19 JULY 2021

ZEVI NLP PROJECT HEADSTART

CONTEXTUAL SEARCH

Vibhansh Gupta
vibhanshg@gmail.com

TASK SUBMISSION REPORT

Zevi is building the future of search. The task at hand was appropriately related to building one such possibility, an intelligent search which would understand the context of each query and use the various features in the provided dataset to filter out relevant results.

In the following report, I provide my thoughts on the task at hand, the dataset provided for it and the procedure I chose for delivering the solution and handling the problems I faced throughout. The past few days have been filled with me constantly brainstorming on how to make an impactful change at basic search techniques without having to compromise on the UX and try to not follow any previously laid out paths. In the following sections, I will try to closely follow the format you mentioned in the document for the project.

EXPLORATORY DATA ANALYSIS (EDA)

The provided dataset consisted of product information from the BestBuy US website. There were 14 different columns and the features were ranging from the category and popularity of the product to it's metadata like a sample image or the URL of the product.

However, simply having multiple features doesn't make a dataset good for all tasks. The objective of the usage plays a very impactful role in this judgement and sometimes you end up wishing for more 'relevant' features.

This was certainly the case for this assessment. For building the solution, I was often left with this feeling for wanting more features. A timestamp of the product release date would've been helpful, while the rating data was flawed with mostly 4 star ratings. It would've been helpful if these ratings weren't rounded off and left at their floating point values.

The first step I took after exploring the dataset was to bring out my trusty duo of notebook and pen. I jotted down features that I thought of as important for the task at hand from a user's point of view. You can see the table I created for the categories (features) in the figure.

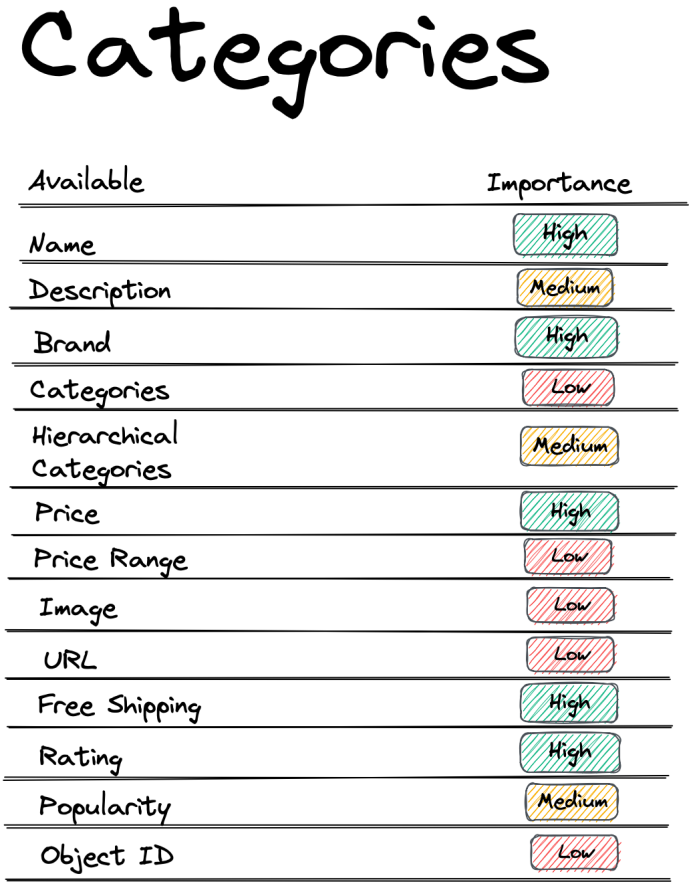


Figure : Importance of the features in dataset

Since the dataset involved a lot of stop words and symbols that would make life difficult for training any solution, I decided to format this data into simpler alphanumeric texts using a RegEx and clear out any stop words in the sentences.

Right from the start, I wanted to build a solution that could be re-trained with the usage data just like in the real world scenario where user queries are expected to go multi-fold after the launch of such a search. So, in order to reach peak performance, the quantity of data was not as important as the quality or rather 'variety' of data. The aim for peak performance is viable only if a machine is prepared to handle all the different possibilities of corner cases and that in turn is very difficult to make happen.

As such, traditional Seq2Seq models on training data were of no use. In fact, most Deep Learning Solutions can never reach such a high accuracy from a single training. So, the only simple solution for this problem is to make your model re-trainable, especially with the queries that an actual user is making so that all corner cases can be covered.

SOLUTION EXPLANATION

The first thing to consider while building this solution was of course a baseline search database where simple matching could be done as efficiently as possible. This was the reason why I chose to index all of the dataset data inside a local ElasticSearch server. Not only is it efficient, it can also use ML Solutions to supercharge its search queries to filter out relevant results.

Once all of the data had been appropriately indexed, it was time to start trying out possible solutions to extract context from the query and improve relevancy. I tried out various solutions from Product Type Clustering to Semantic similarity prediction. But, after thinking from the User Experience side of things, I realised that running multiple algorithms would make it difficult to maintain efficiency.

Hence, a single algorithm was required that could intelligently capture what was being talked about in the query. This job felt best suited for an Entity Recognition algorithm. I started with using spacy's NER tools but soon found out that they weren't of much use for my requirements. This was the case for almost all of the solutions I tried.

The message was clear. A custom Named Entity Recognition (NER) technique was needed. The ideas was to use the provided dataset for it. A little extra time was needed to prepare the training data (especially annotating the dataset). I tried out building both, a custom spacy NER and a NER from scratch using LSTM layers in a Recurrent Neural network with help online. I also referenced a few published papers.

At the end of it all, I landed on using neither for this task. While spacy was good in retrain ability aspects, it lacked the ease of building a model that TensorFlow and Keras provided. Neither of these were fitting the requirement and so, something in between was required.

Hence, I used MonkeyLearn to annotate, train and re-train predictions for corner cases. MonkeyLearn is a free tool that you can use to custom build models and keep a check on their performances (Ease of usage). You can also choose to view previous prediction and annotate them manually for better learning on the corner cases (Retrain Ability).

From, the important features in the dataset, the following NER Tags were extracted. The 'name' column was used as the training data.

By no means was the model perfect. It had a lot of bias for the way in which names were written in the dataset.
An example prediction is given below :

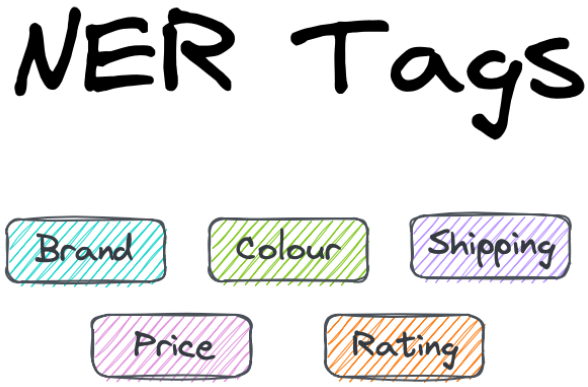


Figure : The custom tags available

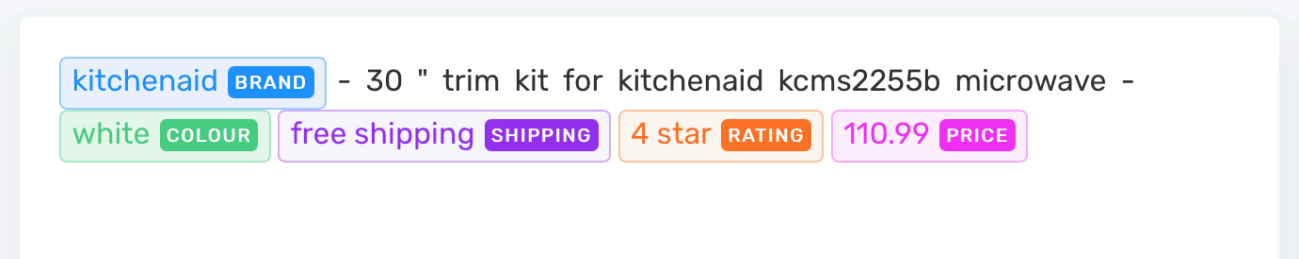


Figure : Running NER on a product

Since I was thinking of building the solution from an API point of view, I always had the pipeline plan needed for the switch from the Jupyter notebook to FastAPI at the back of my mind. Once I was done with the NER, the only thing left was to write custom queries on top of ElasticSearch. Here is an example of the different filters I included :

```
In [314]:
# AVAILABLE FILTERS : BRAND, COLOUR, FREE SHIPPING, RATING & PRICE RANGE
filters = []
if prices['gte_flag']:
    filters.append({ "range": { "price": { "gte": prices['gte_amount'] }}})
if prices['lte_flag']:
    filters.append({ "range": { "price": { "lte": prices['lte_amount'] }}})
if brand is not None :
    filters.append({ "match": { "brand": brand }})
if rating is not None :
    filters.append({ "term": { "rating": rating }})
if shipping is not None :
    filters.append({ "term": { "free_shipping": True }})

musts = [{ "match": {"name": search_term} }]
if colour is not None:
    musts.append({ "match": {"name": colour} })
```

Figure : Adding NER based filters to search

One major problem that I faced was that the NER was not good at detecting prices and the context of those price mentions. It was quintessential for the solution to understand what price range a user meant when they mentioned any price.
A simple 'more than' or 'less than' can change the whole of the results.
In order to achieve this contextual filtering, I had to write a custom function from scratch that would understand if a price was mentioned in the search query and extract its context from the nearby words.
The result included what amount was mentioned and whether it was mentioned for the greater than or the lower than context. Below is an example of how it worked :

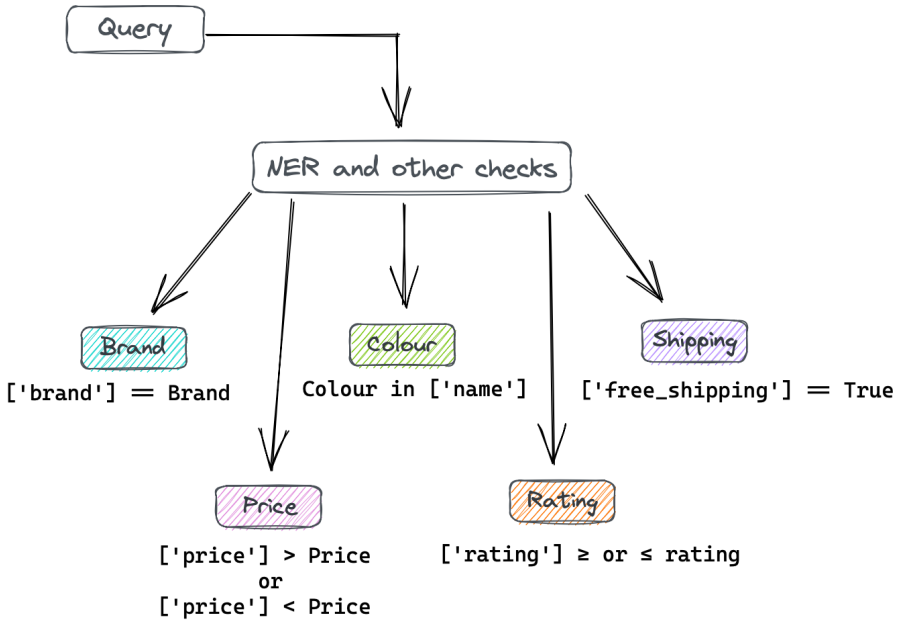
```
In [303]:
l = "Dell laptop 512 GB with free shipping under 500.00 $ but more than 250"
res = price_filter(l)
print(res)

{'gte_flag': True, 'gte_amount': '250', 'lte_flag': True, 'lte_amount': '500.00'}
```

Figure : Successfully extracting context of price mentions

As you can see, the result is an extremely valuable dictionary of contextual information. The rest of the code can, of course, be seen in the Jupyter Notebook.

To recap, here is the idea behind the full process :



SOME EXAMPLE SCREENSHOTS

EXAMPLE 1 : “HP Laptops with 8GB RAM under 500$”

```
QUERY WAS :  HP laptops with 8GB RAM silver under 500.00

NAME :  HP - EliteBook 14" Refurbished Laptop - Intel Core i5 - 8GB Memory - 500GB Hard Drive - Silver  RATI
NG :  2 PRICE :  239.99 FREE SHIP :  True

NAME :  HP - EliteBook 14.1" Refurbished Laptop - Intel Core i5 - 8GB Memory - 500GB Hard Drive - Silver  RA
TING :  1 PRICE :  243.99 FREE SHIP :  True

NAME :  HP - EliteBook 14" Refurbished Laptop - Intel Core i5 - 8GB Memory - 500GB Hard Drive - Silver  RATI
NG :  3 PRICE :  310.99 FREE SHIP :  True

NAME :  HP - EliteBook 11.6" Refurbished Laptop - Intel Core i5 - 8GB Memory - 750GB Hard Drive - Silver  RA
TING :  0 PRICE :  330.99 FREE SHIP :  True

NAME :  HP - EliteBook 12.5" Refurbished Laptop - Intel Core i5 - 8GB Memory - 750GB Hard Drive - Silver  RA
TING :  4 PRICE :  310.99 FREE SHIP :  True

NAME :  HP - EliteBook 15.6" Refurbished Laptop - Intel Core i7 - 8GB Memory - 240GB Solid State Drive - Sil
ver  RATING :  4 PRICE :  429.99 FREE SHIP :  True

NAME :  HP - EliteBook 11.6" Refurbished Laptop - Intel Core i5 - 8GB Memory - 128GB Solid State Drive - Sil
ver  RATING :  1 PRICE :  319.99 FREE SHIP :  True

NAME :  HP - EliteBook 15.6" Refurbished Laptop - Intel Core i5 - 8GB Memory - 240GB Solid State Drive - Sil
ver  RATING :  0 PRICE :  359.99 FREE SHIP :  True
```

EXAMPLE 2 : “Black iPhone lightning cable”

```
QUERY WAS :  Black iPhone lightning cable

NAME :  ReTrak - 3.2' Lightning USB Charging Cable - Black  RATING :  4 PRICE :  14.99

NAME :  StarTech.com - 3.3' Lightning USB Charging Cable - Black  RATING :  4 PRICE :  31.99

NAME :  Apple - iPhone® Lightning Dock - Black  RATING :  4 PRICE :  49.99

NAME :  Griffin Technology - 3' USB-to-Lightning Cable - Black  RATING :  4 PRICE :  14.99

NAME :  mophie - 4.8' Micro USB-to-Lightning Cable - Black  RATING :  4 PRICE :  29.99

NAME :  Belkin - MIXIT 4' Metallic Lightning to USB Cable - Black  RATING :  4 PRICE :  24.99

NAME :  ReTrak - 10' Retractable Lightning Charge-and-Sync Cable - Black  RATING :  5 PRICE :  29.99

NAME :  AudioQuest - Carbon 2.5' USB-to-Lightning Cable - Black/Gray  RATING :  5 PRICE :  129.99

NAME :  Mobile Undead - 4.9' Lightning USB Charging Cable - White/Black  RATING :  0 PRICE :  19.99

NAME :  Modal - Apple MFi Certified 4' Braided Lightning Cable - Red/Black  RATING :  4 PRICE :  19.99
```

EXAMPLE 3 : “Pink iPhone cover”

```
QUERY WAS :  pink iPhone cover

NAME :  OtterBox - Defender Series Protective Cover for Apple iPhone SE - Pink  RATING :  4 PRICE :  49.99

NAME :  ADOPTED - Back Cover for Apple iPhone 6 and 6s - Pink  RATING :  4 PRICE :  39.99

NAME :  Hello Kitty - Polycarbonate Cover for Apple® iPhone® 5 - Black/Pink  RATING :  3 PRICE :  19.99

NAME :  Incipio - PERFORMANCE Back Cover for Samsung Galaxy S7 - Pink  RATING :  5 PRICE :  24.99

NAME :  Incipio - STOWAWAY Back Cover for Samsung Galaxy S7 - Pink  RATING :  4 PRICE :  27.99

NAME :  Apple - Smart Cover for iPad mini 4 - Pink Sand  RATING :  4 PRICE :  39.99

NAME :  Apple - Smart Cover for iPad mini 4 - Light Pink  RATING :  4 PRICE :  39.99

NAME :  Apple - Smart Cover for 9.7-inch iPad Pro - Pink Sand  RATING :  4 PRICE :  49.99

NAME :  Apple - Silicone Cover for 9.7-inch iPad Pro - Light Pink  RATING :  4 PRICE :  69.99

NAME :  Apple - Silicone Cover for 9.7-inch iPad Pro - Pink Sand  RATING :  4 PRICE :  69.99
```

EXAMPLE 4 : “Latest TVs” (EDGE CASE)

In [372]:
```
print("QUERY WAS : ", query,"\n")
body = {"query": {"bool": {"must": musts, # to show must needed results
                           "filter": filters # to show relevant results
                           }     }    }

res = es.search(index='zevi', body=body)
for doc in res['hits']['hits']:
    print("NAME : ",doc['_source']['name']," RATING : ",doc['_source']['rating'],
                "PRICE : ",doc['_source']['price'],"\n")
```

```
QUERY WAS :  latest TV under 1000$
```

In [373]:
```
latest
```

```
'2021'
```

Since there were no TVs to show from the current year, there are no results.
As you can see the filter, ‘latest’ corresponds to the current year.

EXAMPLE 5 : “Products to keep the body warm”

```
QUERY WAS :  products to keep the body warm

NAME :  Omron - Full Body Sensor Body Composition Monitor with Scale - Black  RATING :  4 PRICE :  78.99

NAME :  Warm House - Oslo Electric Fireplace - Black  RATING :  5 PRICE :  299.99

NAME :  Honeywell - 1 Gal. Warm Moisture Humidifier - Black  RATING :  4 PRICE :  29.99

NAME :  Warm House - Bern Floorstanding Electric Fireplace - Black  RATING :  5 PRICE :  129.99

NAME :  Optimus - 1.5-Gal. Warm Mist Humidifier - White  RATING :  3 PRICE :  45.99

NAME :  Optimus - 1.7-Gal. Warm Mist Humidifier - White  RATING :  1 PRICE :  48.99

NAME :  Withings - Body Cardio Scale - Black  RATING :  4 PRICE :  179.95

NAME :  Withings - Smart Body Analyzer - Black  RATING :  4 PRICE :  149.99

NAME :  Crane - 1 Gal. Ultrasonic Warm Mist Humidifier - Slate  RATING :  3 PRICE :  49.99

NAME :  Crane - 1 Gal. Portable Warm Mist Humidifier - Aqua  RATING :  3 PRICE :  49.99
```