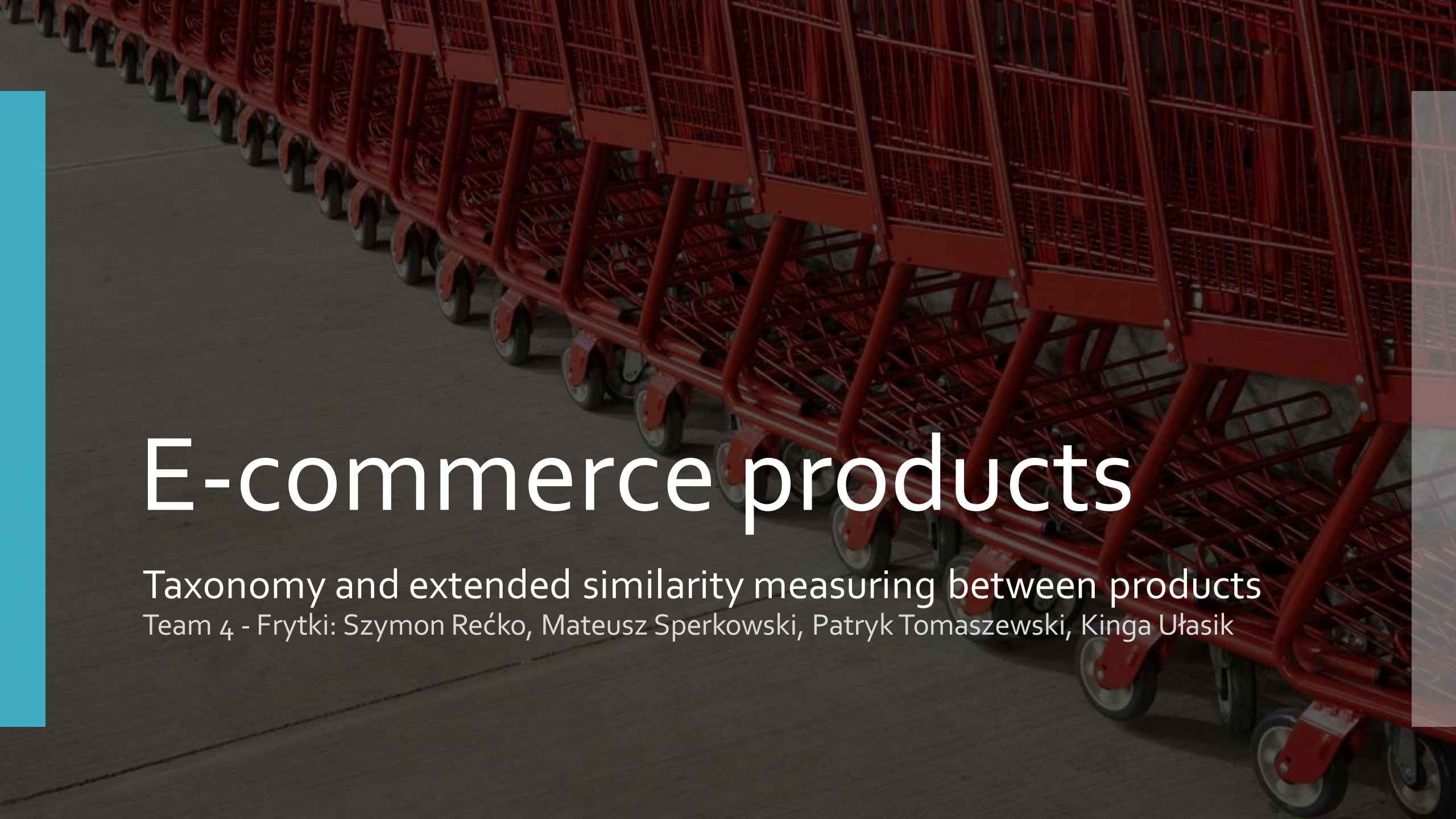


E-commerce products

Taxonomy and extended similarity measuring between products

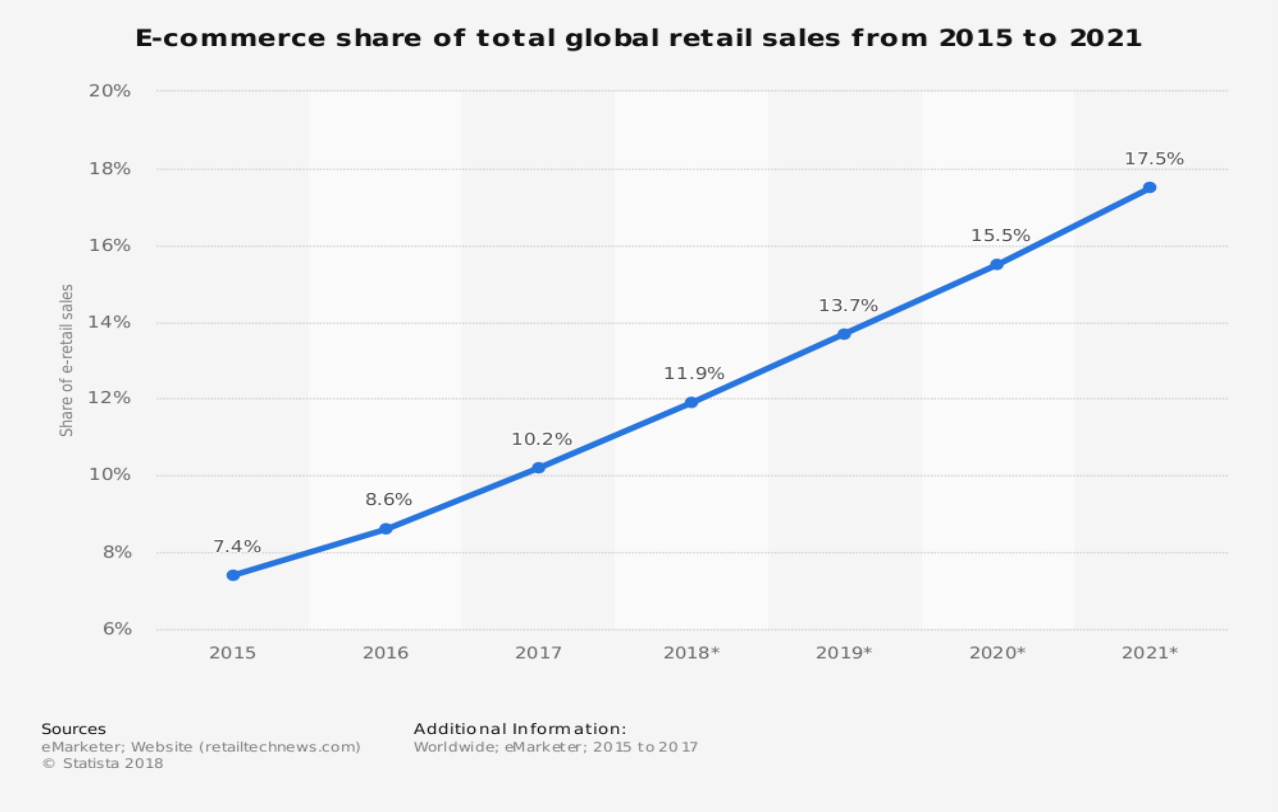
Team 4 - Frytki: Szymon Rećko, Mateusz Sperkowski, Patryk Tomaszewski, Kinga Ułasik

Presentation Plan

- Introduction
- Project Goals
- Datasets
- Experiments flow
- Results

E-commerce

- **E-commerce** is the activity of electronically buying or selling products on online services or over the Internet.



- **Recommendation systems** play a crucial part in increasing the companies business by assisting customers in finding what they need/want

High Level Final Project Goals

- Extracting crucial information from descriptions and titles using LLM's
- Automatic methods for measuring similarity between products
- Incorporating taxonomy into the measurements

Low Level Final Project Goals

Products data processing

- Exploratory Data Analysis
- Seperate the pairs of products and augment their text representations with LLM's.

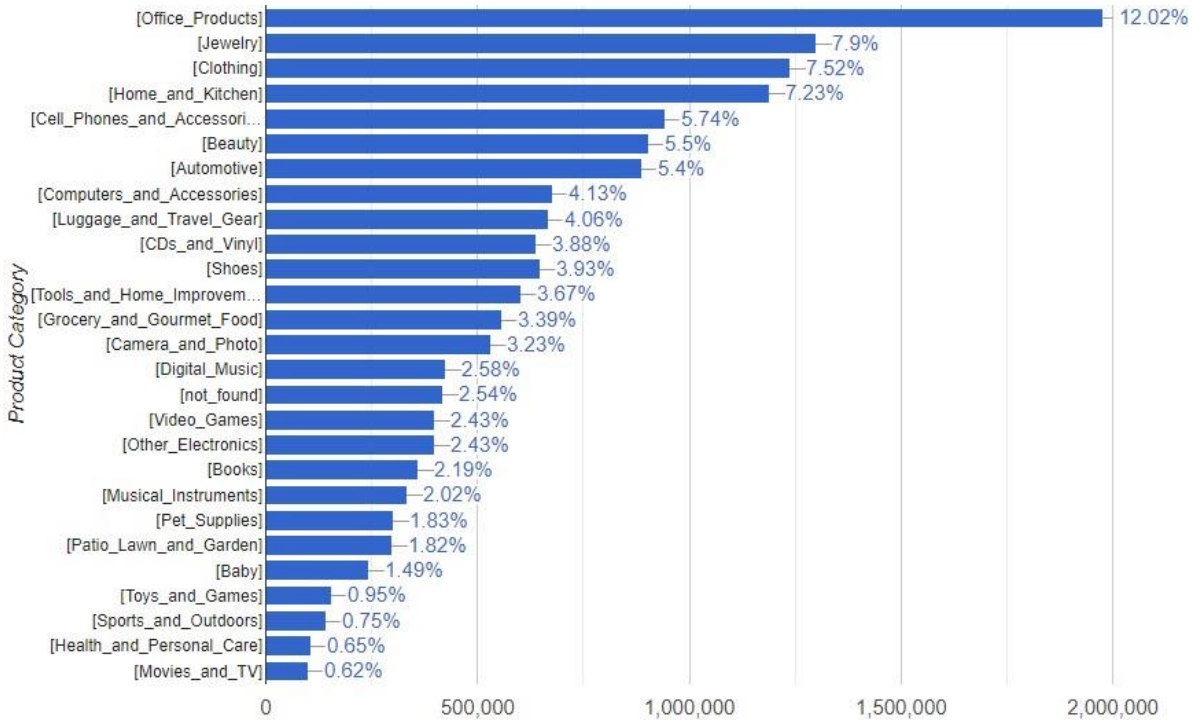
Implementation of modified product similarity pipeline

- Fine-tune BERT-based model (BERT, DistilBERT) and compare with non fine-tuned models (BERT, RoBERTa, DistilBERT).
- Experiment with loss/distance functions.
- Performance tests and evaluation.

Dataset

"Web Data Commons - Training Dataset and Gold Standard for Large-Scale Product Matching"

- 16 million English-language offers sourced from a wide array of 79 thousand websites.
- Includes product categorization based on Amazon product data and TF-IDF scores for 26 product categories.
- Each offer was assigned to one of 26 categories

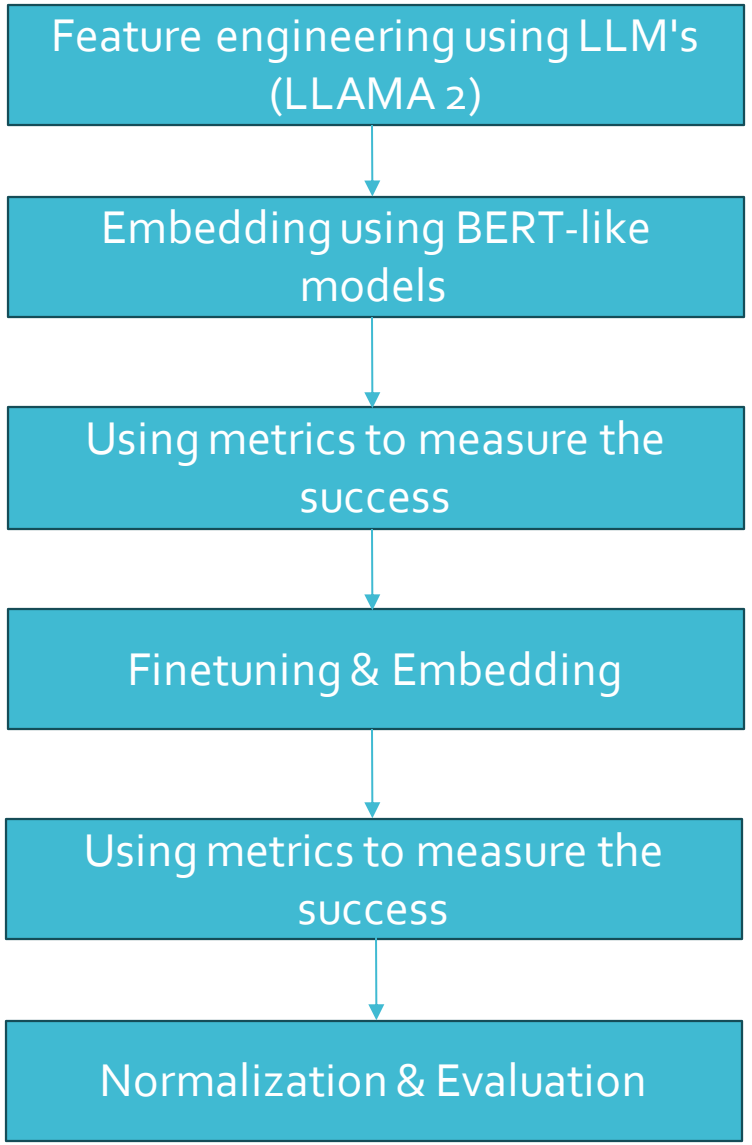


Dataset

GOLD STANDARD

Category	# positive pairs	# negative pairs	% title	% description	% brand	% price	% specTableContent
Computers	300	800	100	82	42	11	22
Cameras	300	800	100	73	25	3	7
Watches	300	800	100	71	15	1	7
Shoes	300	800	100	70	8	1	2
All	**1200**	**3200**	**100**	**74**	**23**	**4**	**10**

Experiments flow



Feature engineering using LLM's (LLAMA 2)

- Impossible to self host
- Hugchat – free alternative to chat-gpt
- Uses LLama 2 and is hosted by Huggingface
- Prompt Engineering
- Due to unsatisfactory LLM results and high computational time the idea to use Polish product matching dataset had to be dropped.

Prompt Engineering

"Given a product title and description, generate a meaningful text representation that captures the essence of the product for effective similarity search. Consider relevant features, attributes, and contextual information to ensure the generated representation reflects the product's unique characteristics, allowing for accurate comparisons in a similarity search algorithm. Do not answer, just create a representation.

TEXT TO REPRESENT:

<product title>

<product description>"

LLM Feature Extraction output

"Intel Xeon 5130 2GHz 4MB L2 Processor – HP
* Intel Xeon 5000 sequence
* 2 GHz processor frequency
* LGA 771 socket J
* 65nm 64-bit technology
* 291M transistors
* 143mm processing die size
* Thermal Design Power (TDP): 65W
* Supports Intel Virtualization Technology (VT)
* Enhanced Intel SpeedStep Technology
* Execute Disable Bit
* Idle States: C0, C1, C2...
* On-die digital thermal sensor
* Protective thermal management features"

BERT-based similarity learning for product matching

Janusz Tracz[1], **Piotr Wójcik**[1], **Kalina Jasinska-Kobus**[1, 2],
Riccardo Belluzzo[1], **Robert Mroczkowski**[1], **Ireneusz Gawlik**[1, 3]
[1] ML Research at Allegro.pl
[2] Poznan University of Technology
[3] AGH University of Science and Technology
{janusz.tracz,piotr.wojcik,kalina.kobus,riccardo.belluzzo,
robert.mroczkowski,ireneusz.gawlik}@allegro.pl

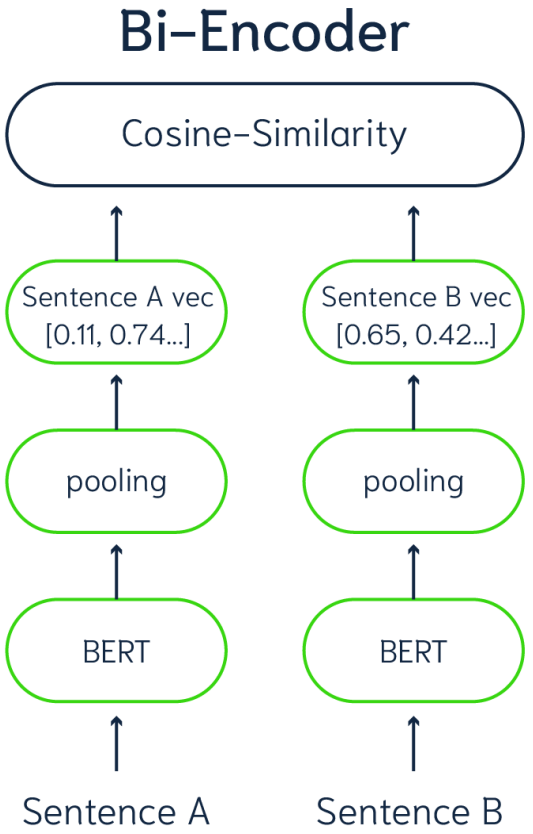
BERT-based similarity learning for product matching

Bi-Encoder architecture

- Products are represented as text
- A previously trained transformed is applied
- A distance between embeddings is calculated as similarity between instances

Used transformer: BERT

Used distance: cosine distance



BERT-based similarity learning for product matching

Similarity learning with triplet loss objective

Training data consists of triples in the form of

$$(o, p^+, p^-)$$

with the elements being the anchor, a matching product, and a non-matching product.

Then, the transformer is adjusted to minimize the following loss function:

$$\mathcal{L}(o, p^+, p^-) = max(0, m + d(\mathcal{E}_\theta(o), \mathcal{E}_\phi(p^+)) - d(\mathcal{E}_\theta(o), \mathcal{E}_\phi(p^-)))$$

Example

Left Item	Right Item	Similarity level	Similarity score	Execution time
Red apple	Green apple	Same fruit	0.99230	36.06ms
Red apple	Lemon	Both fruit	0.97707	34.18ms
Red apple	Brick	Both small objects	0.96443	34.49ms
Red apple	Warsaw University of Technology	Not similar	0.93574	35.39ms

Metrics Ideas and tests

Custom loss

$$Loss(a,(a',b,c)) = LeakyReLU(\cos(a,b) - \cos(a,a'))$$

$$+LeakyReLU(\cos(a,c) - \cos(a,b))$$

Kendall Tau Distance

$$K_d(\tau_1,\tau_2) = |\{(i,j) : i < j, [\tau_1(i) < \tau_1(j) \wedge \tau_2(i) > \tau_2(j)] \vee [\tau_1(i) > \tau_1(j) \wedge \tau_2(i) < \tau_2(j)]\}|.$$

$$METRIC(a, a', b, c, d) = \alpha KDT$$

$$+MSE([\cos(a, a'), \cos(a, b), \cos(a, c), \cos(a, d)], [1, 0.66, 0.33, 0])$$

,where α is a weight of Kendall Tau distance

Final Metric

Example values for metric

- [1, 0.75, 0.5, 0.25, 0] --> 1.0
- [1, 0.75, 0.9, 0.25, 0] --> 0.934
- [1, 0.999, 0.998, 0.997, 0.996] --> 0.814
- [1, 0, 0, 0, 0] --> 0.6125
- [1, 1, 1, 1, 1] --> 0.3125
- [0, 0.25, 0.5, 0.75, 1] --> 0.25

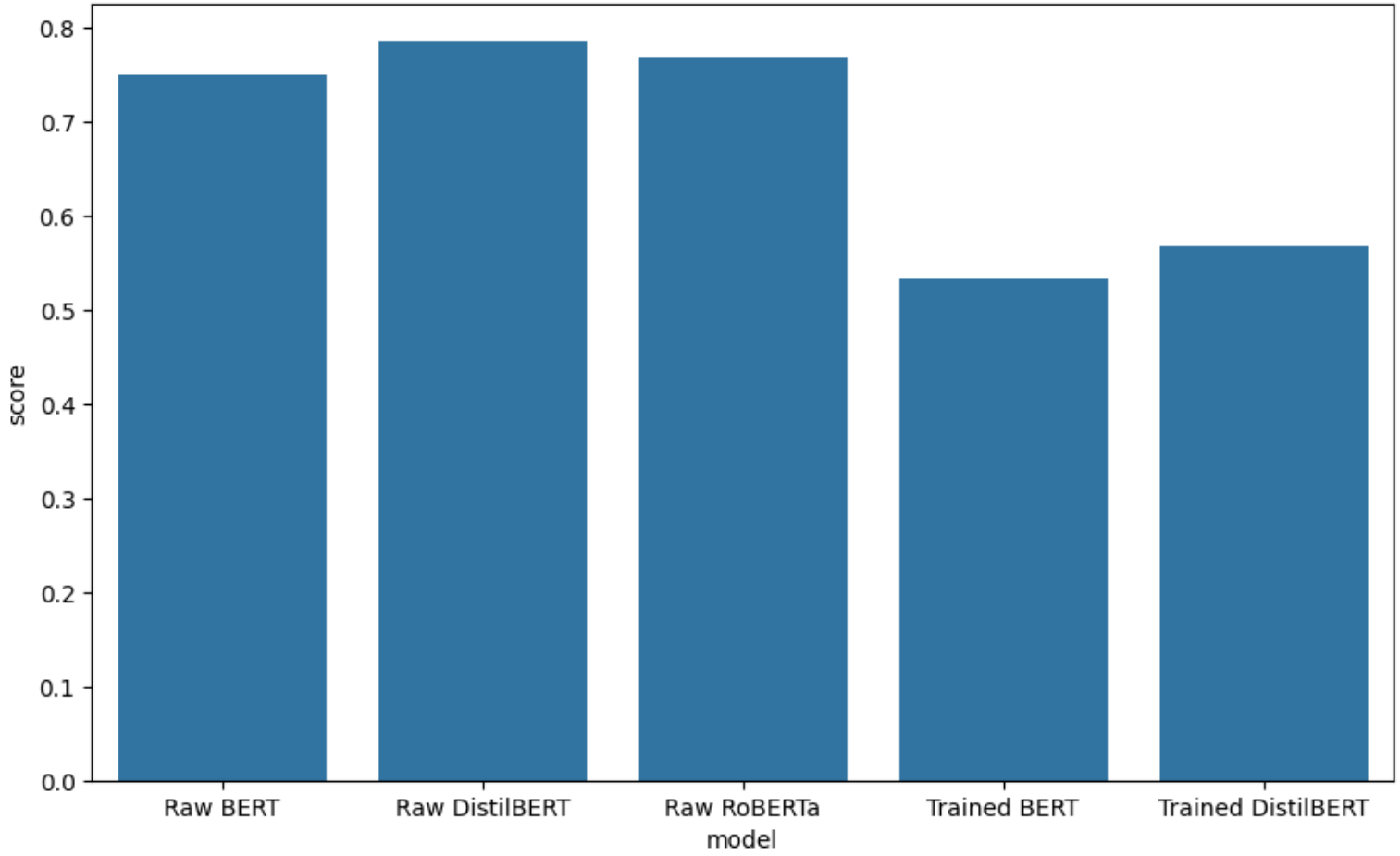
Experiment setup

- Finetuning of a pretrained model
- Training on the golden standard with the triplet loss

$$\mathcal{L}(o, p^+, p^-) = max(0, m + d(\mathcal{E}_\theta(o), \mathcal{E}_\phi(p^+)) - d(\mathcal{E}_\theta(o), \mathcal{E}_\phi(p^-)))$$

- Evaluation done on pentalets with our custom metric
- Three models analysed: BERT, DistilBERT, RoBERTa

Experiment results



Experiment results

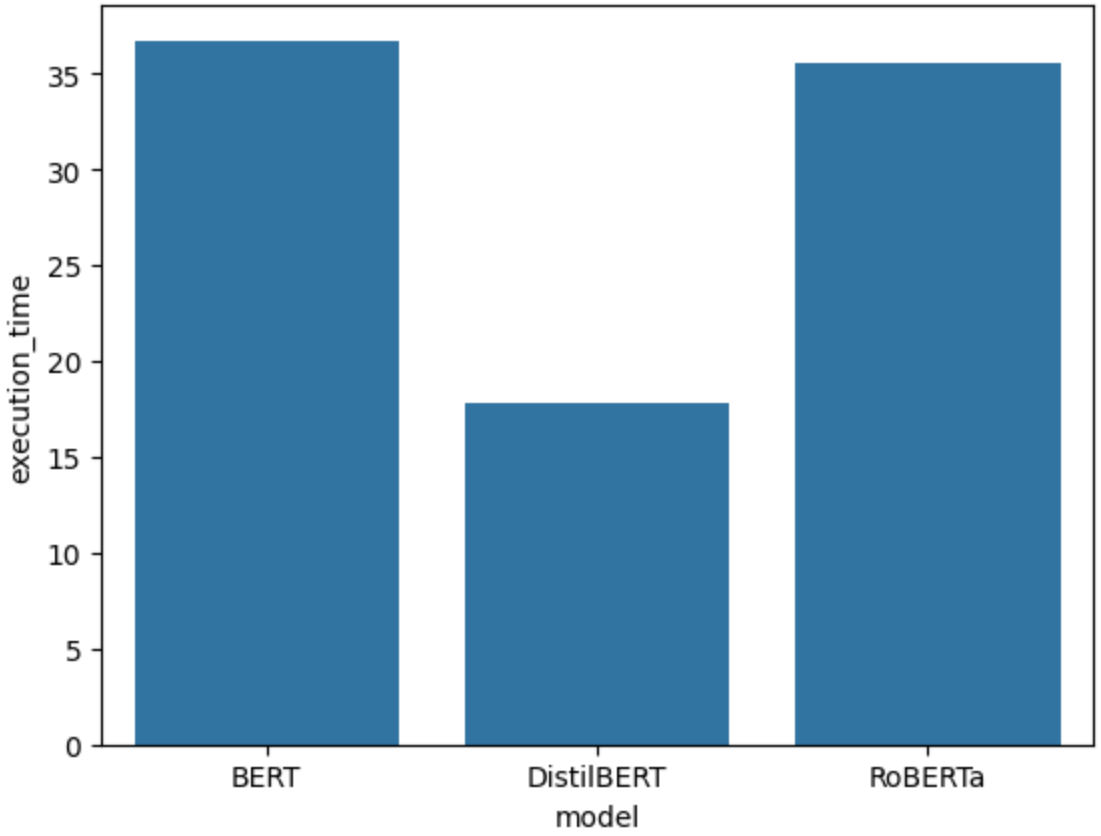
Without training

[1.0000, 0.9597, 0.9527, 0.9481]

With training

[0.9417, 0.9588, 0.9497, 0.9528]

Benchmarking results



Contributions

- Introduction of attribute extraction method with LLM
- Definition of a novel metric for measuring hierarchical similarity
- Tests of triplet loss finetuning on bert-like models
- Comparative study for BERT, DistilBERT, and RoBERTa

Issues

- We couldn't set up the dataset nor the LLM on eden (problems with the available ram/drivers/time).
- Inference time of self hosted LLM is extremely long.
- Inference time of publicly hosted LLM is very long.
- No good LLM for this task.
- API limitations run out fast.
- Free GPU for training run out fast.

Project 2 Possible Ideas

- Transfer Learning to speed up embedding process
- Embedding Analysis
- Model modification for better performance
- Alteration of the loss function to better match the hierarchical nature of the task
- Find new ways to extract crucial information without LLM's
- Scaling the metric

Thank you for attention

Feel free to ask questions