E-commerce products similarity
Project Report for NLP Course, Winter 2023

S. Rećko
WUT
01151399@pw.edu.pl

M. Sperkowski
WUT
01151430@pw.edu.pl

P. Tomaszewski
WUT
01151442@pw.edu.pl

K. Ułasik
WUT
01151444@pw.edu.pl

supervisor: A. Wróblewska
WUT
anna.wroblewska1@pw.edu.pl

Abstract

The rapid growth of e-commerce, fueled by shifts in consumer behavior and accentuated by the COVID-19 pandemic, has underscored the need for efficient recommendation systems. This article presents a novel framework that leverages machine learning and Natural Language Processing techniques to develop an automated and comprehensive measurement of similarity for recommendation systems with leveraging of a taxonomy of similarity, especially in the e-commerce field. The focus lies on accurately measuring product similarity based on various attributes, including taxonomy, textual descriptions, and titles. The framework, designed for real-time calculations, aims to enhance e-commerce search functionalities and improve the overall user experience. The project employs state-of-the-art language models such as BERT, DistilBERT, and RoBERTa, incorporating a custom metric to evaluate the performance of these models. The experiments include training procedures with a Triple Loss function, and the results demonstrate the behavior of the evaluation function.

1 Introduction

The rise of e-commerce has been an unprecedented change in the sales market. Over the past years, e-commerce has witnessed exponential growth, accelerated further by global shifts in consumer behavior, especially the increased adoption of online shopping. The convenience, accessibility, and wide array of online products have transformed how people shop. The COVID-19 pandemic further expedited this shift, with many consumers transitioning to online platforms for everyday needs. As e-commerce continues to expand, the need for efficient recommendation systems becomes even more critical, (Zhang et al., 2019). With an ever-growing number of available products, these systems play a pivotal role in helping consumers navigate the vast array of choices, making their shopping experiences more personalized, streamlined, and enjoyable (Bobadilla et al., 2013).

The overarching goal of this project is a development of an innovative and robust framework leveraging machine learning and Natural Language Processing (NLP) techniques to create an automated and comprehensive measurement of similarity for recommendation systems that accurately captures the semantics and nuances of product relationships, distinguishing between identical, closely related (such as variations within the same product line differing in specifications like memory sizes) and entirely distinct products. The architecture can measure the similarity considering the different levels of dissimilarity between products within e-commerce platforms. This research aims to achieve an advanced level of accuracy in defining and establishing similarity metrics between products based on various attributes, including taxonomy (categories), textual descriptions and titles. To be practical for the end users, calculations must be performed quickly. Therefore, the calculations should work in real-time, so a limit of at least 0.01 seconds (10ms) per pair of products is set. This is considered an important requirement for the project.

Furthermore, the project will focus on technique extracting key information from item descriptions and titles using large language models. The innovation lies in their capacity to comprehend context, nuances, and relationships within text, enabling more accurate and context-aware extraction of key information. These generated attributes will supplement existing data, enriching the product information available on e-commerce platforms and possibly creating a better representation of the products.

Ultimately, the scientific aim is to significantly enhance e-commerce search functionalities, facilitate the simple yet efficient introduction of new products into online marketplaces, by creating simple processing pipeline, and provide users with access to a comprehensive range of available products and complementary items. This research endeavour will contribute to advancing the capabilities of e-commerce platforms by refining the accuracy and depth of product recommendations, thereby improving the overall user experience in online product search and comparison.

1.1 Research questions and contributions

In e-commerce, the sheer volume and diversity of products pose a significant challenge in accurately measuring their similarity across multiple dimensions. Typically items have some forms of a title, product name and description, which are often extended with more detailed descriptions, categories and tags, photos, tables with specifications of the model, seller/producer and many more. When looking at the title or the products name it is often hard to say what is it exactly, without looking at the photos or the descriptions. Current methodologies for comparing and categorizing products, such as feature-based comparisons and hierarchical clustering, often struggle to capture the full picture. For instance, comparing smartphones solely based on processor speed and RAM wouldn't account for design, camera quality, or user experience. This can lead to misleading results and hinder informed decision-making. Perhaps incorporating user reviews and sentiment analysis alongside traditional methods could offer a more holistic approach. This leads to suboptimal search experiences for users and hampers the efficiency of introducing new products into e-commerce platforms. An automated and precise system to measure product similarity, considering varying attributes and features, is crucial for enhancing product search accuracy and user satisfaction.

Research Questions and Hypothesis:

Q: Is there a measure that can incorporate the taxonomy of products in similarity measurements, and if so, can it be incorporated into the loss function for fine-tuning transformer models?

H: Such measure can be found; however, the incorporation may prove difficult. It could require much more inputs to the model, significantly increasing the processing time.

Q: Can we leverage the Large Language Model complexity to augment the data and achieve a better representation of the products?

H: Generative power and generalisation of LLMs can be used to augment the dataset, as they can solve many real-world problems. Issues might arise concerning the API limits for free LLMs or the limited size of these models.

Q: Can the real-time performance of the pair similarity measurement be achieved using transformer models? (Reaching 10 ms per pair comparison)

H: Such performance is achievable; however, it might require novel approaches to minimize the inference time. The complications might come from the limited computational power available in the project.

Q: Can the approach from one dataset be extended to another, possibly in another language, and can it achieve similar results?

H: The architecture should be carefully developed to solve a similar task with minimal changes. However, the performance of the models is heavily dependent on the amount of available data, which is typically harder to come by in languages other than English.

The contributions of our work are as follows:

- Definition and implementation of pipeline for calculating similarity measure
- Prompt engineering for extracting attributes from products
- Definition of a metric for determining quality of a multi-hierarchical similarity measure
- Ablation study of quality of simialrity measure for three different BERT-like models
- Ablation study of execution time for three different BERT-like models

1.2 Report structure

In Section 2, we describe the scientific literature related to our project. We focus on state-of-the-art NLP and text embeddings, e-commerce recommendation systems, similarity measures, and large language models. Furthermore, the closely related works and the datasets used in the research are described in detail. Section 3 describes the research methodology adopted in this project, especially the techniques and tools for both research and result analysis. Section 4 delineates the experiments on the chosen WDC datasets, including the exploratory data analysis, data augmentation using LLMs and finetuning of the transformer model. In Section 5, a discussion of our results compared to the literature is assembled. Section 6 presents the project conclusions, summing up what has been done and proposing future works for this project and the domain.

2 Related works

Machine learning approaches for extracting meaningful attributes from product descriptions include NLP techniques, word embeddings, and Named Entity Recognition (NER). These attributes can be integrated into the similarity measurement process by converting them into product feature vectors. Utilizing metrics such as cosine similarity or Euclidean distance on these vectors allows for an effective quantification of product similarity, enhancing the overall recommendation system. It is done by upselling products to clients (suggesting similar but higher priced versions) or increasing sales by finding similar item but better suited to the particular items. By enhancing the users experience we increase the chances that they will return to the particular e-commerce site.

Siamese Neural Networks (Koch et al., 2015), Graph Neural Networks (Scarselli et al., 2008) (GNN), and Transformer models (Vaswani et al., 2017) like BERT (Devlin et al., 2019) excel in capturing semantics and nuanced relationships between products. Siamese networks are adept at understanding subtle differences. GNNs model complex graph-like dependencies due to their design, however that also requires specific representation of the dataset. Transformers provide contextualized embeddings, collectively offering a robust framework for differentiating between identical, slightly different, and entirely distinct items.

Integrating generated product attributes into existing e-commerce platforms can be achieved by developing an attribute-based search functionality, enhancing recommendation systems, and optimizing the introduction of new products.

Various strategies can be employed to reach minimal similarity comparison time between to products (a pair) while utilizing state-of-the-art (SOTA) models with complex architectures. Techniques like model quantization (Polino et al., 2018) and pruning (Liu et al., 2018) help reduce the computational load, while hardware acceleration using specialized processors like GPUs or TPUs speeds up inference. Additionally, caching and batch processing can be implemented to precompute certain calculations and perform parallelized comparisons, ensuring efficient and real-time processing without compromising the sophistication of the underlying models. Distilling the knowledge of a bigger model to a smaller one by enforcing similar outputs on the training data is another method commonly used for creating smaller networks (Gou et al., 2021).

In recent years, Large Language Models (LLMs) have emerged as transformative tools across various domains, showcasing their remarkable versatility and utility. These models, such as GPT-3 (Brown et al., 2020), have demonstrated an unprecedented ability to understand and generate human-like text, enabling advancements in natural language processing, text generation, and information retrieval. In artificial intelligence, LLMs have been pivotal in enhancing chatbot capabilities, language translation, and content creation. Moreover, they have proven instrumental in automating mundane tasks, facilitating more efficient data analysis, and even contributing to the development of novel applications in healthcare (Thirunavukarasu et al., 2023), finance (Wu et al., 2023), and education (Kasneci et al., 2023). The extensive capabilities of Large Language Models (Wei et al., 2022) underscore their potential to revolutionize how we interact with technology, opening up new frontiers for innovation and problem-solving across diverse fields.

One of possible usages of LLMs for deep learning is their usage in feature engineering. In (He et al., 2023) authors prompted the models to perform a classification task and explain its reasoning. This reasoning is then used as features, enhancing the dataset, and in this way the reach SOTA scores in their task while reducing the required training time. A different approach was used in (Hollmann et al., 2023), whose method is called CAAFE. The model is given a task of finding newly engineered features and writing the required code for the transformation, together with explanations why it should be helpful for the task. In the input prompt a description of the dataset is given from a tabular dataset. Different prompts are used in (McInerney et al., 2023), where expert-crafted queries are used to generate features from health records. These noisy feature are then used in a linear classifier, which trained weights align with the clinical expert expectations. However due to the size of the models, LLMs are highly nondeterministic, even with specified parameters for randomness. Therefore two same prompts can differ widely and conclusions should be drawn carefully ((Ouyang et al., 2023)). Their performance should be measured and evaluated with care, one of the methods for LLM evaluation is introduced in (Banerjee et al., 2023).

2.1 Available Datasets

In recent years, entity resolution has shifted towards deep learning-based matching methods, necessitating large training data. Traditional benchmark datasets containing explicit and implicit feedback from users often prove inadequate for evaluating these methods due to their limited size and source diversity. The ”Web Data Commons” (WDC, (Peeters et al., 2023)) dataset (Web Data Commons - Training Dataset and Gold Standard for Large-Scale Product Matching) addresses these challenges by offering a substantial volume of data, including 16 million English-language offers, sourced from 79 thousand websites. This diversity and scale make it an ideal choice for assessing deep learning-based matches and improving their evaluation and comparison. The dataset includes categorization using distant supervision from Amazon product data. Distant Supervision is the process of specifying the concept which the individual words of a passage, usually a sentence, are trying to convey. Lexica containing terms and their TF-IDF scores for 26 product categories (see Figure 1) were created using publicly available Amazon product reviews and metadata. Offer entity most often consists of name and description and sometimes also a brand, price and specs. Each offer in the dataset is assigned the product category whose terms maximize the sum of TF-IDF scores of the overlapping terms. In cases with minimal overlap, the offer is categorized as ”not found”. We exclusively utilized the ”Gold Standard” to train our product matching method. The gold standard, derived from the English product data corpus, comprises a set of 1,100 pairs of offers from each of the four product categories: Computers Accessories, Cameras & Photos, Watches, and Shoes. For each product, the gold standard includes two matching pairs of offers (positives) and five or six non-matching pairs of offers (negatives).

The Polish dataset for product matching created was created by the authors of (Michał Mozdzonek, 2022), following the approach of the WDC dataset. Having two such datasets could allow for the analysis of both multilingual approaches and the generalization of models to other languages. The dataset was derived from popular Polish stores, involving data collection, subsequent cleaning, and transformation into a tabular format compatible with the WDC dataset.

3 Our approach & research methodology

In this section, we first name the dataset used in the project, followed by a description of methodologies used in our project. The last subsection contains an explanation of our chosen approach.

To create our solution, we have selected the WDC dataset as the primary dataset for our project. More so, feature engineering using LLMs in the experiments on this data only returns English descriptions, automatically translating the Polish text. Therefore, adding these sentences would mix two languages in a single input, which we believed to be a further issue. Thus, the Polish dataset is left as possible future work for this project.

3.1 Related Methodology

Our solution is based on the pipeline in (Tracz et al., 2020). In this article, the authors propose using a transformer architecture, specifically a fine-tuned version of BERT, to embed and then compare a pair of products. As the pre-embedding product representation, a concatenation of the title, attribute values, and units was used. They used a triplet loss objective with the cosine distance for fine-tuning. Additionally, various batch construction strategies for selecting the triplets used in training were analyzed.

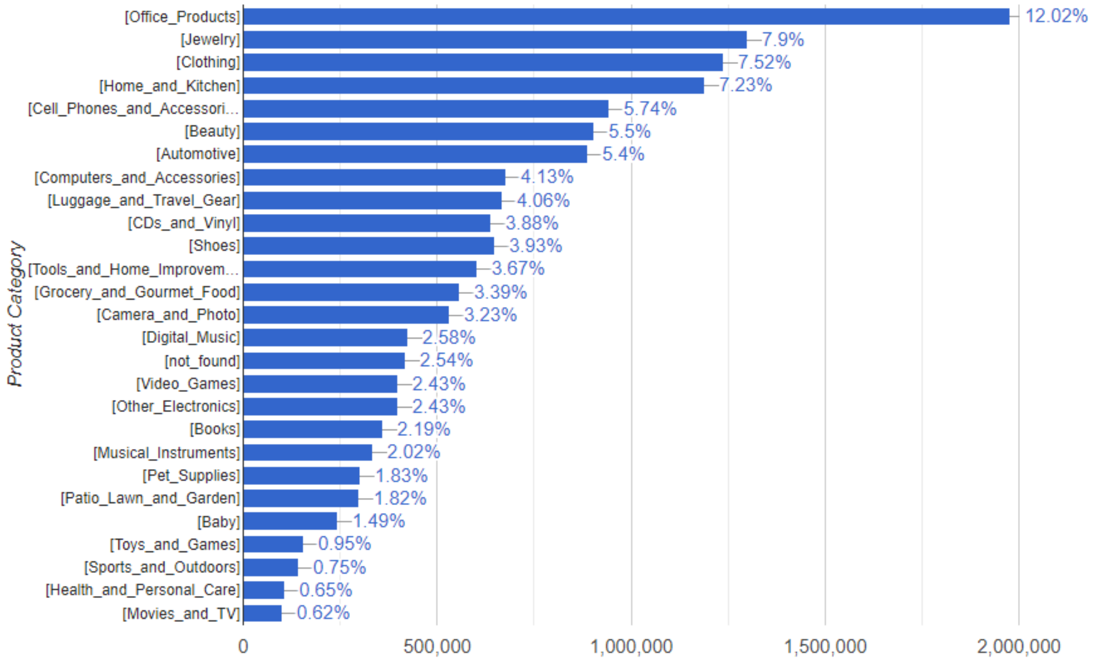

Figure 1: Distribution of offer entities per category in the WDC dataset. Source:

A similar methodology has been described in (Peeters et al., 2020), where a cross-encoder structure was used instead of a bi-encoder. Additionally, textual representation has been replaced by a concatenation of product brand, title, truncated description and truncated specification table.

An alternative approach was presented in (Peeters and Bizer, 2023), where the similarity score was generated with an LLM under proper prompt construction. However, due to performance limitations present in our problem, this solution was disregarded. This led us to the idea of using LLM's as a feature engineering tool, which is described further in the report.

Authors of the (Michał Mozdzonek, 2022) article focused on the Product matching problem and the idea of using transfer learning for data in different languages. Data preparation for the model involved selecting the title column and concatenating it with token marking the beginning and end of the text. This resulted in a single input string for the model, which was then tokenized using a pre-trained model-specific tokenizer.
For the our work's experimental part, using the HuggingFace Transformers library, two types of pre-trained models were used: mBERT and XLM-RoBERTa. The models were pre-trained on Wikipedia articles in about 100 languages. The authors run the models on both WDC and Polish datasets. The F1 score was used as a metric to compare the models.
The mBERT (Devlin et al., 2019) and XLM-RoBERT (Conneau et al., 2020) models consistently outperformed other models. Notably, mBERT excelled, particularly in smaller-sized datasets (small, medium), a trend also observed in the Polish dataset. In the ”Shoes” category, results were slightly lower than those of the mBERT model. However, XLM-RoBERT performed exceptionally well in ”large” datasets.
These results demonstrate that multilingual models effectively address the product matching problem, often yielding results comparable or superior to prior studies.

3.2 Choosen Approach

Our main aim was to test a method for product similarity matching using reliable language models. An important part of the project is the limitation of the execution time, which is set to 10 ms. Because of that, in our approach, we decided to test out BERT (Bidirectional Encoder Representations from Transformers), DistilBERT (Sanh et al., 2019) as well as RoBERTa (Michał Mozdzonek, 2022). According to literature, despite having 40% fewer parameters than the original model, DistilBERT achieves competitive results by running 60% faster. RoBERTa improves upon

BERT by removing the next sentence prediction objective, utilizing dynamic masking during pretraining, and employing larger mini-batches, enhancing performance in various natural language processing tasks. Additionally, as the project description does not state otherwise, we assumed that the imposed time limitation does not include feature extraction. Therefore, we could utilize Large Language Models (LLMs) as feature extractors. This was important for us because LLMs demonstrated remarkable capabilities in understanding context and capturing complex patterns and nuances in language (Hadi et al., 2023).

In the first step, we designed a method to evaluate a solution that we were working on. One would want this metric to be able to compare the similarity between many products simultaneously and output a single number, which would measure how good the model is in ranking products on multiple levels. Ranking is especially important in e-commerce due to the fact that users typically interact with only few items at the top of the list. The input vector for the metric consists of cosine similarities between embeddings of product pairs from the transformer models. The resulting metric (called RankedSimilarity) is a sum of Kendall Tau Distance (KDT) (Cicirello, 2019) and Mean Square Error (MSE).

$$RankedSimilarity(x) = KDT(x) \quad (1)$$

$$+MSE(x, [1, 0.66, 0.33, 0]),$$

where x is a vector of cosine similarity. The former is responsible for keeping the similarities of products in order, and the latter forces similarities to be equally spaced.

KDT measures the dissimilarity between two rankings by counting the number of pairwise disagreements in ordering elements. It considers the number of concordant and discordant pairs (pairs of elements in the same or opposite order in the two rankings) to quantify the similarity or dissimilarity between the rankings. In our case, we ranked cosinus similarities of product pairs.

MSE, in our case, measures how much the vector of ordered cosine similarities between product pairs differs from the vector of values from 1 to 0 with equal differences between them.

For example, let's say that a, b, c, d are four embeddings of products from the transformer model, and we know that product b is the most similar to product a and product d is the least similar to product a. Therefore, we want the cosine similarities to be in the same order:

$$\cos(a, a) > \cos(a, b) > \cos(a, c) > \cos(a, d) \quad (2)$$

and we represent them as a vector:

$$[\cos(a, a), \cos(a, b), \cos(a, c), \cos(a, d)]$$

. KDT assures the order. However, using only that trick, the vector of cosine similarities like

$$[1, 0.99, 0.11, 0.10]$$

would result in a perfect score because the values are in order. MSE combats this by forcing differences of subsequent values to be equal and close as much as possible to

$$[1, 0.66, 0.33, 0]$$

The choice of such values was made by us and with intention to separate the individual levels of similarity as much as possible

The main idea behind similarity measure could be shown in a simple example in Table 1. As we can see, in all three models, the similarities follow expected values – it is higher when compared items that are close to each other (like red apple and green apple) and lower when items are different (like red apple and Warsaw University of Technology). All values are high, reaching even 0.9949 for the most different pairs of items. The values are high because the models are not fine-tuned for this task.

After creating a metric suitable for evaluation, the next step was to transform the original dataset to meet our needs. Example product from dataset transformed from JSON: *title: sandisk sdsdj 016g crz 16gb 9p sdhc card class 2 2mb s bulk, description: sdsdj 016g crz 16gb 9p sdhc card class 2 2mb s bulk, brand:sandisk,price:25 99 usd ,category: Computers and Accessories*. Namely, we extracted a representation of each product using LLM and based on positive pairs, negative pairs and other information about them, we created an evaluation data set in which each observation is a list of products arranged in order of similarity.

Left item	Right item	Similarity level	Cosine Similarity
Red apple	Green apple	Same fruit	0.99230
Red apple	Lemon	Both fruit	0.97707
Red apple	Brick	Both small objects	0.96443
Red apple	Warsaw University of Technology	Not similar	0.93574

Table 1: Example showing intuition behind similarity measuring.

Prompt engineering is an important technique for getting good results from LLMs (Velásquez-Henao et al., 2023). It involves crafting prompts that explicitly guide the LLM towards the desired output, ensuring that it understands the task and generates the most relevant and informative response. It is an iterative process involving experimentation and refinement to optimize the LLM's performance for specific tasks. After many iterations, we came up with the following prompt: *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:* {*title+description*}. We used the prompt along with the titles and descriptions of the products to generate augmented text representations of products that will be used as input for transformer models.
Each record in the evaluation dataset consists of representations of 5 products ordered like this $[anchor, anchor', positive, negative, category]$, where:

1. *anchor* is the main product representation that is compared to other products in a record (the equivalent of product a in the earlier example),

2. *anchor'* is a representation of the same product as *anchor* but generated independently from the first one resulting in a different string for the same product,

3. *positive* is a text representation of the product from the positive pair from the original dataset,

4. *negative* is a text representation of the product from the negative pair from the original dataset,

5. *category* is a text representation of a product from a different category than the original product given by *anchor*.

Subsequent products are less and less similar to the $anchor$, thanks to which the created dataset is suitable for evaluating models' performance using the previously described metric.

To train the BERT-like models, we utilized Triple Loss that is expressed as follows:

$$L(a, b, c) = \max(||a - b||_2 - ||a - c||_2 + \alpha, 0)$$

, where a is embedding of an anchor, b is embedding of product from positive pair and c is an embedding of a product from negative pair. Hyperameter α is a minimal sufficient difference of distances between them.

4 Experiments and results

In our experiments, we took into account the performance of pre-trained BERT, DistilBERT and RoBERTa models before and after performing our training procedure. We evaluated models using our custom metric described earlier and execution time for inference of each model.

4.1 Exploratory Data Analysis (EDA)

We conducted exploratory data analysis to understand the Web Data Commerce dataset better. We focused mainly on basic statistics like number of instances in each class, missing values in each column and length (in words) of text-like features. Additionally, we prepared word clouds from titles and descriptions and attempted to divide the pairs by looking into the percentage of words overlap.

Figure 2 provides information about the pairs count in each category and with each label. The label '0' means that the paired products are not similar, label '1' means that the paired products

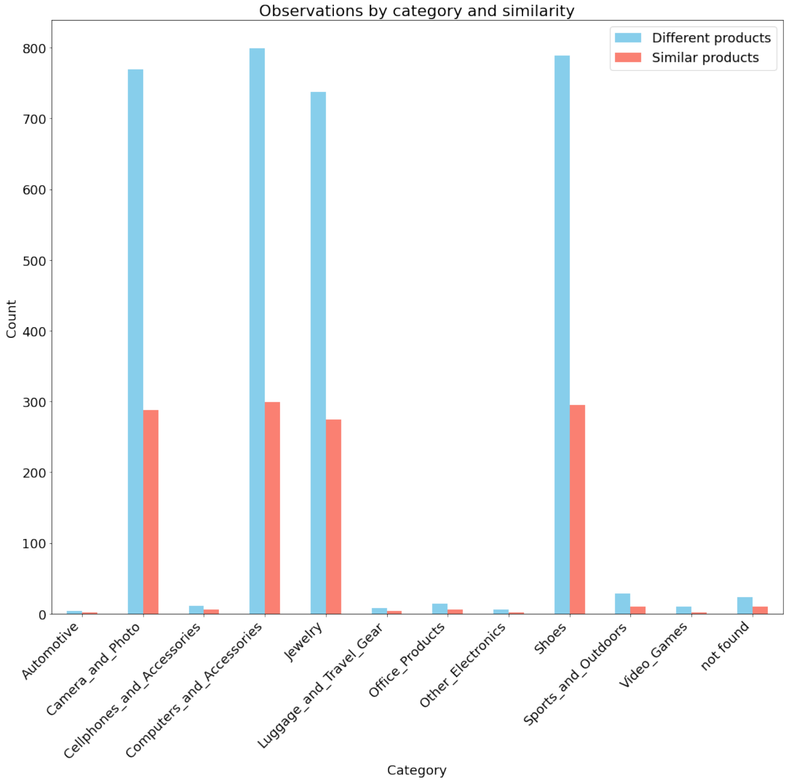

Figure 2: Number of o observations in each category and category label where the label '0' (once again) means that the paired products are not similar, label '1' means that the paired products are the same.

are the same.

Four plots are presented on figure fig. 3 for comparison showing the percentage of missing values in each column in the dataset with additional division into categories (cameras, computers, jewelry, shoes) because those four were chosen for the golden standard part of the dataset. The percentage of missing values will be considered while choosing appropriate columns for additional information about product extraction.

To try to distinguish manually (and also out of curiosity) between similar and different products a metric of the percentage of words overlap was proposed. It is calculated by taking two texts and then calculating how much overlap occurs (how many words are thein $intersection$ of the two texts) and dividing it by the size of a set created from both texts ($union$).

Figure 4 plots the distribution of these values with differentiating between labels (label=0 - Different products, label=1 - The same products). This was done for product titles, descriptions and table content. Table content is described as some additional data about the product. In the final solution, we didn't use this feature due to high missing values and finding other variables more important. Still, we decided that this should be included in EDA.

4.2 Experimental Procedures

In the first step of our experiments, we comprehensively analysed the data to uncover any key features, trends, and anomalies that may affect our project. It is explained thoroughly in 4.1.

The next step focused on using LLMs to generate augmented text representations. Example of such a representation:
Brand: Acer
Model: Aspire
Series: ES1-132
Processor: P194
Type: Business Notebook
Features:
+ Display: 13.3 inches, Full HD (1920 x 1080)
+ RAM: 4GB DDR4
+ Storage: 512GB SSD

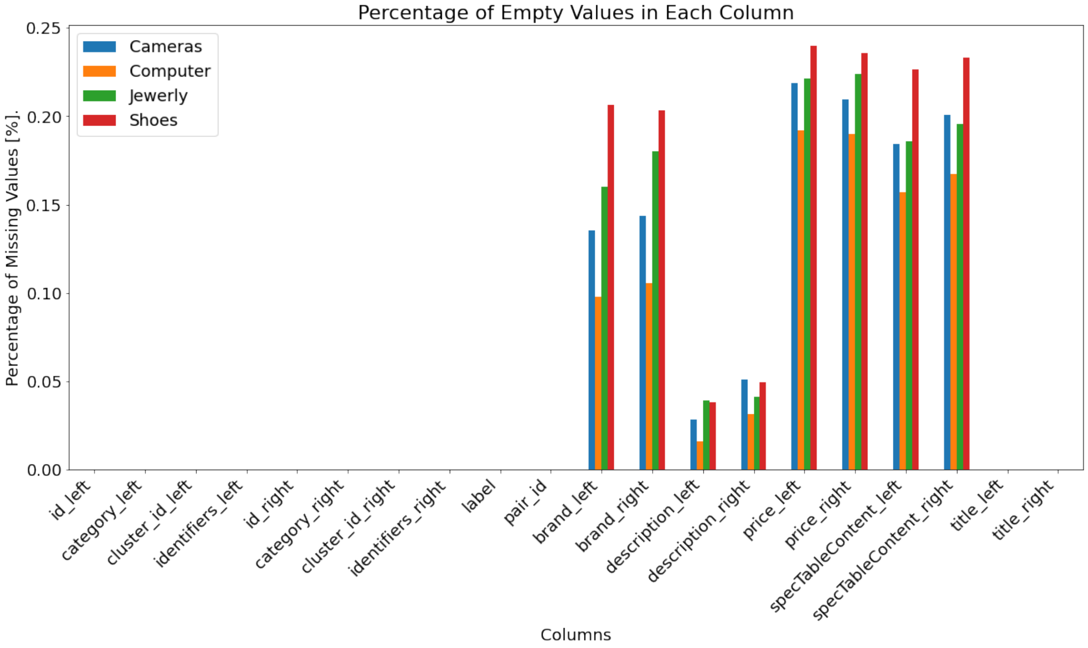

Figure 3: Percentage of missing values in each column in the dataset with additional division into categories (cameras, computers, jewelry, shoes).

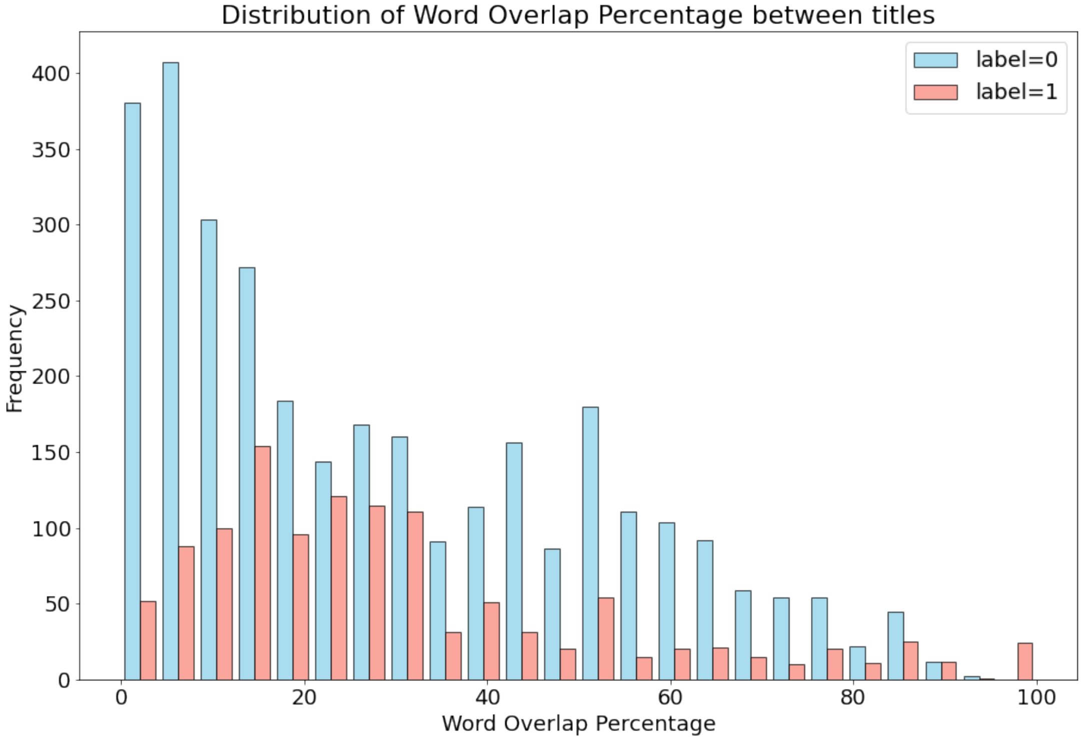

Figure 4: Histogram of percentage of words overlap in titles.

+ Battery Life: Up to 12 hours
+ Ports: 2x USB-C, 1x USB-A, 1x HDMI, 1x Headphone Jack
+ Operating System: Windows 10 Pro
+ Weight: 1.4kg
+ Dimensions: 307.7 x 210.6 x 17.9mm
Context:
+ Target audience: Business professionals, entrepreneurs, and small
business owners
+ Use cases: Work, travel, and remote collaboration
+ Unique selling points: Sleek design, long battery life, and portability

During the project, we considered different models and services where those models were available. One of the best models is GPT3.5, which is available via the OpenAI API. Unfortunately, because it is paid, we decided not to use it. We also tried to host the LLama 7B model locally, but the results were unsatisfactory. Ultimately, we used LLama 70B provided by HuggingFace for free. However, outputs from the model were in English, regardless of the language in the input, which is one of the reasons we decided to abandon experiments on the Polish data set. We attempted to engineer the prompts in such a way that the models outputs in a language matching the description. However no success has been found in this attempt. When testing on a bigger free model (ChatGPT), it returned the output in a matching language without any prompt engineering. This lead us to the conclusion that the model we used was insufficient for this part of the task, since we are using a free LLM with a limited power in comparison to paid versions like GPT-x models.

At the same time, we tried to determine which transformer model would be best suited for the task of calculating product similarity. Due to the project requirements, we were forced to consider smaller, more effective models that would provide inference in less than 0.01 seconds per product pair. We decided to use BERT-based models known for their bidirectional architecture and efficient deployment. We planned to utilize HerBERT for the Polish dataset; however, we rejected this idea due to the limitations mentioned earlier.

The next major step focused on creating an evaluation method for multi-level similarity. We recognised two key characteristics that our metrics should have. The first one is ordering similar products. It is important from the business perspective; for example, placing similar products in a logical order can enhance the overall shopping experience for customers. The second one is to uniformly separate the similarity scores (cosine similarity) to avoid situations where the least similar product has a 0.99 similarity score.

After that, we came back to LLMs. Using them aims to get a richer representation of the product information. Following previous works, we decided to use the title and description of the products. We iteratively refined the prompt to force the model to add some information or descriptions from its knowledge that may be helpful in further processing. This was done by modifying the structure of the sentence, as well as removing and adding different instructions, and then manually assessing whether the modified prompt returns better results on a basis of a few, hand-selected examples.

With the Hugging Face API, a prompt, and a metric to evaluate, we prepared a dataset that fits our needs. Each row consisted of five text representations of the products generated by LLM. Each column represented a different level of similarity to the ”main” product; for more information, see Section 3.2.

Using our augmented dataset, we performed fine-tuning to leverage the knowledge gained during the initial training.

Our final results consist of 5 different BERT models' performance on the WDC dataset. Parameters of the models can be seen in 2 As seen in Figure 5, all the tested models achieve high results in the measure. However, it is worth noting that the finetuned models score worse than the pretrained ones. This issue is further described in Section 5. The best model (BERT) is closely followed by the second best (DistilBERT). However, a significant difference is measured in the inference time of the models, as seen in Figure 5. The distilled version executes nearly twice as fast while barely lagging in the metric score.

5 Discussion

The results are satisfactory as the first attempt at defining the measure for incorporating taxonomy into a similarity comparison. Adding LLM data augmentation in this manner is a significant, novel approach (as far as we know) to elevate the accuracy and context-awareness of the em-

Optimizer	Adam
Beta1 (Adam)	0.9
Beta2 (Adam)	0.999
Learning Rate	0.001
Batch Size	1
Epochs	3

Table 2: Hyperparameters used in training.

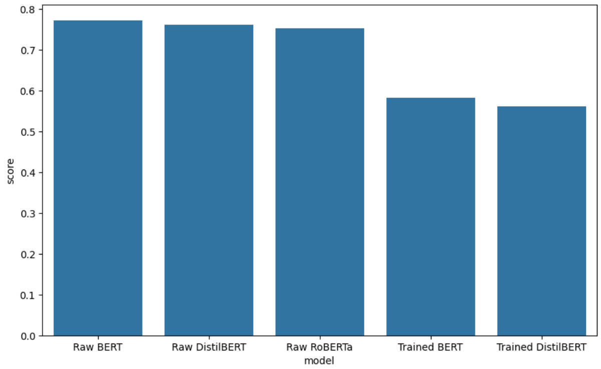

Figure 5: Evaluation metric score for pre-trained models before and after our training

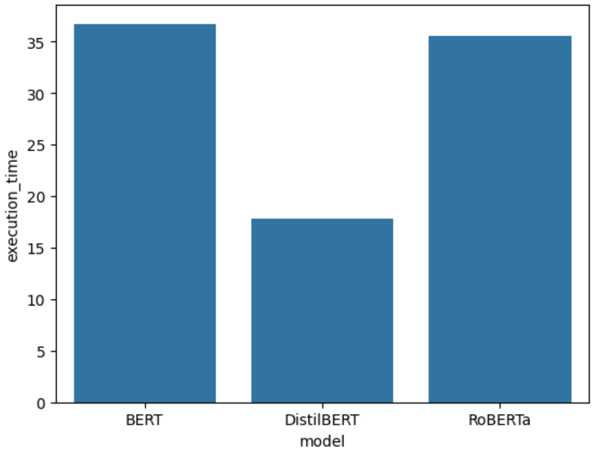

Figure 6: Inference time for chosen models.

bedded vectors. The worse performance of the fine-tuned models is a surprising effect, which might be caused by the low amount of data available in the dataset or the big concept gap between tasks of similarity measurement and prediction of masked sentence elements (as in BERT). The untrained models, on the same example, achieved an example cosine similarity vector like ([1.0000, 0.9597, 0.9527, 0.9481]) while the fine-tuned model ([0.9417, 0.9588, 0.9497, 0.9528]). This shows how exactly these models fail since, with training, they approach the ideal score of ([1, 0.66, 0.33, 0]) rather than get further away from it and lose the decreasing characteristic.

Unfortunately, we did not manage to reach the initially set out threshold of 10ms per comparison. The fastest-performing model scored similarly to the slowest but highest metric model, achieving 18ms per pair. This time consists of both the embedding of text and similarity measurement. However, this shortcoming can be remedied by modifying the underlying system structure in a commercial setup and more closely integrating it with internal processes, such as a registration of a new product. The data for a single product could be embedded once and saved in an indexed vector database, as is usually done in standard e-commerce projects. This would allow for fast retrieval of the data when needed, and the time of similarity measurement is irrelevant, as it reaches $46.6\mu s \pm 4.64\mu s$ (on 700,000 tests), which is around 400 times less. Additional ways of decreasing the inference time are further described in future works in Section 6.

Additionally, in our research, we encountered multiple issues with the limited computational resources, such as:

- Difficulties with setting up the dataset nor the LLM on EDEN (problems with the available RAM or drivers or time; EDEN had problems for a significant amount of time).

- Inference time of self-hosted LLM turned out to be extremely long.

- Inference time of publicly hosted LLM also turned out to be very long.

- API limitations ran out fast.

- Free GPU for training run out fast.

We have used Google Collab to run our training code, as it allows us to do so using GPU acceleration free of charge. However Google dynamically allocates the time limits for users running code, and in our case the allowed time was around 12 hours per week, which allowed for a single training of fine-tuned models. The exact values are unknown and vary per user account, so it is unknown when you will be able to use it once again.

6 Conclusions and future work

To sum up our contribution, the presented study makes several significant contributions to the field of product matching. Firstly, by usage of the Web Data Commons (WDC) dataset, which offers a substantial volume of diverse data for evaluating deep learning-based matching methods, addressing the limitations of traditional benchmark datasets. The study proposes a novel evaluation metric, RankedSimilarity, combining Kendall Tau Distance (KDT) and Mean Square Error (MSE), to effectively measure the similarity between products. Additionally, the proejct explores the use of Large Language Models (LLMs) for feature extraction and augmentation, enhancing the context-awareness and accuracy of embedded vectors. Experimental results demonstrate the efficacy of BERT-based models, particularly DistilBERT, in achieving high similarity scores within stringent time constraints. Despite challenges such as limited computational resources and fine-tuning model performance, the study provides valuable insights into the effectiveness of LLMs in product matching tasks and lays the groundwork for future research in this area.

Several key areas exist to explore in future developments of our similarity measurement for e-commerce project. Firstly, expanding our methodology to include a broader range of datasets is crucial for assessing the generalizability of our product similarity matching system across different domains. Testing the solution on a bigger and cleaner dataset could provide the necessary amount of data for the models, as scaling typically improves the results of neural networks. Additionally, testing our recommendation system on advanced Large Language Models (LLMs) beyond the current state-of-the-art models like GPT-3 or BERT will ensure adaptability to evolving NLP technologies. Experimenting with various prompts for feature extraction and evaluating different system prompts during the recommendation process will contribute to optimizing the system's performance. Furthermore, conducting comparative analyses without the reliance on Large Language Models will provide insights into the specific impact of these models on the recommendation system. Exploring alternative methods for description extraction, including domain-specific techniques or specialized models, is essential for diversifying the approach. Finally, testing the performance and success of the recommendation system in a real e-commerce environment, potentially through collaboration with industry partners or deployment in controlled settings, will validate its practical applicability and reveal opportunities for refinement and enhancement. Continuing research in these areas will contribute to the evolution of e-commerce recommendation systems, ensuring improved user experiences and advancements in the field.

References

[Banerjee et al.2023] Debarag Banerjee, Pooja Singh, Arjun Avadhanam, and Saksham Srivastava. 2023. Benchmarking llm powered chatbots: Methods and metrics.

[Bobadilla et al.2013] Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. 2013. Recommender systems survey. *Knowledge-based systems*, 46:109–132.

[Brown et al.2020] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

[Cicirello2019] Vincent A Cicirello. 2019. Kendall tau sequence distance: Extending kendall tau from ranks to sequences. *arXiv preprint arXiv:1905.02752*.

[Conneau et al.2020] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

[Devlin et al.2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for

Task (Total time est.)	**Main Contributor**	**Secondary Contributor**
Project Proposal (10h)	Everyone	-
Exploratory Data Analysis (6h)	*Kinga Ułasik*	*Szymon Rećko*
Proof Of Concept (6h)	*Mateusz Sperkowski*	*Patryk Tomaszewski*
First Milestone Presentation & Raport (14h)	Everyone	-
Reviews 1 (4h)	Everyone	-
Generating the additional descriptions (4h)	*Szymon Rećko*	*Mateusz Sperkowski*
Tuning the models (8h)	*Patryk Tomaszewski*	*Kinga Ułasik*
Reviews 2 (4h)	Everyone	-
Second Milestone Presentation & Raport (19h)	Everyone	-

Table 3: Work contribution and estimated total time for each task. While two main contributors are listed, all tasks were done with the support and help of everyone in the team when necessary.

language understanding. In *North American Chapter of the Association for Computational Linguistics*.

[Gou et al.2021] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129:1789–1819.

[Hadi et al.2023] Muhammad Usman Hadi, Rizwan Qureshi, Abbas Shah, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, Seyedali Mirjalili, et al. 2023. A survey on large language models: Applications, challenges, limitations, and practical usage. *Authorea Preprints*.

[He et al.2023] Xiaoxin He, Xavier Bresson, Thomas Laurent, Adam Perold, Yann LeCun, and Bryan Hooi. 2023. Harnessing explanations: Llm-to-lm interpreter for enhanced text-attributed graph representation learning.

[Hollmann et al.2023] Noah Hollmann, Samuel Müller, and Frank Hutter. 2023. Large language models for automated data science: Introducing caafe for context-aware automated feature engineering.

[Kasneci et al.2023] Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, et al. 2023. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and individual differences*, 103:102274.

[Koch et al.2015] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. 2015. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille.

[Liu et al.2018] Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. 2018. Rethinking the value of network pruning. In *International Conference on Learning Representations*.

[McInerney et al.2023] Denis Jered McInerney, Geoffrey Young, Jan-Willem van de Meent, and Byron C. Wallace. 2023. Chill: Zero-shot custom interpretable feature extraction from clinical notes with large language models.

[Michał Mozdzonek2022] Sergiy Tkachuk Szymon Łukasik Michał Mozdzonek, Anna Wróblewska. 2022. Multilangual transformers for product matching – experiments and a new benchmark in polish.

[Ouyang et al.2023] Shuyin Ouyang, Jie M. Zhang, Mark Harman, and Meng Wang. 2023. Llm is like a box of chocolates: the non-determinism of chatgpt in code generation.

[Peeters and Bizer2023] Ralph Peeters and Christian Bizer. 2023. Using chatgpt for entity matching. *arXiv preprint arXiv:2305.03423*.

[Peeters et al.2020] Ralph Peeters, Christian Bizer, and Goran Glavaš. 2020. Intermediate training of bert for product matching. *small*, 745(722):2–112.

[Peeters et al.2023] Ralph Peeters, Reng Chiz Der, and Christian Bizer. 2023. Wdc products: A multi-dimensional entity matching benchmark.

[Polino et al.2018] Antonio Polino, Razvan Pascanu, and Dan-Adrian Alistarh. 2018. Model compression via distillation and quantization. In *6th International Conference on Learning Representations*.

[Sanh et al.2019] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

[Scarselli et al.2008] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80.

[Thirunavukarasu et al.2023] Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. 2023. Large language models in medicine. *Nature medicine*, 29(8):1930–1940.

[Tracz et al.2020] Janusz Tracz, Piotr Iwo Wójcik, Kalina Jasinska-Kobus, Riccardo Belluzzo, Robert Mroczkowski, and Ireneusz Gawlik. 2020. Bert-based similarity learning for product matching. In *Proceedings of Workshop on Natural Language Processing in E-Commerce*, pages 66–75.

[Vaswani et al.2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

[Velásquez-Henao et al.2023] Juan David Velásquez-Henao, Carlos Jaime Franco-Cardona, and Lorena Cadavid-Higuita. 2023. Prompt engineering: a methodology for optimizing interactions with ai-language models in the field of engineering. *DYNA*, 90(230):9–17.

[Wei et al.2022] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *Transactions on Machine Learning Research*.

[Wu et al.2023] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*.

[Zhang et al.2019] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM computing surveys (CSUR)*, 52(1):1–38.