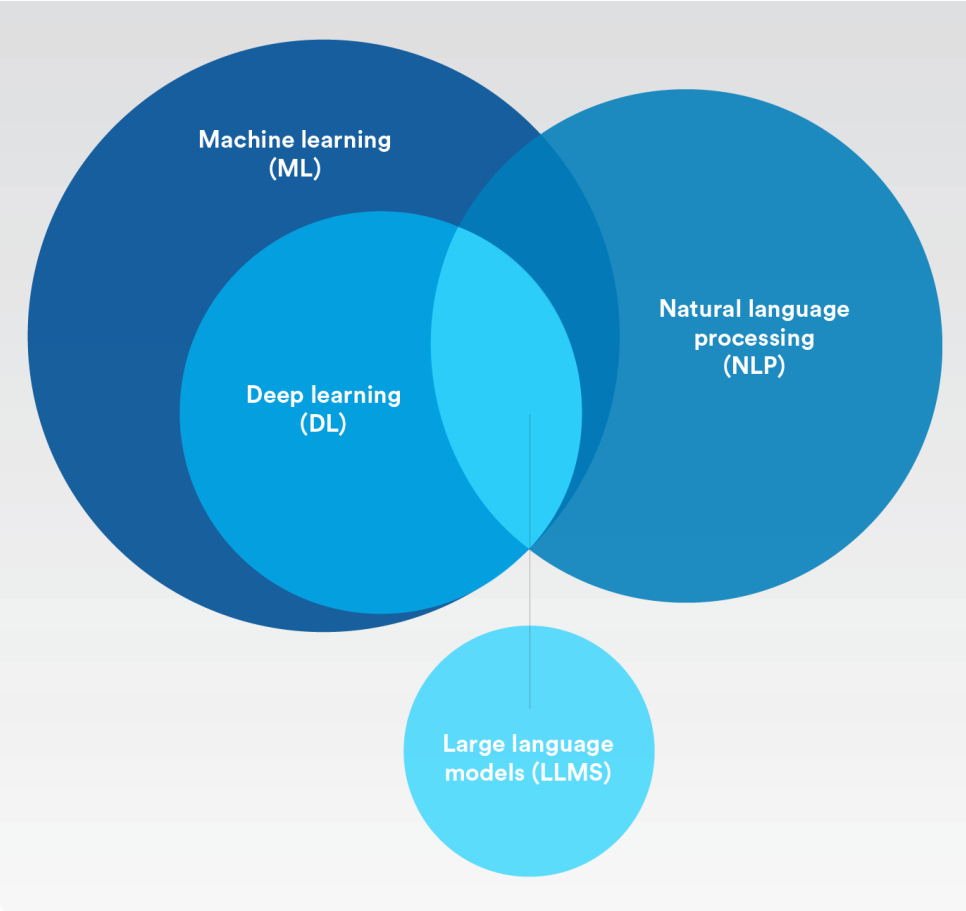
Version 5.0 – 1299 pages

Version 3.0 – 288 pages

Current Solution

Sub Feature	Measure	Jenkins Status (Pass/Fail/ Not Run) Groups/etc	All Measure	qTest review status	Jenkins debug info	Review Status	Test implementation status	Checker implementation status
				drop				
	test : *	952/39/0/991		drop	› info			
MPHY_SPEC_0112	tesT :		tesT :	* drop				
	group:		group:					
MPHY_SPEC_0113	test: rxtx_sane.rxtx_pld_pmc_unavailable	2/0/0/2	test: rxtx_sane.rxtx_pld_pmc_unavailable		› info	☑	☑	
	group: **rxtx_active_lanes_cg	93.33%	group: **rxtx_active_lanes_cg		› info			
MPHY_SPEC_0114	test:		test:	N/A		☑		
	group:		group:					
MPHY_SPEC_0115	test: rxtx_sane.rxtx_pld_pmc_unavailable	2/0/0/2	test: rxtx_sane.rxtx_pld_pmc_unavailable		› info	☑	☑	
	group: **rxtx_active_lanes_cg.tx_available_lanes_cp **rxtx_active_lanes_cg.rx_available_lanes_cp **rxtx_active_lanes_cg.cc_rx_tx_available	86.67%	group: **rxtx_active_lanes_cg.tx_available_lanes_cp **rxtx_active_lanes_cg.rx_available_lanes_cp **rxtx_active_lanes_cg.cc_rx_tx_available		› info			
MPHY_SPEC_0116	test:		test:	* N/A		☑		
	group:		group:					
MPHY_SPEC_0117	test:		test:	* N/A		☑		
	group:		group:					

Another Solution



Ryan, Kevin. "The role of natural language in requirements engineering." [1993] Proceedings of the IEEE International Symposium on Requirements Engineering. IEEE, 1993.

Dias Canedo, Edna, and Bruno Cordeiro Mendes. "Software requirements classification using machine learning algorithms." Entropy 22.9 (2020): 1057.

Khayashi, Fatemeh, et al. "Deep Learning Methods for Software Requirement Classification: A Performance Study on the PURE dataset." arXiv preprint arXiv:2211.05286 (2022).

Ivanov, Vladimir, et al. "Extracting Software Requirements from Unstructured Documents." International Conference on Analysis of Images, Social Networks and Texts. Cham: Springer International Publishing, 2021.

Existing Datasets

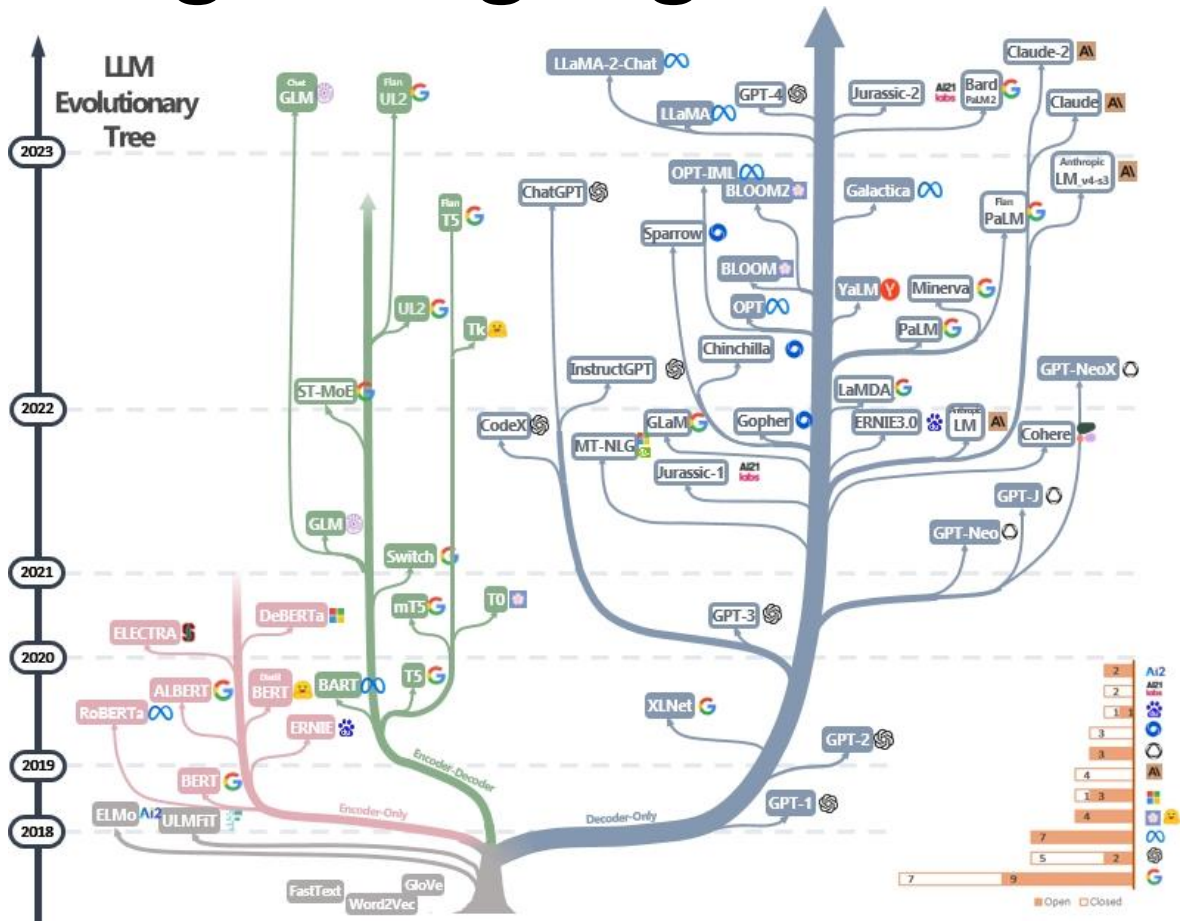
- PURE
  - Ferrari, A., Spagnolo, G. O., & Gnesi, S. (2017, September). PURE: A dataset of public requirements documents. In 2017 IEEE 25th International Requirements Engineering Conference (RE) (pp. 502-505). IEEE.
- PROMISE
  - Sayyad Shirabad, J. and Menzies, T.J. (2005) The PROMISE Repository of Software Engineering Databases. School of Information Technology and Engineering, University of Ottawa, Canada. Available: http://promise.site.uottawa.ca/SERepository
- DOORS
  - Hull, Elizabeth, et al. "DOORS: a tool to manage requirements." Requirements engineering (2002): 187-204.

2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES
SAN JOSE, CA, USA
MARCH 4-7, 2024

Proposed Solution

accellera SYSTEMS INITIATIVE

Large Language Models



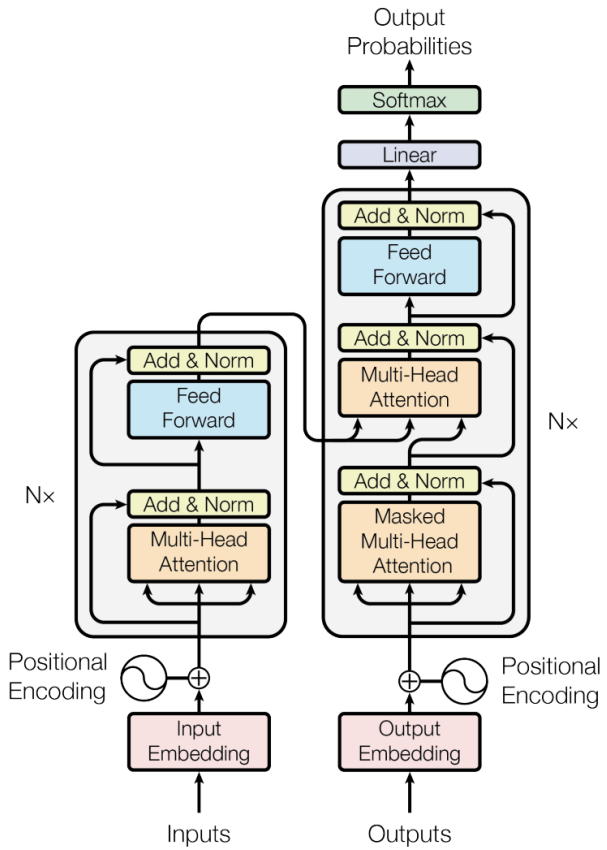
Yang, Jingfeng, et al. "Harnessing the power of llms in practice: A survey on chatgpt and beyond." *arXiv preprint arXiv:2304.13712* (2023).

Ivanov, Vladimir, et al. "Extracting Software Requirements from Unstructured Documents." International Conference on Analysis of Images, Social Networks and Texts. Cham: Springer International Publishing, 2021.

Transformers

BERT

Encoder

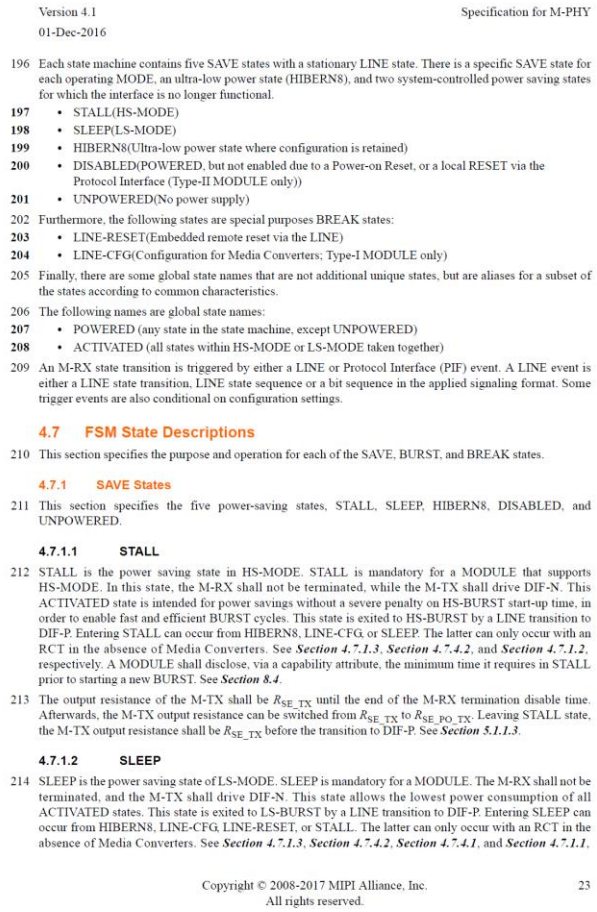


GPT

Decoder

https://heidloff.net/article/foundation-models-transformers-bert-and-gpt/

Used Dataset



Version 4.1 Specification for M-PHY
01-Dec-2016

196 Each state machine contains five SAVE states with a stationary LINE state. There is a specific SAVE state for each operating MODE, an ultra-low power state (HIBERN8), and two system-controlled power saving states for which the interface is no longer functional.

197 • STALL(HS-MODE)
198 • SLEEP(LS-MODE)
199 • HIBERN8(Ultra-low power state where configuration is retained)
200 • DISABLED(POWERED, but not enabled due to a Power-on Reset, or a local RESET via the Protocol Interface (Type-II MODULE only))
201 • UNPOWERED(No power supply)

202 Furthermore, the following states are special purposes BREAK states:

203 • LINE-RESET(Embedded remote reset via the LINE)
204 • LINE-CFG(Configuration for Media Converters; Type-I MODULE only)

205 Finally, there are some global state names that are not additional unique states, but are aliases for a subset of the states according to common characteristics.

206 The following names are global state names:

207 • POWERED (any state in the state machine, except UNPOWERED)
208 • ACTIVATED (all states within HS-MODE or LS-MODE taken together)

209 An M-RX state transition is triggered by either a LINE or Protocol Interface (PIF) event. A LINE event is either a LINE state transition, LINE state sequence or a bit sequence in the applied signaling format. Some trigger events are also conditional on configuration settings.

4.7 FSM State Descriptions

210 This section specifies the purpose and operation for each of the SAVE, BURST, and BREAK states.

4.7.1 SAVE States

211 This section specifies the five power-saving states, STALL, SLEEP, HIBERN8, DISABLED, and UNPOWERED.

4.7.1.1 STALL

212 STALL is the power saving state in HS-MODE. STALL is mandatory for a MODULE that supports HS-MODE. In this state, the M-RX shall not be terminated, while the M-TX shall drive DIF-N. This ACTIVATED state is intended for power savings without a severe penalty on HS-BURST start-up time, in order to enable fast and efficient BURST cycles. This state is exited to HS-BURST by a LINE transition to DIF-P. Entering STALL can occur from HIBERN8, LINE-CFG, or SLEEP. The latter can only occur with an RCT in the absence of Media Converters. See ***Section 4.7.1.3***, ***Section 4.7.4.2***, and ***Section 4.7.1.2***, respectively. A MODULE shall disclose, via a capability attribute, the minimum time it requires in STALL prior to starting a new BURST. See ***Section 8.4***.

213 The output resistance of the M-TX shall be R_{SE_TX} until the end of the M-RX termination disable time. Afterwards, the M-TX output resistance can be switched from R_{SE_TX} to $R_{SE\_PO\_TX}$. Leaving STALL state, the M-TX output resistance shall be R_{SE_TX} before the transition to DIF-P. See ***Section 5.1.1.3***.

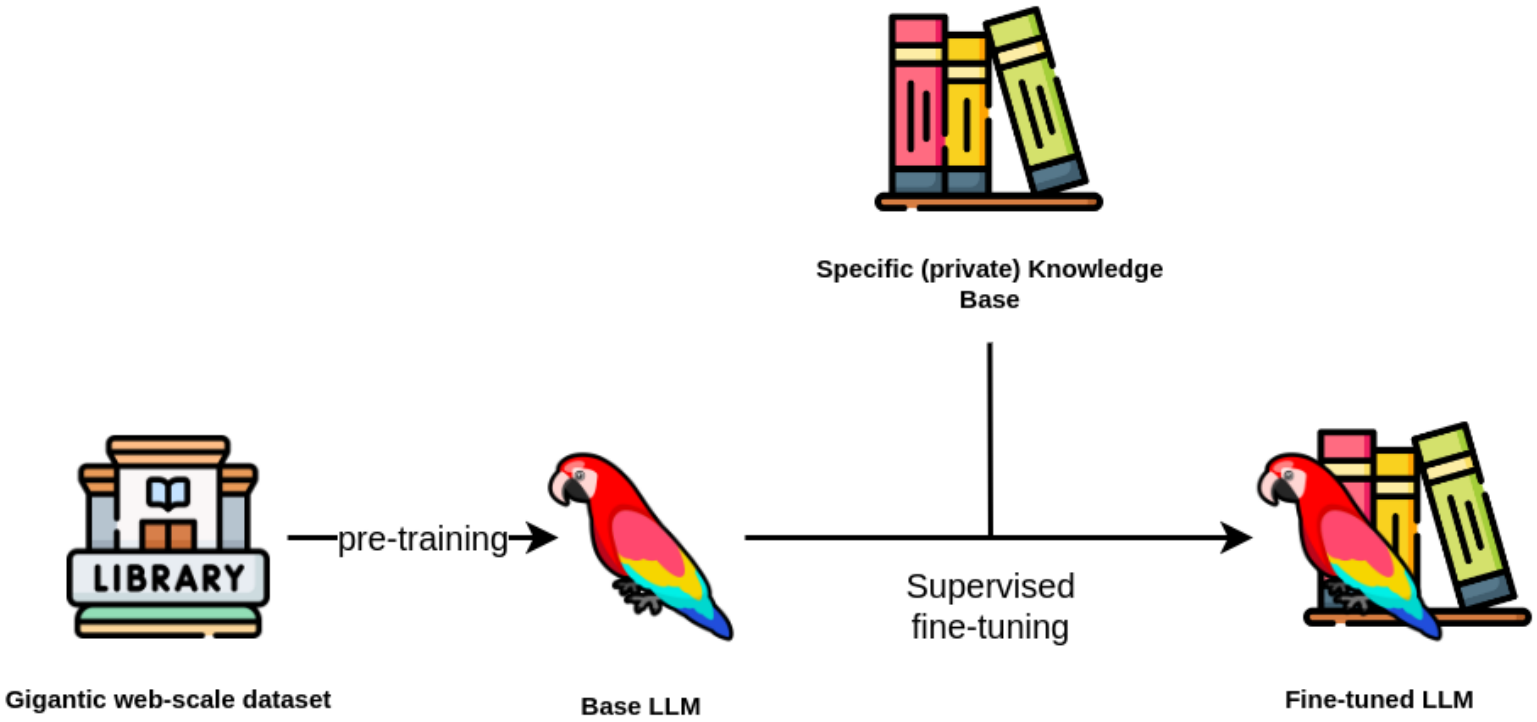
4.7.1.2 SLEEP

214 SLEEP is the power saving state of LS-MODE. SLEEP is mandatory for a MODULE. The M-RX shall not be terminated, and the M-TX shall drive DIF-N. This state allows the lowest power consumption of all ACTIVATED states. This state is exited to LS-BURST by a LINE transition to DIF-P. Entering SLEEP can occur from HIBERN8, LINE-CFG, LINE-RESET, or STALL. The latter can only occur with an RCT in the absence of Media Converters. See ***Section 4.7.1.3***, ***Section 4.7.4.2***, ***Section 4.7.4.1***, and ***Section 4.7.1.1***.

each state machine contains five save states with a stationary line state there is a specific save state for eacl	0
stallhsmode	1
sleeplsmode	1
hibern8ultralow power state where configuration is retained	1
disabledpowered but not enabled due to a poweron reset or a local reset via the protocol interface typeii n	1
unpoweredno power supply	0
furthermore the following states are special purposes break states	0
lineresetembedded remote reset via the line	1
linecfgconfiguration for media converters typei module only	0
finally there are some global state names that are not additional unique states but are aliases for a subset o	0
the following names are global state names	0
powered any state in the state machine except unpowered	0
activated all states within hsmode or lsmode taken together	0
an mrx state transition is triggered by either a line or protocol interface pif event a line event is either a line	0
this section specifies the purpose and operation for each of the save burst and break states	0
71 save states	0
this section specifies the five powersaving states stall sleep hibern8 disabled and unpoweredB	0
711 stall	1
stall is the power saving state in hsmode stall is mandatory for a module that supports hsmode in this state	1
the output resistance of the mtx shall be rsetx until the end of the mrx termination disable time afterwards	0
sleep is the power saving state of lsmode sleep is mandatory for a module the mrx shall not be terminated a	1

M-PHY 4.1 Specification:
1130 text fragments (332 requirements and 798 non-requirements)

Fine-Tuning



https://neo4j.com/developer-blog/fine-tuning-retrieval-augmented-generation/

2024
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
UNITED STATES
SAN JOSE, CA, USA
MARCH 4-7, 2024

Model Training

accellera
SYSTEMS INITIATIVE

Datasets

Training set

- Based on M-PHY 4.1. specification
- Parsed manually
- Requirements can be bigger than a sentence

Validation set

- Based on M-PHY 4.1. specification
- Parsed automatically (PyPDF2)
- The requirement has a lenght of one sentence

Training and Inference

```python
# Required libraries
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, recall_score, precision_score, f1_score
import torch
from transformers import TrainingArguments, Trainer
from transformers import AutoTokenizer, AutoConfig, AutoModelForSequenceClassification
from transformers import EarlyStoppingCallback

# Defining the model

model_name = "gpt2"
tokenizer = AutoTokenizer.from_pretrained(model_name)
tokenizer.pad_token = tokenizer.eos_token
config = AutoConfig.from_pretrained(model_name, num_lables=2)
config.pad_token_id = config.eos_token_id
model = AutoModelForSequenceClassification.from_pretrained(model_name, config=config)

# Preparing the data
df = pd.read_csv('training_set.csv', header=None)
X_train = list(df[0])
y_train = list(df[1])
X_tokenized_train = tokenizer(X_train, padding="max_length", truncation=True)
```

```python
# Training
args = TrainingArguments(
    output_dir="output",
    evaluation_strategy="steps",
    eval_steps=500,
    per_device_train_batch_size=2,
    per_device_eval_batch_size=2,
    num_train_epochs=10,
    seed=0,
    load_best_model_at_end=True,
)
trainer = Trainer(
    model=model,
    args=args,
    train_dataset=train_dataset,
    eval_dataset=valid_dataset,
    compute_metrics=compute_metrics,
    callbacks=[EarlyStoppingCallback(early_stopping_patience=3)],
)
trainer.train()

# Predicting for unknown texts

test_trainer = Trainer(model)
raw_pred, _, _ = test_trainer.predict(test_dataset)
y_pred = np.argmax(raw_pred, axis=1)
```

Training Environment

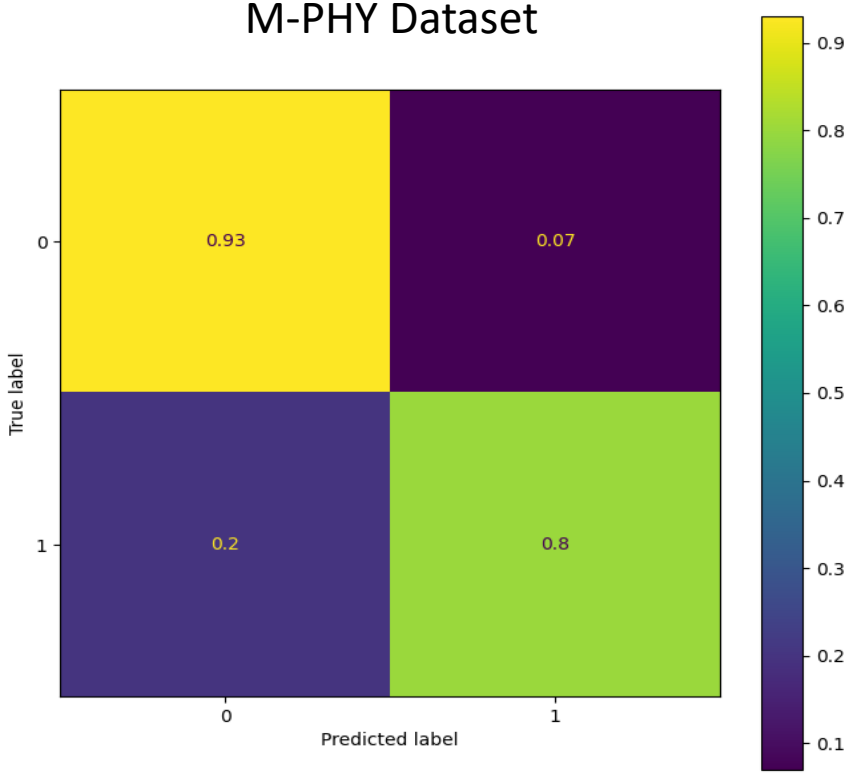
Nvidia A100

Nvidia V100

Google Colab

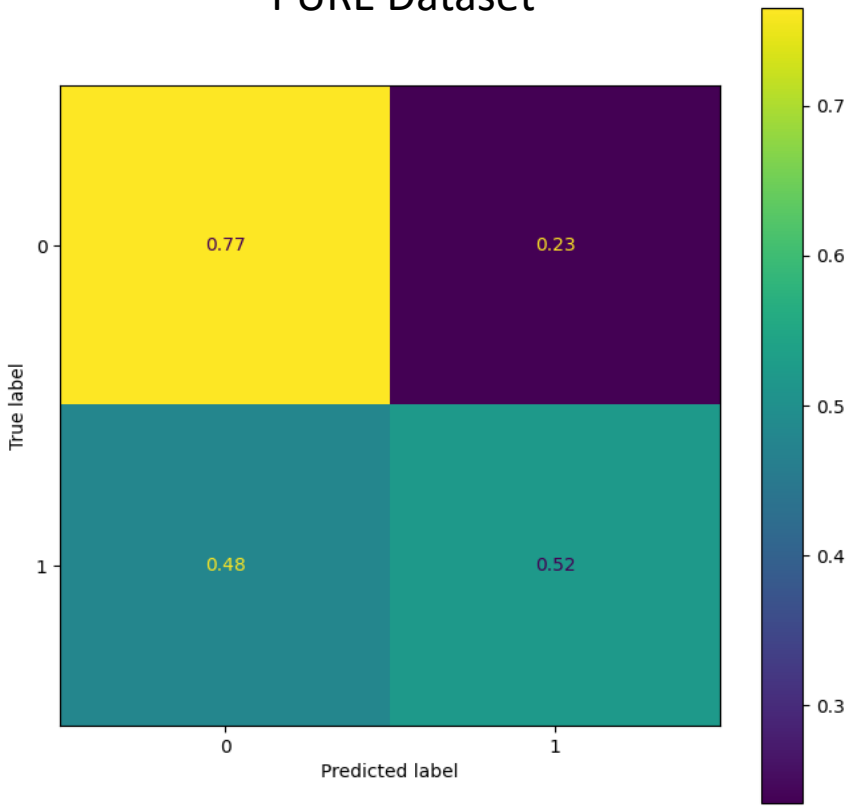
Training Results

M-PHY Dataset



Precision = 0.618 Recall = 0.874

PURE Dataset



Precision = 0.513 Recall = 0.778

Comparison to ChatGPT 3.5

Could you please classify each sentence of the following text as a requirement or non-requirement?

10.3.5 Invalidate Flow Control
Due to the variety of caching architectures and queuing strategies, implementations may vary greatly with respect to
invalidation latency and throughput. It is possible that a TA may generate Invalidate Requests at a rate that exceeds the
average ATC service rate. When this happens, the credit based flow control mechanisms will throttle the TA issue rate. A
side effect of this is congestion spreading to other channels and Links through the credit based flow control mechanism.
Depending on the frequency and duration of this congestion, performance may suffer. It is highly recommended that TA
and its associated software implement higher level flow control mechanisms.
To assist with the implementation of Invalidate Flow Control, an ATC must publish the number of Invalidate Requests it
can buffer before back pressuring the Link. This field applies to all invalidations serviced by the Function, independent
of the size of the invalidation. This value is communicated in the Invalidate Queue Depth field in the ATS capability
structure (see Section 7.8.8). A value of 0 0000b indicates that invalidate flow control is not necessary to this Function.
IMPLEMENTATION NOTE
Invalidate Flow Control
A Function may indicate that invalidate flow control is not required when one or more of the following is true:
1. The Function can handle invalidations at the maximum arrival rate of Invalidate Requests.
2. The Function will not or very rarely cause Link backpressure (performance loss is negligible).
3. The Function can fully buffer the maximum number of incoming invalidations without back pressuring
the Link.

Question

In the provided text, the sentences can be classified as follows:

Requirements:

1. "To assist with the implementation of Invalidate Flow Control, an ATC must publish the number of Invalidate Requests it can buffer before back pressuring the Link."
2. "This field applies to all invalidations serviced by the Function, independent of the size of the invalidation."
3. "A value of 0 0000b indicates that invalidate flow control is not necessary to this Function."

Non-requirements (Implementation Note):

1. "A Function may indicate that invalidate flow control is not required when one or more of the following is true:"
2. "The Function can handle invalidations at the maximum arrival rate of Invalidate Requests."
3. "The Function will not or very rarely cause Link backpressure (performance loss is negligible)."
4. "The Function can fully buffer the maximum number of incoming invalidations without back pressuring the Link."

The sentences under "Requirements" specify conditions or actions that are necessary for implementing Invalidate Flow Control. The sentences under "Non-requirements" provide additional information but do not impose any specific requirements.

Answer

2024
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
UNITED STATES
SAN JOSE, CA, USA
MARCH 4-7, 2024

Results

accellera
SYSTEMS INITIATIVE

Inference on PCIe Specification

10.3.5 Invalidate Flow Control

Due to the variety of caching architectures and queuing strategies, implementations may vary greatly with respect to invalidation latency and throughput. It is possible that a TA may generate Invalidate Requests at a rate that exceeds the average ATC service rate. When this happens, the credit based flow control mechanisms will throttle the TA issue rate. A side effect of this is congestion spreading to other channels and Links through the credit based flow control mechanism. Depending on the frequency and duration of this congestion, performance may suffer. It is highly recommended that TA and its associated software implement higher level flow control mechanisms.

To assist with the implementation of Invalidate Flow Control, an ATC must publish the number of Invalidate Requests it can buffer before back pressuring the Link. This field applies to all invalidations serviced by the Function, independent of the size of the invalidation. This value is communicated in the Invalidate Queue Depth field in the ATS capability structure (see Section 7.8.8). A value of 0 0000b indicates that invalidate flow control is not necessary to this Function.

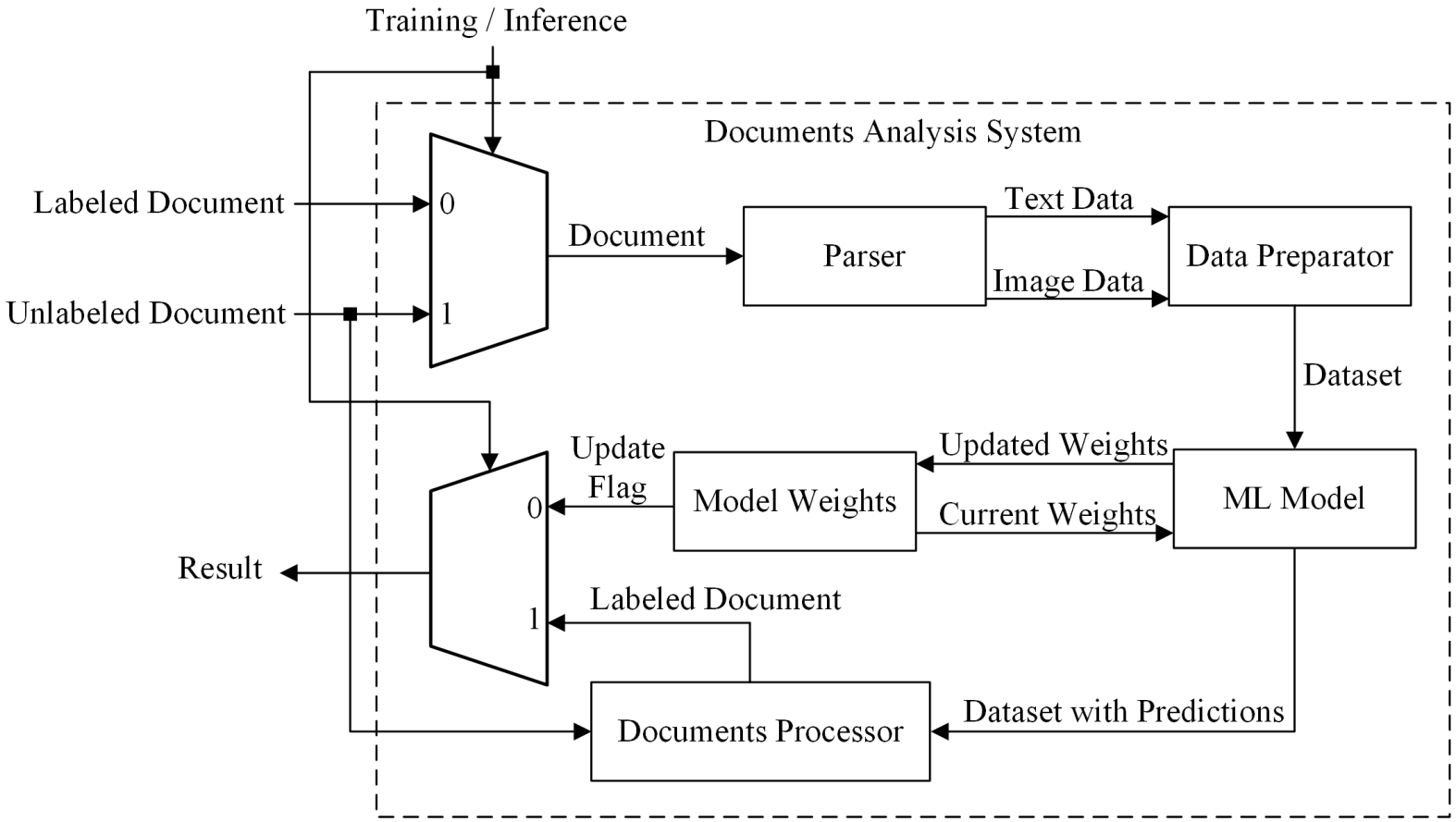
IMPLEMENTATION NOTE
Invalidate Flow Control

A Function may indicate that invalidate flow control is not required when one or more of the following is true:

1. The Function can handle invalidations at the maximum arrival rate of Invalidate Requests.
2. The Function will not or very rarely cause Link backpressure (performance loss is negligible).
3. The Function can fully buffer the maximum number of incoming invalidations without back pressuring the Link.

Using this markup as an input, an experienced engineer shows 20% better performance

Automated System



Implementation Libraries

PyPDF2

PyTorch

pandas

HUGGING FACE

NumPy

2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

SAN JOSE, CA, USA
MARCH 4-7, 2024

Conclusion

accellera SYSTEMS INITIATIVE

Conclusion

- Training dataset based on M-PHY Specification has been prepared
- LLM (GPT-2) has been trained on this dataset
- Automated system for technical documents analysis has been proposed
- The proposed solution gives a valuable inputs to improve the routine process on PCIe specification
- The approach can be extended in order to extract more information from spefications

Questions


Go raibh maith agat
Kiitos
Tak
Multumesc
Thank you
Ačiū
Merci
Gracias
Grazie
Danke
Grazzi
Děkuji
Dziękuję
Köszönöm
Hvala
Efcharistó
Dank u wel
Obrigado
TÄNAN
Ďakujem
Tack
Blagodarya