

The background is a dark, deep blue space filled with numerous glowing, translucent cubes. These cubes are arranged in a somewhat chaotic but structured pattern, with some appearing to be in motion or floating. Bright, starburst-like light rays emanate from several points, particularly from the right side, creating a sense of depth and energy. The overall aesthetic is futuristic and technological.

# Data Quality Metrics

(IN COLLABORATION WITH BMW GROUP)

Developers: Vladana Djakovic, Valari Pai, Ekaterina Shmaneva

Supervisors: Dr Maka Karalashvili (ext.), Prof. Dr Matthias Schubert (int.)

# CONTENT

I

## Introduction

II

## Theoretical aspects

(Summarization&classification tasks,  
Existing methods)

III

## Background

(Data processing & used model)

IV

## Implementation details

(Storyline, insights, results)

V

## Summary

(Future work, conclusion)

# MOTIVATION

When (or after) the car is produced, different defects occur. These defects are recorded and stored in the data source – the “Knowledge base” – that summarizes similar defects and assigns them to the prebuilt defect cluster. Each defect contains high amount of human written-text data, which makes analysis time-consuming and complicated

# OUR GOAL

To build a model that will process the human created text data of different length, create a summary of it, classify it based on the „sense“ of the generated summary and evaluate the quality of it

# Summarization

– a text that is produced from one or more texts, that contains a significant portion of the information in the original text(s).

If it was created with the computer, it is called **automatic summarization**.

Can be **abstractive** and **extractive**.



# Classification

– categorizing open-ended text into two or more predefined classes based on some rules or similarities between these texts.

Can be performed based on of the three approaches:

- **Rule-based systems**
- **ML-based systems**
- **Hybrid systems**





## Models, used only for summarization

(e.g. Sumy)



## Models, used only for classification

(e.g. Naive Bayes, SVMs)



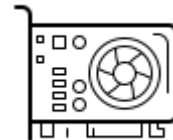
## Models, used for both tasks

(e.g. Gensim, CNNs, RNNs,  
BERT-based models,  
GPT models, XLNet, T5)





**Data access and  
security issues**



**Insufficient  
resources issues**





## **Data access and security issues**

(new open-source dataset should be found, that would match the original one)

# Data



## Amazon Product Review Dataset

Information



10 columns:

Structure

*ID, Product ID, User ID, Profile Name,  
Helpfulness Numerator, Helpfulness  
Denominator, Score, Time, Summary, Text*



568.427 reviews

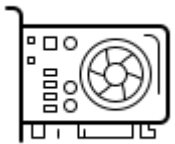
Content



2 columns kept: *Summary, Text*

Useful data





## **Lack of proper computational resources**

(lightweight models should be found to complete the task)



# OUR CHOICE: T5 model summarization

## **Encoder & Decoder blocks**

(decoder block helps model to create better summary)

## **The output is a text string**

(many other models have labels/spans as output  
→ improper output for summarization task)

## **Robust and extensible**

(weights are assigned more properly,  
the model can be easily modified to other tasks)





# OUR CHOICE: DistilBERT model classification

## **Small, fast, cheap**

(40% less parameter than BERT → 60% faster)

## **Distilled & transfer-learning adapted**

(mix of the distillation and transfer-learning  
→ Above 90% accuracy on classification)

## **Open-source & flexible**

(model available via HuggingFace,  
retains 97% of BERT performance)

# PROJECT TIMELINE



**Oct, 2021**

(Getting to know the supervisor, the project and the goal of it, searching for the data)



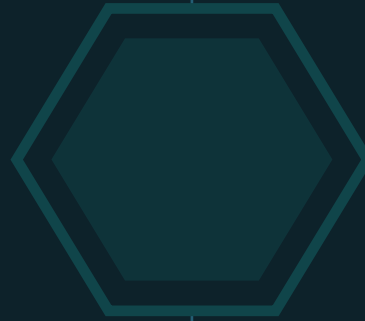
**Nov, 2021**

(Exploration of the dataset, metric extraction & processing ideas, building a data loader)



**Dec, 2021**

(Research on summarization techniques, exploring necessary packages)



**Jan, 2022**

(First-choice model research, baseline model building (RoBERTa), research on classification)

# PROJECT TIMELINE



**Feb, 2022**

(RoBERTa issue handling, parameter fine-tuning, classification implementation)



**Mar, 2022**

(Classification model issue handling, testing and parameter fine-tuning)



**Apr, 2022**

(Second-choice model research and implementation (Google T5 model))



**May, 2022**

(New model issue handling, parameter fine-tuning, documentation preparation)

# Note on summarization model change

## RoBERTa

vs.

## Google T5

Pre-training

5 datasets containing  
about 160GB of text

Multi-task un-&supervised  
tasks on 16 datasets

Pre-training time

1 day (1024xV100 GPUs,  
batch size 8k)

~12 hours (Titan RTX,  
batch size 8k)

Base of the model

BERT (bi-directional  
transformer model)

—

Parameter set

354M parameters  
(RoBERTa-large)

11B parameters  
(t5-11b)

Optimizer used

Adam  
(learning rate = 0.0006)

AdamW & AdaFactor  
(learning rate = 0.0003)

ROUGE Score  
(official paper)

F-measure (ROUGE-L)  
= 25.67

F-measure (ROUGE-L)  
= 38.35





# Implementation: Set Up

## Computational System

Google Colab Pro  
(up to 24GB RAM, K80, P100, T4 GPUs)

## Environment

Python ver. 3.8.5 and above

## Model & documentation



# Tokenization

- a way of separating a piece of text into smaller units called tokens. Either words, characters or subwords can be **tokens**.

Tokenizer breaks natural language text into chunks of information that can be considered discrete elements.

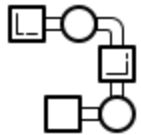
This immediately turns an unstructured string (text document) into a numerical data structure suitable for machine learning.



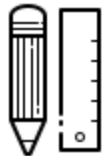
# Implementation: T5 Tokenizer & Optimizer



robust & in-built



uses SentencePiece



based on Unigram

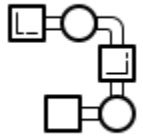


easy configurable



inherits main methods of 📖 PreTrainedTokenizerFast

# Implementation: DistilBERT Tokenizer



uses WordPiece algorithm



reachable via 🧠 Transformers



easy configurable



inherits most properties of 🧠 BERT Tokenizer



# Implementation: DistilBERT Classes

Initial text of the review

Generated summaries

Positive

*Reviews showing that customer was satisfied with the purchased product*

Positive

Negative

*Reviews showing that customer was not satisfied with the purchased product*

Negative



# Performance: Metrics

## Running time

shows the amount of time that was required to perform the training (only)

## Validation and training losses

describe the performance of the model, indicating how well it is fitting the training and the new data correspondingly

## ROUGE Score

compares automatically produced summary against reference (human-written) ones

## Accuracy

defines the number of correctly predicted data points out of all the data points

# Performance: T5

10k sample

vs.

100k sample

Running time                      approx 30 min

approx 3 hours

Train loss                                      2.481

2.849

Validation loss                                3.532

3.859

## ROUGE Scores

Rouge-1                                      0.266

0.143

recall

1.0

1.0

precision

0.421

0.250

f-measure

Rouge-L

0.266

0.143

recall

1.0

1.0

precision

f-measure

0.421

0.250

# Performance: Classification

	10k sample	vs.	100k sample
Running time	approx 36 min		approx 3.5 hours
Train loss	0.137		0.094
Accuracy	0.953		0.963
Validation loss	0.274		0.194
Validation accuracy	0.901		0.935
Classification error	0.1		0.1



# FUTURE

1. Model adaptation to the BMW data
2. Further summarization model fine-tuning to make the model more precise
3. Expanding the classification of the data (based on the information, that the summaries contain)

# CONCLUSIONS

I

## Why task is important

(Analysing the human-written defects is not easy and time consuming)

II

## What models exist

(*Summarization*: BERT (&variations), GPT, T5, CNN, RNN;  
*Classification*: Naive Bayes, SVM, summarization ones)

III

## What models were chosen

(*Summarization*: Google T5 model;  
*Classification*: DistilBERT)

IV

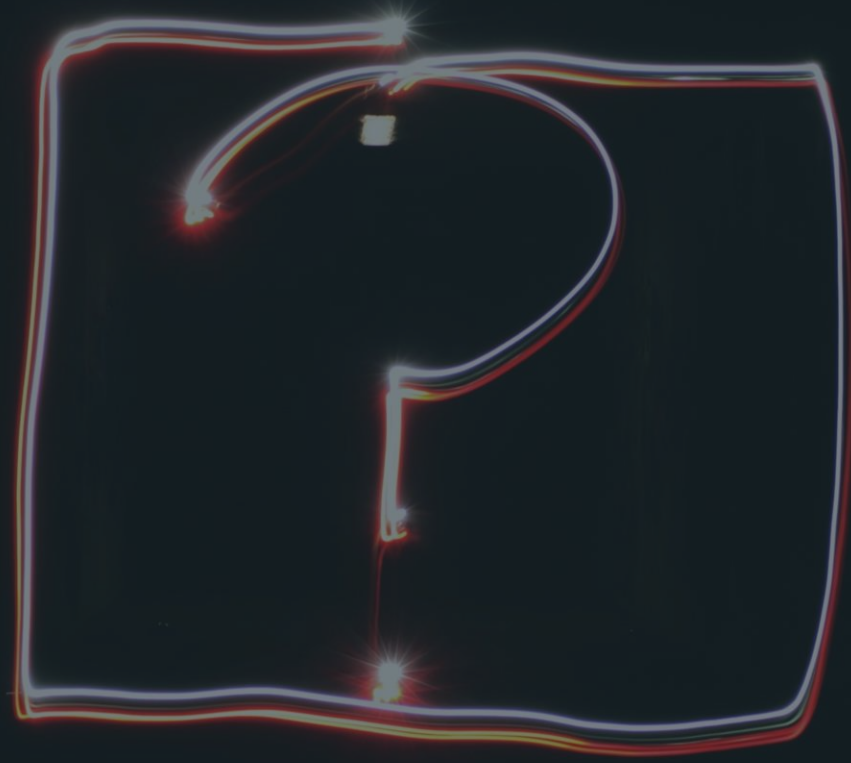
## Performance analysis

(*Summarization*: all of the n-grams in the generated summaries are present in the reference text;  
*Classification*: overall accuracy > 90%, error = 0.1)

V

## What else can be done

(*Summarization*: fine-tuning to increase the quality;  
*Classification*: expanding the number of classes)















THANK YOU

# References

---

## Literature:













-  Jacob Devlin and others. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding  
<http://arxiv.org/abs/1810.04805>
-  Aastha Singh. Evolving with BERT: Introduction to RoBERTa  
<https://medium.com/analytics-vidhya/evolving-with-bert-introduction-to-roberta-5174ec0e7c82>
-  Yinhan Liu and others. RoBERTa: A Robustly Optimized BERT Pretraining Approach  
<https://arxiv.org/abs/1907.11692>
-  Rohan Jagtap. RoBERTa: Robustly Optimized BERT-Pretraining Approach. DataSeries  
<https://medium.com/dataseries/roberta-robustly-optimized-bert-pretraining-approach-d033464bd946>
-  Anubhav. Step by Step Guide: Abstractive Text Summarization Using RoBERTa  
<https://anubhav20057.medium.com/step-by-step-guide-abstractive-text-summarization-using-roberta-e93978234a90>
-  Manmohan Singh. Summarize Reddit Comments using T5, BART, GPT-2, XLNet Models  
<https://towardsdatascience.com/summarize-reddit-comments-using-t5-bart-gpt-2-xlnet-models-a3e78a5ab944>
-  Sukanya Bag. Text Summarization using BERT, GPT2, XLNet  
<https://medium.com/analytics-vidhya/text-summarization-using-bert-gpt2-xlnet-5ee80608e961>
-  Summarization with GPT-3. KDnuggets. Section: Products and Services  
<https://www.kdnuggets.com/2022/04/packt-summarization-gpt3.html>
-  Priya Shree. The Journey of Open AI GPT models. Walmart Global Tech Blog  
<https://medium.com/walmartglobaltech/the-journey-of-open-ai-gpt-models-32d95b7b7fb2>
-  Maria Yao. 10 Leading Language Models For NLP In 2021  
<https://www.topbots.com/leading-nlp-language-models-2020/>
-  Qiurui Chen. T5: a detailed explanation  
<https://medium.com/analytics-vidhya/t5-a-detailed-explanation-a0ac9bc53e51>
-  Pedro Marques. Fine tuning a T5 text-classification model on colab. Pedrormarques  
<https://pedrormarques.wordpress.com/2021/10/21/fine-tuning-a-t5-text-classification-model-on-colab/>



# References

---







## Literature:

-  Mathew Alexander. Data to Text generation with T5; Building a simple yet advanced NLG model  
<https://towardsdatascience.com/data-to-text-generation-with-t5-building-a-simple-yet-advanced-nlg-model-b5cce5a6df45>
-  Abstractive Summarization Using Google's T5. Turbolab Technologies Blog. Section: Technology  
<https://turbolab.in/abstractive-summarization-using-googles-t5/>
-  Eduard Hovy. Text Summarization. The Oxford Handbook of Computational Linguistics  
<https://www.oxfordhandbooks.com/view/10.1093/oxfordhb/9780199276349.001.0001/oxfordhb-9780199276349-e-32>
-  Praveen Dubey. Understand Text Summarization and create your own summarizer in python  
<https://towardsdatascience.com/understand-text-summarization-and-create-your-own-summarizer-in-python-b26a9f09fc70>
-  Text Classification: What it is And Why it Matters  
<https://monkeylearn.com/text-classification/>
-  Shrivar Sheni. Text Summarization Approaches for NLP - Practical Guide with Generative Examples  
<https://www.machinelearningplus.com/nlp/text-summarization-approaches-nlp-example/>
-  Susan Li. Multi-Class Text Classification with Doc2Vec & Logistic Regression  
<https://towardsdatascience.com/multi-class-text-classification-with-doc2vec-logistic-regression-9da9947b43f4>
-  In'és Rold'os. Go-to Guide for Text Classification with Machine Learning. MonkeyLearn Blog. Section: Text Classification  
<https://monkeylearn.com/blog/text-classification-machine-learning/>
-  Leo Laugier. Extractive Document Summarization Using Convolutional Neural Networks Reimplementation  
<https://www.semanticscholar.org/paper/Extractive-Document-Summarization-Using-Neural-Laugier/ed0f189bbbbbccceefb41ccb36e1c5b62bc36d2fb>
-  Christian Heumann and others. LMU Course: Deep Learning for Natural Language Processing  
<https://moodle.lmu.de/course/view.php?id=17645>
-  Zhenzhong Lan and others. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations  
<https://arxiv.org/abs/1909.11942>
-  Victor Sanh and others. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter  
<https://arxiv.org/abs/1910.01108>


# References

---

## Literature:

-  Zihang Dai. XLNet: Generalized Autoregressive Pretraining for Language Understanding  
<https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1606199&dswid=-4773>
-  What is a Data Steward? — Experian Business  
<https://www.experian.co.uk/business/glossary/data-steward/>
-  SAS-institute. What is a data scientist?  
[https://www.sas.com/en\\_us/insights/analytics/what-is-a-data-scientist.html](https://www.sas.com/en_us/insights/analytics/what-is-a-data-scientist.html)
-  Rachel Draelos. Best Use of Train/Val/Test Splits, with Tips for Medical Data  
<https://glassboxmedicine.com/2019/09/15/best-use-of-train-val-test-splits-with-tips-for-medical-data/>
-  Andrew Fogarty. Summarization: T5  
[http://seekinginference.com/applied\\_nlp/T5.html#rouge-metrics](http://seekinginference.com/applied_nlp/T5.html#rouge-metrics)
-  Kavita Ganesan. An intro to ROUGE, and how to use it to evaluate summaries  
<https://www.freecodecamp.org/news/what-is-rouge-and-how-it-works-for-evaluation-of-summaries-e059fb8ac840/>

## Data:

-  Amazon. Amazon Product data  
<https://jmcauley.ucsd.edu/data/amazon/>

## Imagery:

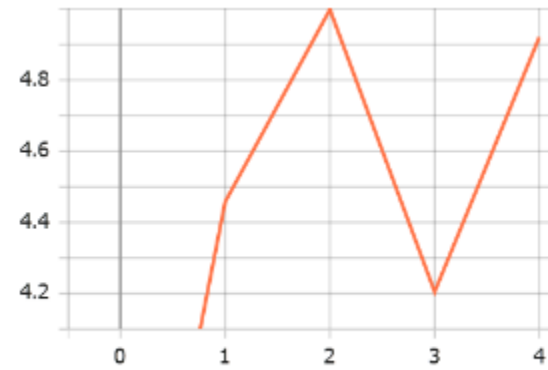
-  unsplash.com
-  pinterest.de
-  behance.net

## Graphics:

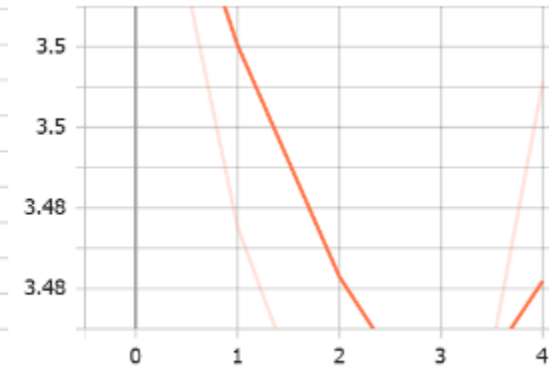
-  icons8.com

# Performance analysis: T5 Graphs

10k sample

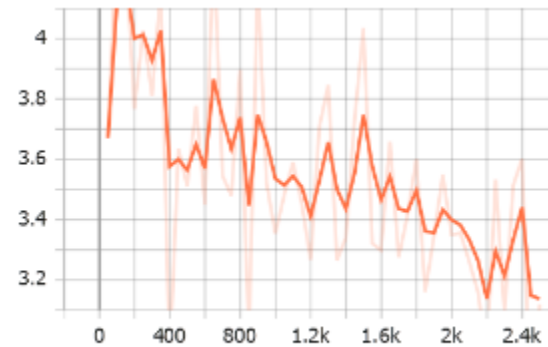


(a) *Train loss calculated epoch-wise*

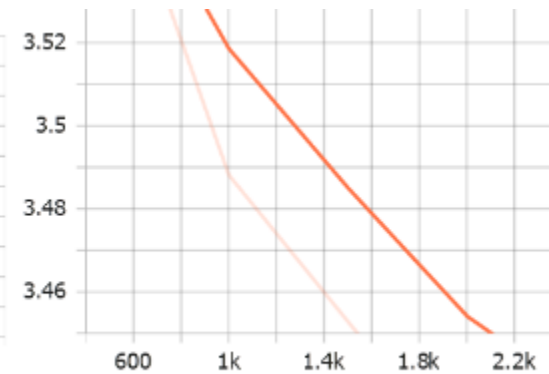


(b) *Validation loss calculated epoch-wise*

100k sample



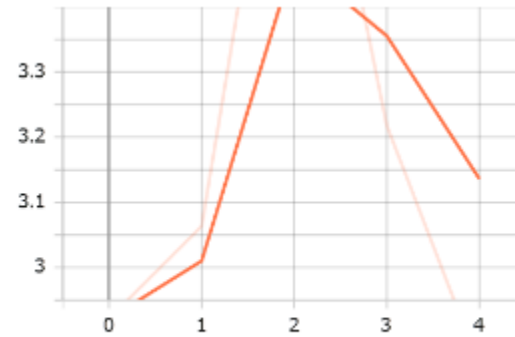
(a) *Train loss calculated batch-wise*



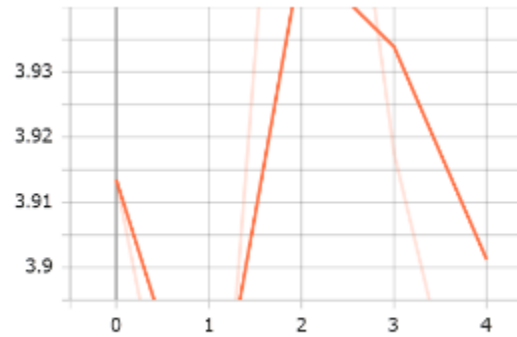
(b) *Validation loss calculated batch-wise*

# Performance analysis: T5 Graphs

10k sample

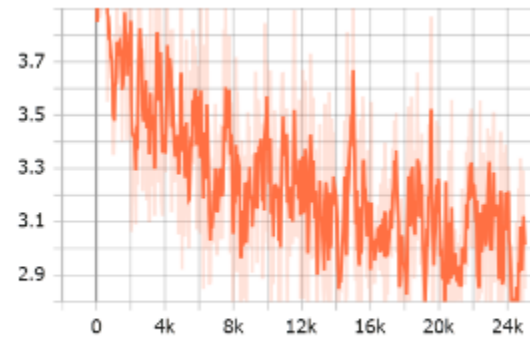


(a) *Train loss calculated epoch-wise*

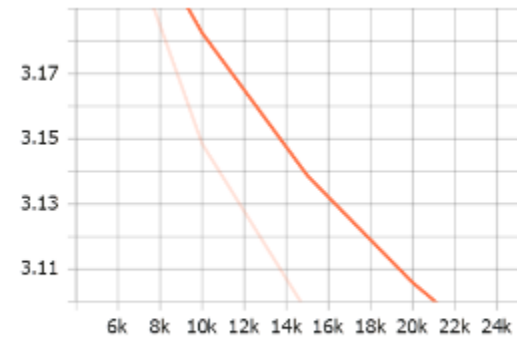


(b) *Validation loss calculated epoch-wise*

100k sample



(a) *Train loss calculated batch-wise*



(b) *Validation loss calculated batch-wise*