Transformer Based Ecommerce Recommendation System (A Foundational Model)

Samarpita Bhaumik
M.Tech 2023-25
IIIT Bangalore
samarpita.bhaumik@iitb.ac.in

Sunnidhya Roy
M.Tech 2023-25
IIIT Bangalore
sunnidhya.roy@iiitb.ac.in

Prof. Raghuram Bharadwaj
Assistant Professor
IIIT Bangalore
raghuram.bharadwaj@iiitb.ac.in

ABSTRACT

E-commerce recommendation systems, utilizing collaborative filtering, content-based filtering, and hybrid models, play a crucial role in personalizing user experiences and boosting sales. Collaborative filtering groups users with similar preferences to generate real-time recommendations, while content-based filtering suggests products based on user attributes and preferences extracted from search queries and reviews. Hybrid models combine these approaches, refining recommendations using customer purchase history and reviews. Despite their effectiveness, challenges like cold start issues, scalability, and data validity periods persist. Solutions involve enhancing personalization, contextual relevance, and leveraging scalable cloud-based systems. Continued innovation in these recommendation models is vital for e-commerce platforms to remain competitive and maximize sales in the evolving digital landscape.

Our solution works on enhancing the contextual relevance of the users input and provide them with recommendations which are much more relevant and sits well with the users context, ideally in ecommerce websites one has to enter specific product to get search results which are mostly of the same type as the user wanted, our approach enables the users to type in Natural Language what they want and our Model which have been trained on around 54000 products from different categories understands what the user wants based on the context of the input and recommends them products.

Our solution exploits the context space of the products and gives great results even when trained on small amount of data due to computational limitation. If it were to be trained on billions of products available and if we extend the solution which we got using 54000 datapoints, the results would be even more amazing.

Also, as our solution is capable to recommend users product across different categories it will in turn result in more user engagement which ultimately will immensely help businesses.

Categories and Subject Descriptors

H.3.5 **[Information Systems]:** On-line Information Services; I.2.6 **[Computing Methodologies]:** Learning.

General Terms

Algorithms, Experimentation.

Keywords

Recommendation Systems, Ecommerce Recommendation, Transformer, BERT, Multiclass Classification, TF-IDF.

1. INTRODUCTION

In the rapidly evolving world of e-commerce, recommendation systems have become indispensable tools for personalizing user experiences and significantly boosting sales. These systems, which include collaborative filtering, content-based filtering, and hybrid models, have traditionally served as the backbone of product recommendations. Collaborative filtering operates by grouping users with similar preferences to generate real-time recommendations. This method works exceptionally well when there is ample user interaction data available, as it compares the behaviors of similar users to suggest relevant products. However, collaborative filtering faces significant challenges, notably the cold start problem, where new users or items lack sufficient interaction history to generate reliable recommendations. Moreover, this approach can sometimes fail to capture the nuanced preferences of users, as it primarily relies on historical interaction data, potentially limiting the diversity and relevance of the recommendations.

Content-based filtering, on the other hand, focuses on the characteristics of items and users. It analyzes product descriptions, user profiles, and other attributes to recommend similar items to those a user has interacted with in the past. While this method can mitigate cold start issues to some extent, its recommendations often become monotonous, as it tends to suggest items within the same category, limiting the discovery of diverse products that might also be of interest to the user. Hybrid models attempt to balance the strengths and weaknesses of both collaborative and content-based filtering by combining these approaches. Although hybrid models are more robust, they still encounter challenges related to data sparsity and computational complexity, especially as the volume of data grows. These challenges necessitate continuous innovation to maintain the effectiveness and efficiency of recommendation systems.

Despite the advancements made in traditional recommendation systems, the demand for even more sophisticated and user-centric solutions is growing. E-commerce platforms must innovate continuously to stay competitive, maximize sales, and enhance user engagement. This is where our innovative, context-aware product recommendation system comes into play, offering a transformative approach to product recommendations.

Our context-aware product recommendation system redefines the capabilities of traditional systems by enhancing contextual relevance. Unlike conventional systems that require users to enter specific product queries to retrieve search results confined to a particular type, our approach allows users to express their needs in natural language. For instance, users can input requests like "Suggest me some good handmade products" or "I have a birthday party tomorrow, suggest me some products." Our advanced model, trained on a dataset of approximately 54,000 products spanning various categories, comprehends the context of these inputs and recommends products accordingly.

Our context-aware recommendation system offers significant advantages over traditional methods. By utilizing advanced natural language processing (NLP) techniques using BERT transformer model, it accurately interprets user intent and context, providing personalized and relevant suggestions that enhance user satisfaction. Unlike conventional systems confined to single categories, our approach presents a diverse range of products across categories, encouraging broader exploration and a richer shopping experience. This system improves user engagement, leading to higher satisfaction, loyalty, and conversion rates. Even with a relatively small dataset, our model delivers impressive results, suggesting exponential performance gains with larger datasets. This scalability benefits new or smaller e-commerce platforms. Additionally, our system's ability to drive cross-category sales and uncover hidden demand patterns boosts sales, taps into new market segments, and optimizes inventory management, enhancing revenue streams and profitability for e-commerce platforms.

Our solution uses the product description, features, category data which is preprocessed and converted to a meaningful sentence on which we then apply the power of TF-IDF algorithm to give a kickstart with contexts to a pre-trained BERT transformer which is trained and fine-tuned on the information about products which we meaningfully generated. Post training, we give our input in Natural Language to the trained model and it recommends us products.

This is a very robust solution and it has the potential to be very powerful once trained on huge amount of data.

2. OUR PROPOSED SOLUTION:

2.1 A little about our dataset.

We selected the publicly available dataset "Amazon Reviews 23" from UCSD: Amazon Reviews'23 (amazon-reviews-2023.github.io) [1].

This is a large-scale **Amazon Reviews** dataset, collected in **2023** by McAuley Lab, and it includes rich features such as:

1. **User Reviews** (*ratings, text, helpfulness votes*, etc.);
2. **Item Metadata** (*descriptions, price, raw image*, etc.);
3. **Links** (*user-item / bought together* graphs).

Fig 1. Dataset Information

For Item Metadata

Field	Type	Explanation
main_category	str	Main category (i.e., domain) of the product.
title	str	Name of the product.
average_rating	float	Rating of the product shown on the product page.
rating_number	int	Number of ratings in the product.
features	list	Bullet-point format features of the product.
description	list	Description of the product.
price	float	Price in US dollars (at time of crawling).
images	list	Images of the product. Each image has different sizes (thumb, large, hi_res). The "variant" field shows the position of image.
videos	list	Videos of the product including title and url.
store	str	Store name of the product.
categories	list	Hierarchical categories of the product.
details	dict	Product details, including materials, brand, sizes, etc.
parent_asin	str	Parent ID of the product.
bought_together	list	Recommended bundles from the websites.

Fig 2. Dataset Information


```
......
{
  "main_category": "All Beauty",
  "title": "Lurrose 100Pcs Full Cover Fake Toenails Artificial Transparent Nail Tips Nail Art for DIY",
  "average_rating": 3.7,
  "rating_number": 35,
  "features": [
    "The false toenails are durable with perfect length. You have the option to wear them long or clip them short, easy to trim and file them to in any length and shape you like.",
    "ABS is kind of green enviromental material, and makes the nails durable, breathable, light even no pressure on your own nails.",
    "Fit well to your natural toenails. Non toxic, no smell, no harm to your health.",
    "Wonderful as gift for girlfriend, family and friends.",
    "The easiest and most efficient way to do your toenail tips for manicures or nail art designs. It's fashion, creative, a useful accessory brighten up your look, also as a gift."
  ],
  "description": [
    "Description",
    "The false toenails are durable with perfect length. You have the option to wear them long or clip them short, easy to trim and file them to in any length and shape you like. Plus, ABS is kind of green enviromental material, and makes the nails durable, breathable, light even no pressure on your own toenails. Fit well to your natural toenails. Non toxic, no smell, no harm to your health.",
    "Feature",
    "- Color: As Shown.- Material: ABS.- Size: 14.3 x 7.2 x 1cm.",
    "Package Including",
    "100 x Pieces fake toenails"
  ],
  "price": 6.99,
  "images": [
    {
      "hi_res": "https://m.media-amazon.com/images/I/41a1Sj7Q20L._SL1005_.jpg",
      "thumb": "https://m.media-amazon.com/images/I/31dlCd7tHSL._SS40_.jpg",
      "large": "https://m.media-amazon.com/images/I/31dlCd7tHSL.jpg",
      "variant": "MAIN"
    },
```


Fig 3. Dataset Information

We have worked on 50000 products spanning 27 categories. The attached pictures contain the meta data about the products which we will be utilizing for our solution pipeline.

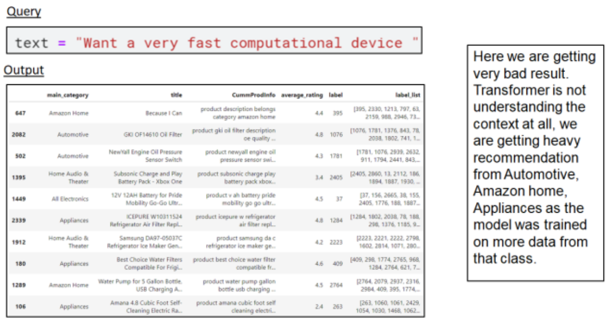
2.2 Analysis of the dataset

The original data was unbalanced meaning a single category was dominating the context space which resulted in biased recommendation towards that class so to strike balance and get better recommendation we sampled 2000 data points from 27 categories, this was required as we were training on 54000 datapoints only primarily due to resource limitation. In case the model is trained on billions of data where each category contains millions of products the imbalance would hardly matter and the context space and the embedding vectors would be increasingly richer

Fig 4. Sampling equally from every category to remove class imbalance due to less amount of data.

2.3 Bad Result due to Class imbalance

As we mentioned this is happening mainly because of less amount of data.


Query

```
text = "Want a very fast computational device "
```

Output

	main_category	title	CummProdInfo	average_rating	label	label_list
647	Amazon Home	Because I Can	product description belongs category amazon home	4.4	395	[395, 2330, 1213, 797, 63, 2159, 988, 2946, 73...
2082	Automotive	GKI OF14610 Oil Filter	product gki oil filter description oe quality ...	4.8	1076	[1076, 1781, 1376, 843, 78, 2038, 1802, 741, 1...
502	Automotive	NewYall Engine Oil Pressure Sensor Switch	product newyall engine oil pressure sensor swi...	4.3	1781	[1781, 1076, 2939, 2632, 911, 1794, 2441, 843,...
1395	Home Audio & Theater	Subsonic Charge and Play Battery Pack - Xbox One	product subsonic charge play battery pack xbox...	3.4	2405	[2405, 2860, 13, 2112, 186, 1894, 1887, 1930, ...
1449	All Electronics	12V 12AH Battery for Pride Mobility Go-Go Ultr...	product v ah battery pride mobility go go ultr...	4.5	37	[37, 156, 2665, 38, 155, 2405, 1776, 188, 1887...
2339	Appliances	ICEPURE W10311524 Refrigerator Air Filter Repl...	product icepure w refrigerator air filter repl...	4.8	1284	[1284, 1802, 2038, 78, 188, 298, 1376, 1185, 9...
1912	Home Audio & Theater	Samsung DA97-05037C Refrigerator Ice Maker Gen...	product samsung da c refrigerator ice maker ge...	4.2	2223	[2223, 2221, 2222, 2798, 1602, 2814, 1071, 280...
180	Appliances	Best Choice Water Filters Compatible For Frigi...	product best choice water filter compatible fr...	4.6	409	[409, 298, 1774, 2765, 968, 1284, 2764, 621, 7...
1289	Amazon Home	Water Pump for 5 Gallon Bottle, USB Charging A...	product water pump gallon bottle usb charging ...	4.5	2764	[2764, 2079, 2937, 2316, 2984, 409, 395, 1774,...
106	Appliances	Amana 4.8 Cubic Foot Self-Cleaning Electric Ra...	product amana cubic foot self cleaning electri...	2.4	263	[263, 1060, 1061, 2429, 1054, 1030, 1468, 1062...

Here we are getting very bad result. Transformer is not understanding the context at all, we are getting heavy recommendation from Automotive, Amazon home, Appliances as the model was trained on more data from that class.

Fig 5. Result due to class imbalance

2.4 Our Final Dataset

As you can see the **"CummProdInfo"** column contains a meaningful concatenation of the columns **"title", "description"** and **"main_category"** , the **"label"** column has the LabelEncoded value for each item and **"label_list"** column contains list of labels of the top 50 similar items to each product which we will find out using TF-IDF and cosine similarity.

	main_category	title	CummProdInfo	average_rating	label	label_list
51169	Appstore for Android	Best Classical Music Ringtones	product best classical music ringtones descrip...	3.3	7636	[7636, 47005, 13524, 36822, 372, 24632, 22891,...
5015	Arts, Crafts & Sewing	PEPPERLONELY 50Gram Mixed Color Assorted Leaf ...	product pepperlonely gram mixed color assorted...	4.0	34687	[34687, 34688, 34689, 34686, 8985, 23303, 4065...
12360	Software	Total Security 2012 - 3 Users/1 Year [Download]	product total security user year download desc...	4.5	47089	[47089, 23970, 7980, 47340, 7982, 7974, 25528,...
7807	Baby	Cloud b Twilight Keychain, Ladybug (Discontinu...	product cloud b twilight keychain ladybug disc...	3.5	11516	[11516, 30822, 17765, 40220, 27496, 50934, 130...
64654	Digital Music	NCT - Universe, Jewel Case (JOHNNY Cover incl....	product nct universe jewel case johnny cover i...	4.6	32012	[32012, 19424, 40595, 15666, 3126, 5497, 32263...
...	...	...	...	...	...	...
44329	Musical Instruments	Instrument Clinic Shellac Sticks (6 pack), for...	product instrument clinic shellac stick pack s...	5.0	23918	[23918, 52857, 40902, 24959, 11316, 2355, 2522...
71739	Toys & Games	Eaglemoss B1 Centauro Italian Wheeled Tank Des...	product eaglemoss b centauro italian wheeled t...	3.3	15732	[15732, 22344, 29395, 29396, 6433, 41375, 827,...
42121	Office Products	Trading Card Thickness Guide- Measure Thicknes...	product trading card thickness guide measure t...	5.0	47180	[47180, 29345, 2110, 48147, 47179, 33692, 2269...
67068	Tools & Home Improvement	Lavalier Microphone Lapel Microphone Wired Sax...	product lavalier microphone lapel microphone w...	5.0	27261	[27261, 51741, 45054, 21518, 47025, 46036, 374...
78089	Appstore for Android	Funky Radio	product funky radio description funk music sta...	3.9	18883	[18883, 11338, 23969, 22618, 50100, 30445, 845...

54000 rows × 6 columns

Fig.6. Final Dataset

2.5 Model Selection

Initially we started with **"BertForSequenceClassification"** but it was taking a lot of time to train so we then shifted to **"DistilBertForSequenceClassification"** for multi-label classification. **"BertForSequenceClassification"** was taking almost 10hrs to train on 54000 datapoints that too only for 3 epochs.

Hence we shifted to **"DistilBertForSequenceClassification".** The reason why we shifted to the model was that it is a smaller model almost 44% smaller than BERT, it is 60% percent faster than BERT both while training and inference and it retains 97% of BERT's performance. Please refer to the article: (Large Language Models: DistilBERT — Smaller, Faster, Cheaper and Lighter | by Vyacheslav Efimov | Towards Data Science) **[2]**. **Knowledge Distillation helps in the speed of the model while retaining the model performance.**

On 54000 data points with 27 categories our model training took almost 10hrs to train.

Table 1. Comparison between BERTModel and DistilBERTModel

Model	Training Time for 12 epochs
BERTForSequenceClassification	~34hrs
DistilBERTForSequenceClassification	~10hrs

2.5.1 Knowledge Distillation

The concept of distillation is quite intuitive: it is the process of training a small student model to mimic a larger teacher model as close as possible. Distillation would be useless if we only run machine-learning models on the cluster we use to fine-tune them, but sadly, it isn't the case. Therefore, distillation comes in whenever we want to port a model onto smaller hardware, such as a limited laptop or a cellphone, because a distilled model runs faster and takes less space. Please refer to the following article **[7].** The main reason for us to use **DistilBERTForSequenceClassification** was because of computational limitation and also because **DistilBERTForSequenceClassification** retains 97% of original **BERTForSequenceClassification** model.

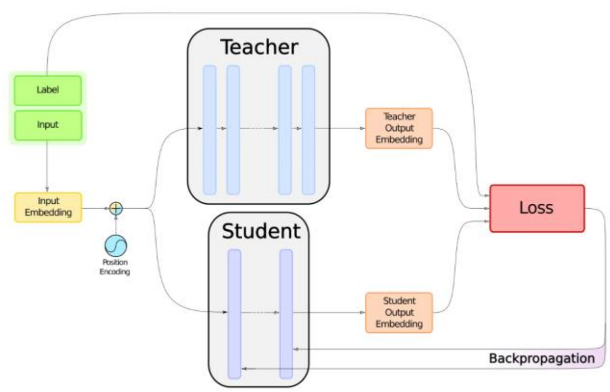


Fig 7. Knowledge Distillation using Teacher Student Network

2.6 Algorithm Overview

To begin, we already created the dataset as outlined in the section 2.4 above.

(1) The first step involves data preprocessing specifically on the **"CummProdInfo"** column. This includes removing HTML tags, special characters, and stopwords, converting text to lowercase, word tokenization, and lemmatization.

(2) Subsequently, labels are assigned to each item using LabelEncoder to ensure uniqueness. (This was done as there can be two products with the same title but belonging to 2 different categories).

(3) Moving forward, TF-IDF is utilized to vectorize each product's "CummProdInfo," enabling the calculation of cosine similarity between every pair of products in the dataset.

(4) The top 50 similar product indices are selected based on similarity scores, with the option for various thresholding techniques (one can set a dynamic thresholding as well).

(5) Using these similarity scores, a new column named **"label_list"** is created, containing the list of labels for all similar items to a given product.

(6) MultiLabelBinarizer is then employed to generate a 54,000-dimensional vector for each item. The 1s in the vector is indicating similarity to all the items in the **"label_list"** column for that particular item. And any suitable x where $0 \leq x < 1$ for rest of the items. (This vector is used as a ground truth for the model training, this helps to provide a kickstart to the Transformer by providing them with a starting context space denoting the items belonging to different categories which might be contextually similar to that particular item, we can denote this to be **"cross-attention"** and then based on the value of x we group items of the same category which we can denote as **"self-attention"**. This dual approach helps the model derive contexts not only from items within the same category but also across different categories. As a result, the embedding vector is increasingly refined, capturing subtle context changes and providing a deeper understanding of the user's input. This enhanced embedding vector effectively interprets the user's needs, emotions, and other nuanced requirements. If you did not understand please refer to section 2.8 below for analysis of the algorithm.

(7) Following this, the input data is tokenized using **DistilBERTTokenizer**, resulting in input IDs and attention masks. Subsequently, a train-test split is performed, and the train dataset, comprising input IDs, attention masks, and labels, is fed into a pre-trained model which is thus fine-tuned for our ecommerce recommendation task as a multilabel classification problem.

(8) Once the model is trained, validation and testing procedures are carried out to evaluate its performance.

2.7 Algorithm Pipeline

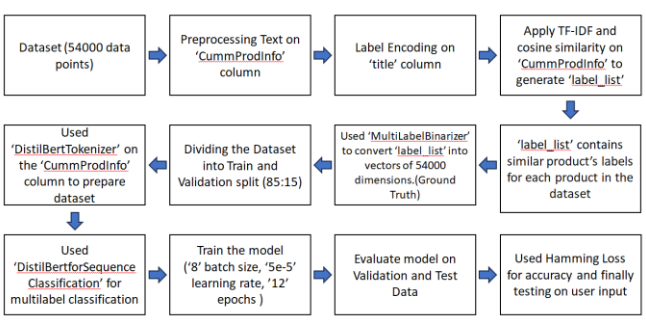


Fig 8. Solution pipeline

2.8 Let us Analyze our algorithm

(1) So why did we apply TFIDF? And let us analyze how it might help our model learn better?

TFIDF provides a context representation for each sentence. We decided to use this context as the foundational context for our transformer model on top of which it learns even better context and deliver improved recommendations through highly specialized embedding vectors for each product capable to detect even a subtle change in the input context. Through various experiments, we found that this technique significantly enhances our recommendations. The quality of recommendation measured using User's satisfaction improved from say ~40% to ~95%.

One key observation is that using this technique allows items of different types, which may belong to different categories but share semantic similarities, to learn from each other (cross-attention). Meanwhile, items within the same category learn within their own group (self-attention). For example, when given the input, "I have a birthday tomorrow, suggest me some items," our model's recommendations are not limited to the birthday or gift categories but span across various categories like toys, appliances, handmade items, etc. This indicates that similar items from different categories learn from each other, ultimately leading to an improved representation for each token and enriching the context space [3] [6].

(2) What is novel about our approach?

(i) First, we are allowing the user to type in their natural language thus removing the boundary to give a clear title of the product. This approach can make the application utilizing this model to recommend products much more robustly with a clear focus on the user's nuanced requirements, NLU thus helps to handle any kind of inputs from the user. One important use-case particularly important for India is that, as majority of the population speaks broken English our model is capable to pick up context even from a poorly constructed sentence because of the rich embedding space created during training.

(ii) Based on our data analysis, we discovered that similar products often transcend single categories, prompting us to leverage TF-IDF + cosine similarity method to establish our base context. This approach proved beneficial as it provided our Transformer model with an initial understanding of contexts. Transformers excel at learning contextual information, and they capitalized on this capability to develop robust representations for each token and generate an effective embedding space [3] [6].

(iii) We then pondered how we could reframe the recommendation task into another problem domain. This led us to consider whether it resembled a **multilabel classification problem**. Trusting our instincts, we embarked on a rigorous training process involving 54,000 data points. Over a span of 10 hours, with a learning rate of 5e-5, 12 epochs, and a batch size of 8, our efforts yielded tangible results that we can now demonstrate and is available in section 2.9.

(iv) Our approach to solving this recommendation task can be deemed novel. We amalgamated various concepts and interconnected them to achieve our ultimate objective, resulting in a robust and effective solution. We are confident that with further training on a larger dataset, employing more epochs, and fine-tuning hyperparameters, this model could serve as a foundational framework not only for e-commerce recommendation systems but also for recommendation systems across diverse domains.

Thus, through both self and cross-attention mechanisms, our model achieves a better understanding and delivers superior recommendations.

3. PERFORMANCE MEASURE

Epoch	Training Loss	Validation Loss	F1 Score	Hamming Loss	Roc Auc	Runtime	Samples Per Second	Steps Per Second
1	0.007300	0.007164	0.000000	0.000926	None	471.696500	17.172000	1.075000
2	0.006700	0.006287	0.000000	0.000926	None	444.953100	18.204000	1.139000
3	0.005200	0.004911	0.002422	0.000915	None	555.100200	14.592000	0.913000
4	0.004500	0.004143	0.022242	0.000869	None	545.044300	14.861000	0.930000
5	0.003800	0.003648	0.067162	0.000807	None	537.207800	15.078000	0.944000
6	0.003300	0.003321	0.111612	0.000760	None	566.852800	14.289000	0.894000
7	0.003000	0.003124	0.148054	0.000726	None	556.577700	14.553000	0.911000
8	0.002900	0.002968	0.182326	0.000697	None	553.967100	14.622000	0.915000
9	0.002700	0.002868	0.205497	0.000678	None	555.096400	14.592000	0.913000
10	0.002600	0.002811	0.220361	0.000665	None	541.769100	14.951000	0.936000
11	0.002500	0.002772	0.231821	0.000656	None	538.502900	15.042000	0.941000
12	0.002400	0.002760	0.234919	0.000654	None	534.410700	15.157000	0.949000

Fig 9. Training Results

As we can see both the training loss and validation losses are decreasing with increasing number of epochs indicating proper training of our model. We have use Hamming Loss metric for Loss Computation.

Hamming Loss is defined as how many times on average, the relevance of an example to a class label is incorrectly predicted. Therefore, hamming loss considers the prediction error (an incorrect label is predicted) and missing error (a relevant label not predicted), normalized over total number of classes and total number of examples. **Smaller the value of Hamming Loss better is the performance of our model. We can see a decreasing Hamming Loss with every passing epoch [4].** The equation is defined as:

$$Hamming\ Loss = \frac{1}{nL}\sum_{i=1}^{n}\sum_{j=1}^{L} I(\, y_i^j \ \neq \ \hat{y}_i^j \,)$$

where n is the number of datapoints (in our case 54000), L is the number of labels (in our case it is again 54000), I is the indicator function and y_i^j indicates true label and $\hat{y}_i^j$ indicates predicted label.

From the equation we can understand that this metric can be used for a multilabel classification task in our case Ecommerce Recommendation task.

And below is the performance on the validation split.

```
Out[25]:
        {'eval_loss': 0.002760367002338171,
         'eval_f1_score': 0.23491857326905963,
         'eval_hamming_loss': 0.000653968633371362,
         'eval_roc_auc': None,
         'eval_runtime': 533.6055,
         'eval_samples_per_second': 15.18,
         'eval_steps_per_second': 0.95,
         'epoch': 12.0}
```

Fig 10. Validation Results

4. RESULT AND ANALYSIS

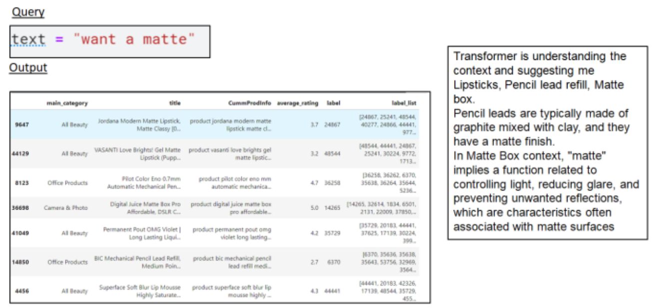


Fig 11. Result 1

The model is able to identify the context even from a poorly constructed input string but it is able to recommend good enough products.

Fig 12. Result 2

This example demonstrates a casual conversation a person is trying to have with the system where he is providing his situation and the model is able to figure out what products to recommend.

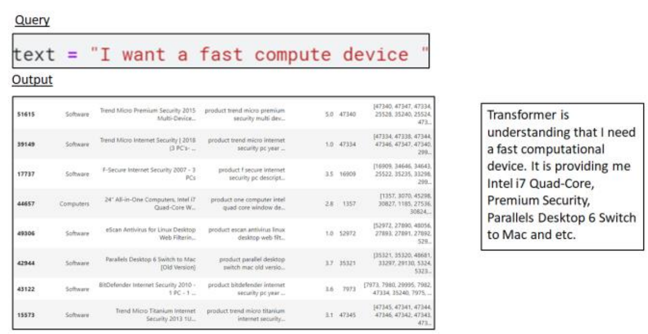


Fig 13. Result 3

This example illustrates that the model has identified what fast might mean, it is recommending faster Processor and even security products as we know that with enhanced security the system is bound to work faster. We can see how beautifully the context is captured in this case.

Query

```
text = "Want a gas and cooker"
```

Output

	main_category	title	CummProdInfo	average_rating	label	label_list
47528	Appliances	WindMax Euro Style 30 in Stainless Steel 5 Bur...	product windmax euro style stainless steel bur...	2.9	50780	[50780, 1659, 1658, 1285, 19737, 9139, 22981, ...
25250	Appliances	Hotfield 12 Inch Gas Cooktop, Built-in 2 High ...	product hotfield inch gas cooktop built high e...	5.0	22981	[22981, 1732, 1640, 1285, 19737, 19738, 1639, ...
26393	Appliances	30 inch Gas Cooktop Gas Stove 5 Burners Stainl...	product inch gas cooktop gas stove burner stai...	3.8	1659	[1659, 1658, 50780, 1285, 22981, 19737, 9139, ...
21511	Appliances	23 Inch Gas Cooktop 4 Burners Gas Stove gas ho...	product inch gas cooktop burner gas stove gas ...	2.4	1285	[1285, 22981, 1640, 33537, 19738, 1732, 1658, ...
23792	Appliances	Gas Cooktop 24 Inch, GASLAND Chef GH60SF 4 Bur...	product gas cooktop inch gasland chef gh sf bu...	4.1	19737	[19737, 1320, 22981, 19736, 1732, 1658, 1659, ...
33556	Appliances	30 inch Gas Cooktop Built-in Gas Stoves NG Gas...	product inch gas cooktop built gas stove ng ga...	3.8	1658	[1658, 1659, 50780, 1285, 22981, 19737, 9139, ...
49443	Appliances	34 Inch Gas Cooktop OM501G, Tempered Glass Gas...	product inch gas cooktop om g tempered glass g...	3.5	1732	[1732, 1320, 22981, 1285, 19737, 19738, 1640, ...
7620	Appliances	Gas Cooktop 12 Inch, 2 Burners Gas Stovetop Bu...	product gas cooktop inch burner gas stovetop b...	4.4	19735	[19735, 22981, 522, 1285, 1658, 9139, 1320, 16...
43681	Appliances	30 Inch Gas Cooktop 5 Burners Gas Stove Gas Ho...	product inch gas cooktop burner gas stove gas ...	3.9	1640	[1640, 22981, 19738, 1285, 1732, 33537, 1325,

Transformer is understanding the context and suggesting me something on which I can cook.

Fig 14. Result 4

This example is also a great one, in our dataset we do not have any specific item like gas or cooker, they are totally separate items but the model is understanding the context and as a recommendation giving cooktop as response something one uses to cook on like a stove/ cooktop etc., it might be possible that the user wanted a cooktop but could not form the English sentence properly, the model picked up that subtle context as well.

These are empirical evidence that suggests that this Novel Foundational approach can be used to build any E-Commerce Recommendation System, as a matter-of-fact Recommendation Systems across any domain.

5. FUTURE SCOPE

(1) We are working on a novel Loss function so that the model learns quicker and better.

(2) We are also planning to incorporate user preferences as well so that on top of the model's rich embedding space we can have a fusion technique where one model gives embedding for a product and another model gives a rich user representation based on user's purchase history, favorites, dwell time and many more nuanced behavior patterns.

(3) Once User based implementation is attained we can even go for Contextual Bandits to further fine tune the recommendation.

6. ACKNOWLEDGMENTS

Our sincere gratitude to Professor Raghuram Bharadwaj for his selfless support throughout the experimentation, also a special vote of thanks to the TAs who helped us whenever we asked for. Lastly, special thanks to our parents, well-wishers and friends.

7. REFERENCES

[1] Amazon Reviews'23 from UCSD
Amazon Reviews'23 (amazon-reviews-2023.github.io).

[2] DistilBERT Model Description
(Large Language Models: DistilBERT — Smaller, Faster, Cheaper and Lighter | by Vyacheslav Efimov | Towards Data Science)

[3] Attention is All you need Paper from Google
https://arxiv.org/abs/1706.03762

[4] Metrics for Multilabel Classification
https://mmuratarat.github.io/2020-01-25/multilabel_classification_metrics

[5] Hugging Face Documentation on BERT, Tokenizer, code help
https://huggingface.co/docs/transformers/en/model_doc/bert

[6] An Improved Text Sentiment Classification Model Using TF-IDF and Next Word Negation
https://arxiv.org/pdf/1806.06407

[7] Distillation of BERT-Like Models: The Theory
https://towardsdatascience.com/distillation-of-bert-like-models-the-theory-32e19a02641f