# Multi-Modalities Recall: E-Commerce Platform

YASSER OTIEFY, Universität Passau, DE
AYMEN CHAABANI, Universität Passau, DE
SEYEDEHSARA HASHEMITARI, Universität Passau, DE

## 1 ABSTRACT

Personalized information retrievals systems are of great need nowadays especially with the huge explosion when it comes to Data content that is being investigated and the need for precise results which can be reflected by the huge investments that companies are reserving for this field in order to do research and to improve the existing information retrievals solutions. In this report, we are going to discuss two variants of deep learning models solutions for information retrieval problems trained on different data sizes going from 10.000 to 1 million Query samples and based on the usage of Fasttext, Glove, and Bert pre-trained models. In fact, the first variant is using only Fasttext and Glove, while the second variant uses FastText, Glove, and Bert pre-trained models. The two presented models variants use moreover CNN network with the usage of many layers such as LSTM or Dense layers and provided as results for example trained on 100.000 as training sample size 41% and 43% for the Model with Bert and 39% and 42% for the model without Bert on respectively Test A and Test B. In addition, other processes are being presented such as Data preprocessing or Hyperparameter optimization to enhance the model's performance which will be discussed in detail in this report.

## 2 INTRODUCTION

### 2.1 Economical Background

Since 2014, the worldwide e-commerce retail sales are increasing rapidly, and in 2020 it reached 4280 billion US Dollars, while in 2014 it amounted to 1336 billion US Dollars. These revenues are projected to continue their expansion to reach 6388 Billion US Dollars(reference), figure 1 represents the worldwide E-commerce revenues distribution between 2014 and 2026[2] .The rapid growth in the worldwide E-commerce retail revenues reflects the importance of this field when it comes to boosting the economy. In fact, E-commerce, referring to the paper[10], has many benefits on the economy such as: increasing the exportation rate, increasing productivity, stabilizing the economy by taxing, encouraging competing innovation.
Another point can be reflected by this rapid growth is the complexity of this field content when it comes to the different types and categories of products and the number of different products in the same category being sold which represents a huge problem for the seller when it comes to selecting the best gain/sell ratio products and for the client in choosing the best corresponding product to buy[14].

### 2.2 Problem Context

The information retrieval system represents one of the most useful client services provided by E-commerce platforms and a subclass of the information filtering systems aiming to remove based on criteria given by the user a redundant or useless set of information. In fact, it helps the eventual buyers in choosing the best corresponding products referring to their properties. In this context, a problem of complexity and the possibility of multiple outputs or even inefficient outputs comes into light when taking into account the huge set of possible chosen products and a degree of similarity that links each set of products to others.[12]

### 2.3 Problem statement

As a solution to this problem, the Multi-modalities Recall project is being presented in the KDD Cup 2020 Contest [1] which will be explained more in the Related work section. The problem relays, as given in figure 2, on presenting a model solution that gets as inputs

[1]https://tianchi.aliyun.com/competition/entrance/231786/rankingList/Final

Authors' addresses: Yasser Otiefy, otiefy01@ads.uni-passau.de, Universität Passau, DE; Aymen Chaabani, chaaba02@ads.uni-passau.de, Universität Passau, DE; Seyedehsara Hashemitari, hashem02@ads.uni-passau.de, Universität Passau, DE.

a natural language query representing the user's products preferences, a set of 30 possible products candidates resulting from the Information retrieval phase and has to deliver as output the best 5-matching products to given Natural Language query with reference to an Evaluation process based on nDCG@5 as an evaluation metric which will be explained more during the evaluation section.

## 3 DATA DESCRIPTION

In order to build an efficient predictive model, some requirements have to be matched by the input datasets, from these requirements we propose[8]:

- Authenticity: The provenance of the data set must be known and must have a coherence with the data input usability field.
- Reliability: The dataset must have a correct content with that the model can train itself in a correct environment and provide reliable outputs
- Richness: The dataset must be large enough so that the model can be able to train itself on a large data set.

With respect to all the given points and in the context of the Modern E-Commerce Platform: Multimodalities Recall, the input dataset is provided by The Chinese E-commerce Leader named "Taobao" in a format of many "tab-separated values" files. The dataset is formed from a pair of queries taken from previous research of products in this E-commerce Platform and products corresponding to these queries. The dataset is divided into training, testing, and validation subsets in a way that :

- Training dataset: The training set is formed of 3 Million pairs of queries and product details.
- Testing dataset: The testing sets are divided into two subsets named "Test_A" and "Test_B", consisting of 1000 pairs of queries and product details. The "Test_A" was given to the challenge competitors to test their models and "Test_B" was kept secret to classify after the challenge competitors in the competition.
- Validation set: The validation set consists of 500 queries that will be fed to the model to test its performance.

Each dataset from the announced previously excepting the Validation set is formed from 9 columns(Columns information enclosed in the appendix) separated by a tabulation as shown in figure 3. Figures 4 shows the distribution of the Query Length and Class Labels in the given Training Data set having 3 million query and products pairs.

## 4 RELATED WORK

The proposed problem was already a subject of the KDD Cup 2020 Challenges for Modern E-Commerce Platform: Multimodalities Recall [2] in which many teams were competing for providing solutions for this problem. We can enumerate from these teams: The competition winners 'WinnieTheBest'[3] with an MCAN, Visual Bert based solutions and the 8-th position Ranked team 'Ai-Light'[4] with

[2]https://tianchi.aliyun.com/competition/entrance/231786/information
[3]https://github.com/steven95421/KDD_WinnieTheBest
[4]https://github.com/Ai-Light/KDD2020Multimodalities

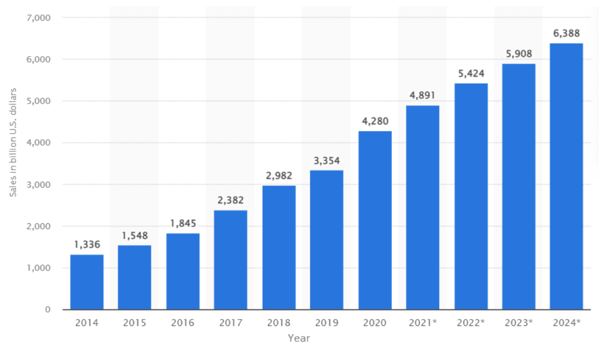


Fig. 1. Worldwide E-commerce sales distribution [2][3]

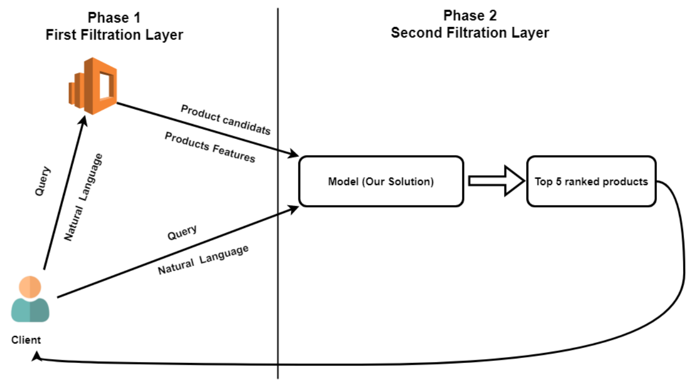


Fig. 2. Recommendation System

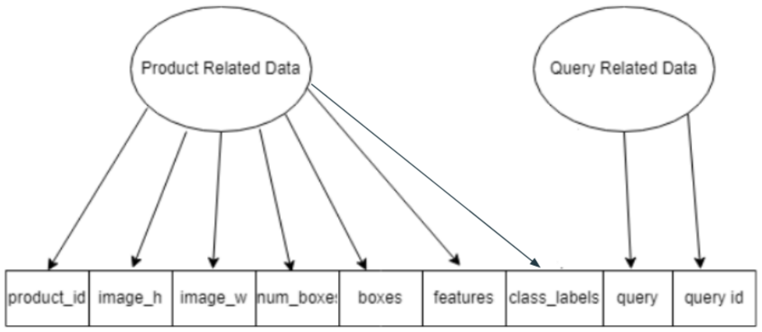


Fig. 3. Data Description

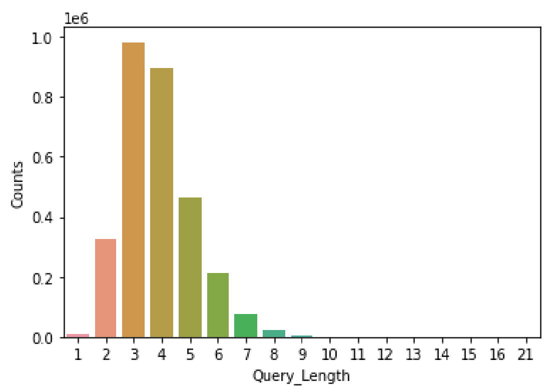
a baseline solution from which we got our inspiration to provide the project's solution. For more information about the Challenge leaderboard visit the tianchi aliyun web page [5]

## 5 DATA PREPROCESSING

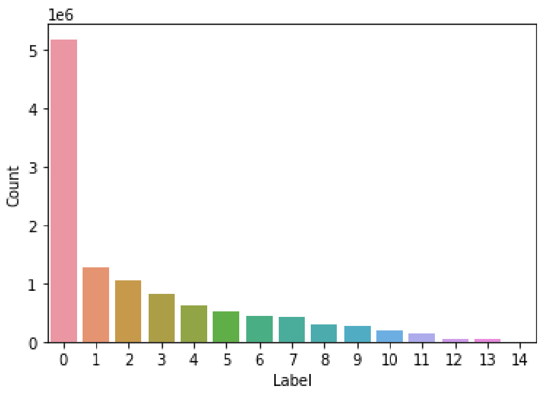
Data preprocessing represents mainly the process of transforming input data having eventually multiple anomalies types into understandable input data in order to boost the output performance of the model which is represented by its decision accuracy. Many data preprocessing steps can be used to perform transformations on a given dataset[1] from these steps there is :

- Data Cleaning: This process represents mainly the detection of the corrupted records in the input data meaning inaccurate, inconsistent, or irrelevant data, then performing the needed actions to correct them.
- Data Transformation: This process represents a crucial step in achieving the data conversion from the unprocessed to the understandable data inputs, it can be performed through mainly three sub-steps: data normalization, aggregation, and generalization.
- Data Reduction: This process helps in reducing the amount of model input data by transforming it into an ordered and simplified form to enhance the model's performance.

[5]https://tianchi.aliyun.com/competition/entrance/231786/rankingList/Final



(a) Query Length Distribution



(b) Class Label Distribution

Fig. 4. Data Distribution Statistics

Referring to our Multi-modalities Recall project, the used preprocessing strategy can be divided into two main steps :

## 5.1 Query word Embedding preparation

Since many text embedding tools are being used which are Bert; FastText and Glove (Global Vectors for Word Representation), the pre-trained models have to be imported so that after we need just to call them in the Query and image features text Embedding phase.

## 5.2 Data Preprocessing

Since this phrase refers to any operation that is being processed on the Data, it is mainly divided into :

*5.2.1 Data decoding.* Data decoding refers mainly to converting the data input from an encoded base into a plain-text or any other data form which can be useful for further processing. In this context, three columns in our data set are provided in a base64 encoded string state which is “boxes”,” features” and “class labels” and need decoding.

*5.2.2 Data Tokenization.* During this step, the input data is being converted from a meaningful state into a random string of characters called a token serving as a reference to the original data. The "Bert Tokenizer"[6] library is being used with a maximum Tokenization length set to 10 with respect to the input query length distribution given in figure 4

*5.2.3 Negative sampling.* In order to enhance the model’s performance when it comes to the quality of the input data, negative sampling is being used. Negative sampling[13], as its name represents, is a technique based on selecting a subset from the input data set representing all the positive samples and modifying it in a way to generate a false data content which helps during the model training process and immunizes it during the evaluation phase. Referring to the Multi-modalities Recall project, a negative sample is being generated "on the fly", i.e each generated batch goes through a random negative sampling phase, and in each selected record in the batch, the Query gets modified in a way using to be pointed to another product.

*5.2.4 Image features position.* Each record’s features positions set is processed in a way to produce the area of the surrounding object’s box by taking as inputs top left and bottom right coordinates. This area represents one of the model’s inputs.

[6] https://huggingface.co/docs/transformers/main_classes/tokenizer

## 6 EXPERIMENT SETUP

Our challenge now is to find a way to process product-related and query features in a way to find the relevancy score between them. One way to do that is by building a deep neural network that maps these features to binary values (0,1) that classify if these features are relevant to each other or not and by optimizing this deep network on categorical cross-entropy the output from the model represents the confidence of being relevant or not and based on that we can solve our ranking problem using this confidence score. In the following part we propose 2 model variants that achieved very close nDCG@5 results, also we will list the needed hyperparameters to reproduce such a network [7].

### 6.1 Model Architectures

As shown in figures 6 and 5 the neural network is operating on 2 main towers, a tower for product features processing to construct product pool and the other tower that process the query to construct text pool, for clarification, a pool is the final output vector from a processing tower as shown in figures 5 and 6 which is inspired by [8]. After the simultaneous processing of both product and query features the network will join both pools for the final relevancy evaluation by passing the joint pools to a stack of dense layers and final layer with 2 neurons and softmax layer to predict multinomial probability distribution, this is the higher view of the network. As shown in figure there is a difference between the 2 variants which is the leveraging of BERT [5] model in query processing which adds more complexity to the model. Here are some interesting layers which are very important to construct product (Images) and query (Text) pools:

- Fasttext [6] and Glove [11] embeddings: The network leverage the representation introduced by both embedding and concatenation to represent every word with an embedding vector of 600 dimensions 300 for Fasttext and 300 for Glove.
- Bert: A transformer trained by google on billions of tokens to be used as a context-aware embedding which means the embeddings of words are not fixed as Fasttext and Glove but dynamic, it changes based on the word context in every input sentence.
- Character Embeddings Layer: This embedding layer is responsible for the character-level representation of the class labels associated with every detected object in the image associated with every product.
- Position Embeddings: This layer is a simple dense layer to represent objects position features.
- Bidirectional LSTM Layer: It is responsible for the process of different product embeddings (object, characters, and positions) to capture patterns from this huge feature pool to finally construct one pool that could be used later to predict the relevancy between both text and image pools as mentioned earlier.

For more details about model architecture please refer to the footnote link [9] and [10]

## 7 EVALUATION

### 7.1 nDCG@5

Model evaluation represents a crucial step with the importance that it has when it comes to evaluating the provided solution referring to the specifications given in the starting point. So choosing a good model evaluator that suits and can estimate the model output represents an important topic to study[4]. A ranking metric has to be used in our project context since the model output is not univalent but a set of five products. In this case, a good evaluation metric has to be provided that in a ranking, given n model outputs, the possible number of combinations is represented by n! which represents one of the criteria for choosing a good evaluation metric. The focus of this part is on the Normalized Discounted Cumulative Gain (NDCG) which has been provided as the model evaluation metric. In fact, it is used as a multiple information ranking metrics used most frequently in the web search, having two main advantages :

[7] https://github.com/yasserotiefy/Applied_AI_Lab_WiSe2021_Passau
[8] https://github.com/Ai-Light/KDD2020Multimodalities
[9] https://github.com/yasserotiefy/Applied_AI_Lab_WiSe2021_Passau/blob/main/ai-light/models/model_with_bert.png
[10] https://github.com/yasserotiefy/Applied_AI_Lab_WiSe2021_Passau/blob/main/ai-light/models/model_baseline.png

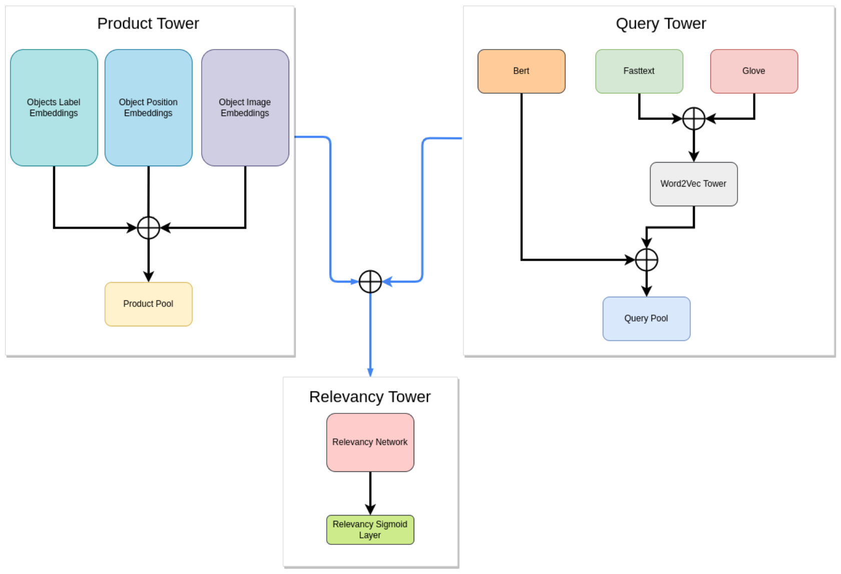


Fig. 5. Architecture with BERT

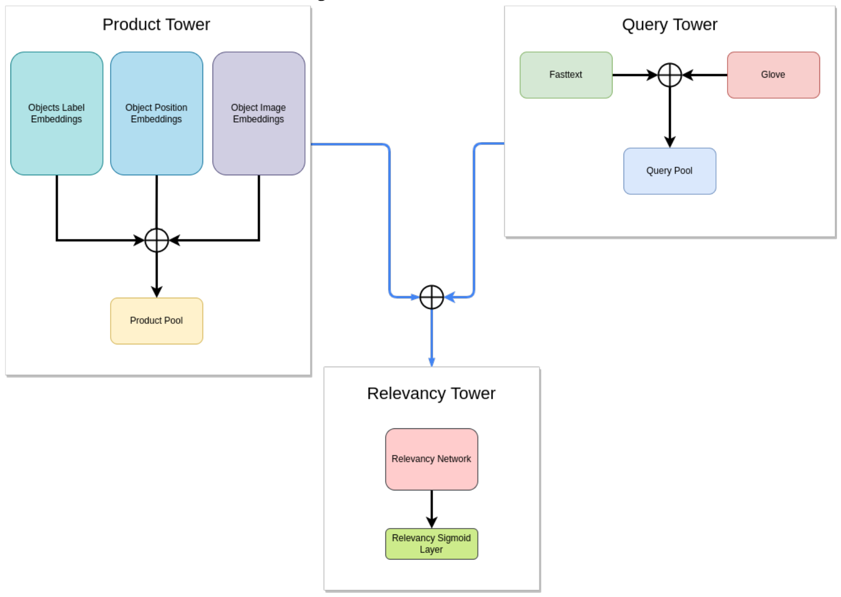


Fig. 6. Baseline Architecture

- Graded relevance: This metric supports multiple relevance levels in order to fit many input elements to evaluate.
- Discount function: The NDCG involves a discount function over the rank.

Mathematically nDCG@k is represented by three equation as it is represented below :

$$DCG = Rel_1 + \sum_{i=2}^{k} \left( \frac{Rel_i}{log_2(i)} \right) \tag{1}$$

$$IDCG = Rel_1 + \sum_{i=2}^{k} \left( \frac{Rel_i}{log_2(i)} \right) \tag{2}$$

$$nDCG = \frac{DCG}{IDCG} \tag{3}$$

Where :

- $Rel_i$: Relevance of the i-th product of the K-set.
- IDCG: Ideal DCG representing the output that the model has to provide.

## 7.2 Experiment Evaluation

**Model Performance Analysis**: As given in figures 7,8,9,10 and Table 1 , given many data input sizes, the two models variants performance increases in a good way, for example in figure 10, to go approximately on the 15-th epoch from 34% on a training set of 10 thousand samples to 43% on a training set of 100 thousand samples. Another point that can be reflected by analyzing figure 10 is the regrouping of curves for the two models variants for each training size which reflects that using Bert in our project context and in these training conditions have not helped the model's performance to increase as much as added overload when its comes to the training time, the model with bert takes compared to the model without as given in figure 9. Another point represented in figure 10 is the performance of the Hyperband hyperparameter optimizer resulted in performance which will be explained more in the Discussion section

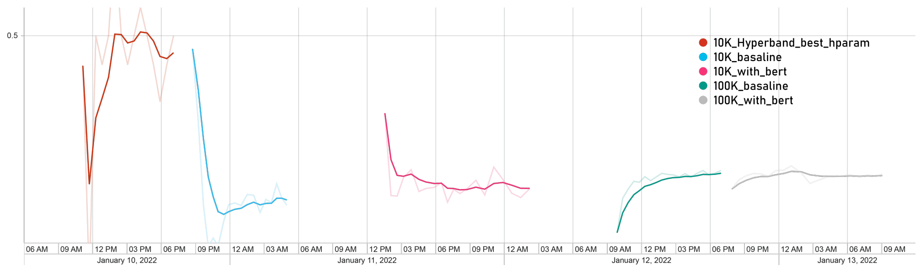


Fig. 7. Different Model Variants Accuracy Over Time

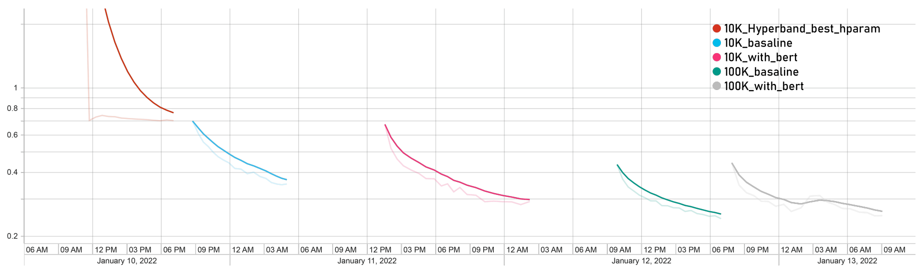


Fig. 8. Different Model Variants Binary Cross-Entropy Loss Over Time

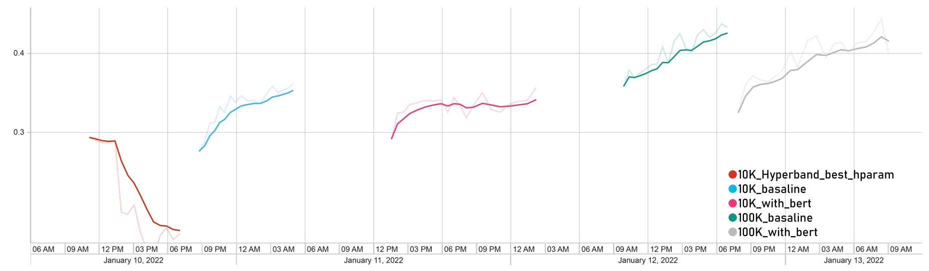


Fig. 9. Different Model Variants nDCG@5 Score Over Time

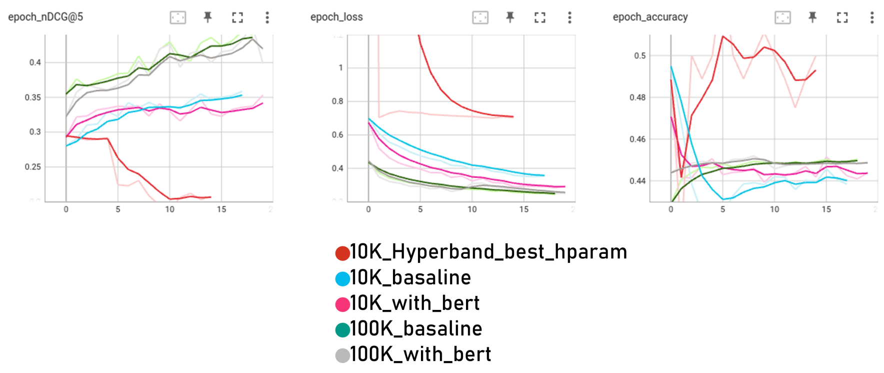


Fig. 10. nDCG@5, Loss, and Accuracy over Epochs

| ***Model/Test*** | TestA | TestB |
|---|---|---|
| Bert 1M | **0.532** | **0.544** |
| Bert 100K | *0.409* | *0.432* |
| Bert 10K | 0.338 | 0.361 |
| Without 100K | *0.392* | *0.418* |
| Without 10K | 0.326 | 0.352 |

Table 1. Shows the final evaluation on TestA and TestB of Bert/Without Bert models trained on different training samples

| activation1 | dropout_prop1 | activation3 | activation2 | dense_units5 | dense_units3 | dropout_prop2 | learning_rate | dense_units4 | activation4 | activation5 | epoch_accuracy |
|---|---|---|---|---|---|---|---|---|---|---|---|
| sigmoid | 0.6 | sigmoid | tanh | $1.79 \cdot 10^3$ | 768 | 0.8 | 0.0001 | $1.63 \cdot 10^3$ | relu | elu | 1 |
| sigmoid | 0.6 | relu | relu | $1.28 \cdot 10^3$ | $1.15 \cdot 10^3$ | 0.8 | 0.01 | 544 | sigmoid | tanh | 1 |
| elu | 0.4 | sigmoid | relu | $1.02 \cdot 10^3$ | 512 | 0.8 | 0.01 | $1.92 \cdot 10^3$ | selu | selu | 1 |
| tanh | 0.3 | relu | elu | 320 | $2.02 \cdot 10^3$ | 0.6 | $1 \cdot 10^{-5}$ | $1.47 \cdot 10^3$ | tanh | sigmoid | 1 |
| elu | 0.4 | sigmoid | relu | $1.02 \cdot 10^3$ | 512 | 0.8 | 0.01 | $1.92 \cdot 10^3$ | selu | selu | 1 |
| elu | 0.4 | sigmoid | relu | $1.02 \cdot 10^3$ | 512 | 0.8 | 0.01 | $1.92 \cdot 10^3$ | selu | selu | 1 |
| tanh | 0.2 | sigmoid | tanh | $1.66 \cdot 10^3$ | 576 | 0.9 | 0.01 | 768 | sigmoid | selu | 1 |
| elu | 0.8 | selu | selu | $1.15 \cdot 10^3$ | $1.41 \cdot 10^3$ | 0.2 | 0.01 | $1.09 \cdot 10^3$ | relu | relu | 1 |
| tanh | 0.1 | sigmoid | selu | $1.89 \cdot 10^3$ | 544 | 0.9 | 0.01 | 544 | tanh | elu | 1 |
| tanh | 0.2 | sigmoid | tanh | $1.66 \cdot 10^3$ | 576 | 0.9 | 0.01 | 768 | sigmoid | selu | 1 |

Table 2. Top-10 Provided Hyperparameters resulted from the HyperBand hyperparameter Optimizer with Validation Accuracy as Objective Function and 20K samples as a validation set.

## 8 HYPERPARAMETER OPTIMIZATION

After preparing the model and testing it using a given set of hyperparameter inputs, an important step helping in enhancing the model's efficiency is to apply a hyperparameter optimization in a way to generate the best combination from a given set of possible hyperparameter configurations to maximize a given model evaluation function, for this purpose, the Keras Tuner's "Hyperband" algorithm is being used in our project solution. This process is being applied to the baseline model since applying it on the model containing Bert will take too much training time to provide a result. The Table2 shows the best 10 models based on the validation accuracy knowing that the 20K validation samples are randomly sampled from the 3M samples training data. The main idea of "Hyperband"[7] is that given a set of finite budgets representing the set of possible model hyperparameter configuration and a "timer" the "Hyperband" has to provide a result that maximizes an objective function with respect to this timer in our case this function is the "validation accuracy". As it is represented in figures 10, 11 and after performing the hyperparameter optimization step a contradicting result has been obtained for example in place of getting a better result on the validation set, we got lower nDCG@5 scores. A possible reason for getting a such result comes from using "validation accuracy" as an objective function in place of "nDCG@5" which differs from the mathematical definition itself, adopting "nDCG@5" as an objective function represents one of the projected future works.

## 9 ENSEMBLING

In this section, we will discuss ensembling and its potential to enhance the overall system performance, in this case, nDCG@5 score. Our approach to doing ensembling is by calculating the weighted mean of the relevancy score generated from the chosen models in ensembling, in our case we will choose to work with Bert 100K and Without 100K models shown in the table 1and we will consider TestB score as our model's weight to calculate the ensembling score on TestB. Ensembling using such a method achieved 0.436 nDCG@5 which means a boosting with about 0.004, unfortunately, we have not the needed time to test this method using more models as the training process takes so much time, but it looks promising to test ensembling in this challenge as the top-ranked teams shows that there is a significant increase in nDCG@5 score by using such approach.

## 10 DISCUSSION

### 10.1 Wishes to Accomplish, Pitfalls to Avoid and Advices

At the first, TFX(TensorFlow Extended) which is an end-to-end pipeline as ML framework was so interesting to us because of some reasons:

- The ability to scale a huge data by a powerful pipeline parallelized on Apache beam.
- Generate statistics over on whole data
- Getting advantage from the benefit of TFX pipeline Modularity
- Tracking the artifacts created by the different components of TFX pipeline as shown in the figure13
- Using the TFX Model Analysis Component to perform a Deep analysis of the trained model performance on different data slices

As a summary, we decided to use this Platform to get benefits from TFX Pipeline to make it easier for us to create an efficient and robust ML pipeline using only this Platform. In the figure13, you can see how different components of TFX make a pipeline(output of each component is input for next components), check the footnote for more details about TFX [11]. Using this framework we have achieved to make the following components run:

- Example Generation: Ingests data and convert data to TF-format
- Statistics Generation: Consumes output of ExamGen and creates statistics with visualization to better understand how each feature value are distributed

[11] https://www.tensorflow.org/tfx/tutorials

- Schema Generation: It can specify data types for feature values, whether a feature has to be present in all examples, allowed value ranges, and other properties. A SchemaGen pipeline component will automatically generate a schema by inferring types, categories, and ranges from the training data.

These steps had been done against a few challenges such as converting data from tsv format to csv(ExampleGen component does not accept tsv fromat). But at the Transform component, we got blocked because of strict structures of TFX making limitations on data format and it was so complex and time-consuming for us to solve the problems due to the non-ability to integrate some useful libraries such as Pandas and Numpy within the TFX Pipeline that this platform is using as a backbone Apache beam that does not accept the usage of the above-mentioned libraries, so we spent more than two months to learn and work with this framework which wasn't enough to get comfortable with it so that is why switched after to above-explained solution. After spending the mentioned duration working with this platform, some advice can be proposed to help in dealing with such platform :

- Make sure to acquire as prerequisite analytical, Python, Big Data, Tensorflow, and Software Engineering Skills.
- Minimum devote 3 months for TFX study
- Work on Parallel pipelines (Avoid Blocking)
- If Serving is not the issue and just needs to do experiment tracking use other tools such as (MLflow, Tensorboard, …etc.)

## 11 FUTURE WORK

Based on our experience from working on this challenge we were discussing some ideas that could be considered as future work as they have a high potential to enhance the model performance. Here are some ideas that could be considered:

- Try other transformer variants trained on different languages to process both class labels and text queries.
- Use statistical heuristics to generate negative samples.
- The data class labels of objects are not balanced as shown in the figure 4 so based on that maybe investing some time to generate more data to balance, check cGAN [9]
- Doing hyperparameter optimization on nDCG@5 as evaluation function, not accuracy to directly optimize the desired optimization metric.
- Another path that could be very promising but challenging at the same time is to change the whole architecture to optimize it on the nDCG@5 directly end-to-end this also known as neural learn to rank [12]. So the output from the model in this way will not be the relevancy score of query and an associated product but the ranked list of products directly based on the input query and list of products.
- Finally, consider more advanced ensembling techniques to enhance the model performance by using more trained models at the same time but will require eventually more training time and computational resources.

## 12 CONCLUSION

Multimodalities recall for modern E-commerce platforms was a proposed problem of the KDD Cup2020. The challenge is to create a ranking system to filter the 5 most relevant products (represented by image-features, image-boxes-labels, image-boxes-pos,...) to NL query. At the first, we were hoping to be able to implement our model through a TFX pipeline[13] but we were blocked because of a lack of knowledge of this framework and the strict structure of TFX. After experimenting with different text representation variants such as Glove, Fasttext and Bert, it showed the best model is with stacking all text embeddings to get the query pool, however, the model did not have a significant difference compared to static embeddings (Glove and Fasttext). Regarding hyperparameter optimization, it seems that using validation accuracy as our objective function is not the best option to optimize the network's hyperparameters. On the other hand, ensembling showed some potential to boost the model performance, however, it needs more experiments to know its limits and what it really can do. Finally, trying to optimize the sparse categorical cross-entropy in order to solve the ranking problem

[12]https://www.tensorflow.org/ranking
[13]https://www.tensorflow.org/tfx/tutorials

indirectly is not the best way to tackle such a problem on the other hand as we suggested in the future work section. Make the network learn how to rank end-to-end by optimizing nDCG@5 directly.

## 13 TEAM MEMBERS CONTRIBUTION

### 13.1 Yasser Otiefy

First of all, I want to thank my teammates and lab staff for such great effort they have done in this lab, it was a great experience to work on such a challenge under the supervision of Prof. Dr. Florian Lemmerich and Max Klabunde. Secondly as requested by lab staff here are my contribution to this lab. Regarding the coding part [14], Worked on Converting TSV formate to CSV formate to be read by TFX ExampleGenerator then feed the generated TFrecords to Statistics Generator to generate statistics which are shown in different presentations and tried to work a little bit on TF transform but unfortunately it was not successful. Did some EDA on query and object class labels features shown in figure4. Also, experimented with different model variants with different text embeddings such as Glove, Fasttext, and Bert. Also, worked on Hyperparameter optimization using Keras tuner, then worked on tracking experiments results using Tensor board. Tried ensembling on Bert and without Bert models trained on 100K and it showed about 0.004% boosting in the final nDCG@5 metric. Regarding the scientific writing part, contributed to writing experiment setup, ensembling, and future work sections and reviewed other sections, finally helping in presentations preparation.

### 13.2 Aymen Chaabani

Regarding the coding part, I have worked first on understanding TFX as it is a new Platform to work with, how each component performs, and the baseline solution from which we got the inspiration to tackle the challenge, after this I have started working on the statistics generation component waiting for Sara's output from the example generation component and I had also to work on this component since many problems has Sara faced when dealing with it such as the conversion of data types and the attribution of the data input files to component splitter to create each split. After this, I have worked on the Transform component in which I tried to generate the negative samples "on the fly" for each batch with an output ratio of 50% for negative samples and 50% of positive samples but then we got some problems related to the data transformation which took too much time without having a useful result which pushed me to split the work among the group and I had to work on this report, on the final presentations and partially on the TFX presentation while leaving at the same time full focus and understanding on the coding part that is provided.

### 13.3 Seyedehsara Hashemitari

At the first, I spent a long time learning TFX as it was totally new for me, I watched a lot of Coursera training videos for preprocessing, transformation(feature engineering). I had almost finished Tunning component that we stopped working with TFX. In the coding part, I tried to do ExampleGeneration component to create TF.examples and divide input data to train test, and value parts which were problematic because of tsv format of input data, consequently I skipped this part when against a lot of effort I could not solve the challenge. So I start to write a sample for Train-module-file for the train component but we were blocked on previous steps. After an unsuccessful experience on TFX I started gathering information and preparing TFX presentation, next step was getting involved with a report in the discussion, future works, and conclusion part, I had provided some visualizations but Yasser's version was better. At the same time following code processing to understand the details of our project was my concern. Finally, I was a part of preparing a final presentation.

## REFERENCES

[1] Vivek Agarwal. 2015. Research on Data Preprocessing and Categorization Technique for Smartphone Review Analysis. (Dec. 2015). https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.741.596&rep=rep1&type=pdf

[2] Kelvin Cheng. 2021. Analysis of the development of e-commerce in the retail industry in recent years. 50, 1 (March 2021). https://doi.org/10.1145/1188913.1188915

[3] Stephanie Chevalier. 2021. Retail e-commerce sales worldwide from 2014 to 2023. (July 2021). https://doi.org/10.1145/1219092.1219093

[4] Koby Crammer and Yoram Singer. 2002. *Pranking with Ranking.* https://papers.nips.cc/paper/2001/file/5531a5834816222280f20d1ef9e95f69-Paper.pdf

[14] https://github.com/yasserotiefy/Applied_AI_Lab_WiSe2021_Passau

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]* (May 2019). http://arxiv.org/abs/1810.04805 arXiv: 1810.04805.

[6] Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. FastText.zip: Compressing text classification models. *arXiv:1612.03651 [cs]* (Dec. 2016). http://arxiv.org/abs/1612.03651 arXiv: 1612.03651.

[7] Diederik P. Kingma and Jimmy Lei Ba. 2015. ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION. (2015). https://arxiv.org/pdf/1412.6980.pdf

[8] Florencia Mangini. 2018. *MINIMUM REQUIREMENTS FOR A DATASET.* https://www.thinkingondata.com/minimum-requirements-for-a-dataset/

[9] Mehdi Mirza and Simon Osindero. 2014. Conditional Generative Adversarial Nets. *CoRR* abs/1411.1784 (2014). arXiv:1411.1784 http://arxiv.org/abs/1411.1784

[10] Aleksandar Parishev, Goran Hristovski, Petar Jolakoski, and Viktor Stojkoski. 2020. E-COMMERCE IMPACT ON ECONOMIC GROWTH. 50 (Nov. 2020). https://doi.org/10.1145/1188913.1188915

[11] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).* Association for Computational Linguistics, Doha, Qatar, 1532–1543. https://doi.org/10.3115/v1/D14-1162

[12] Pradeep Kumar Singh, Pijush Kanti Dutta Pramanik, Avick Kumar Dey, and Prasenjit Choudhury. 2021. . https://www.researchgate.net/publication/339172772_Recommender_Systems_An_Overview_Research_Trends_and_Future_Directions

[13] Jason Tam. 2018. Negative Sampling (in Numpy). *HBC TECH* (March 2018). https://tech.hbc.com/2018-03-23-negative-sampling-in-numpy.html

[14] Yingyi Wu, Haiquan Chen, and Huan Wang. 2019. *The Influence of Product Diversity on Consumers' Impulsive Purchase in Online Shopping Environment.* https://doi.org/10.4236/ajibm.2019.93046

## 14 APPENDIX

### 14.1 Data Description Details

(1) product_id: This column represents the index of the product
(2) image_h: This column represents the height of the product image
(3) image_w: This column represents the width of the product image
(4) num_boxes: This column represents the number of detected object bounding boxes for the image
(5) boxes: This column represents a [num_boxes, 4] shaped 2-D array specifying the location of each object bounding box in the image. For each row of this array, the 4 values specify the top/left/bottom/right coordinates of the corresponding box. This array is encoded into a base64 string,
(6) features: This column represents a [num_boxes, 2048] shaped 2-D array specifying the 2048-dimension feature computed by the detector of each object bounding box in the image. This array is also encoded into a base64 string
(7) class_labels: This column represents a [num_boxes] shaped 1-D array specifying the category-id of each object. There are 33 object classification categories in this dataset. The correspondence between each category-id and its category name is given in multimodal_labels.txt.
(8) query: This column represents a natural language query that matches the corresponding product
(9) query_id: This column represents the index of the query

### 14.2 Hyperparameter optimization figures

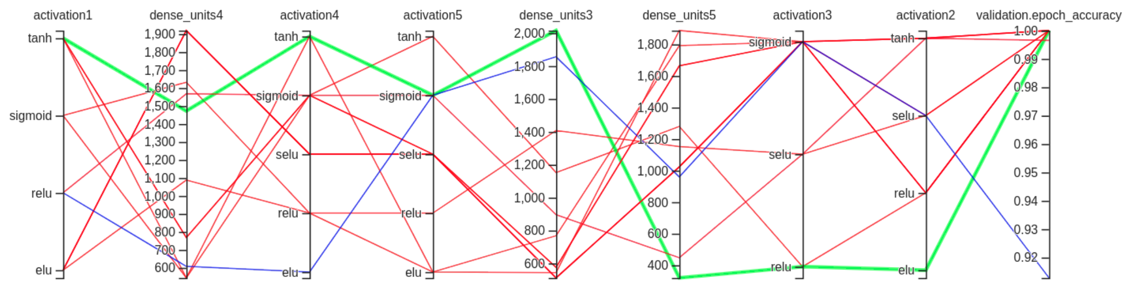


Fig. 11. Best Parameters Paths

### 14.3 TFX pipeline

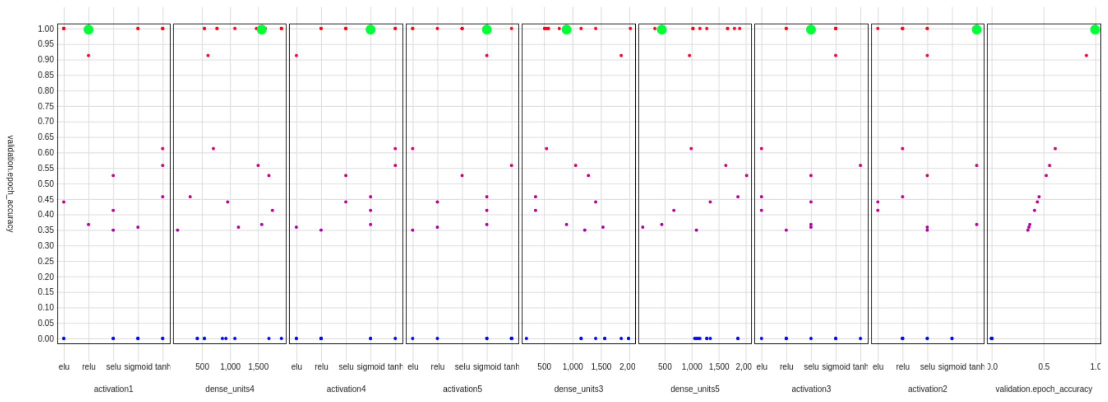


Fig. 12. Another view for Best Parameters

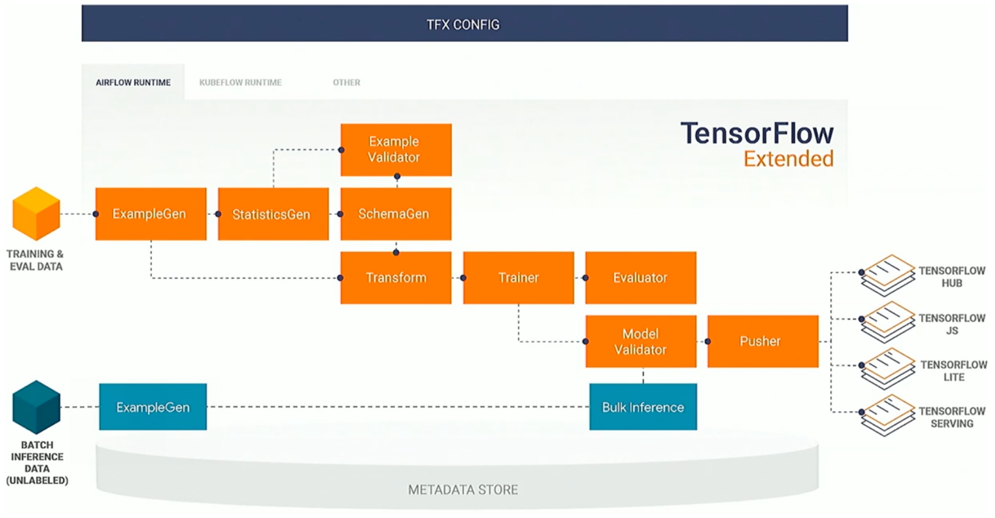


Fig. 13. All TFX Pipeline Components that can be used to produce an ML Pipeline