CA684I Machine Learning Assignment
Zalando Product Matching Challenge

Cheong Hao Io
MSc. Artificial Intelligence
Dublin City University
Dublin, Ireland
hao.cheong3@mail.dcu.ie

ABSTRACT
In modern era, customer buying behavior has emerged from physical store purchase to now shopping from digital devices via use of internet technology. All the advanced e-commerce technology are now fulfilling needs effortlessly by perform a finger click on the digital devices and all data in relation to buying, browsing and transection recorded to company database in real time. To fulfil the market demands, many advanced e-commerce platforms have implemented the product matching strategy to customers and recommends any desired product in cheapest offer price. This strategy can bring positive outcomes to both customers and retailers, it can directly increase sale volume, customer retention rate, as provided function can quickly sort out customer's desired product in best offer in matter of seconds.

In this paper, we are going to discuss, explore and leverage several machine learning techniques to resolve a real-world Zalando product matching problem. There are several datasets provided which include offers training, offers testing, the scope is to find matching product models that can effectively locate any common sell items between "Zalando" and "AboutYou", the outcome is targeted to enhance the overall F1 matching score.

Keywords
"TF-IDF", "Cosine Similarity", "NGrams", "String Matching", "Product Matching", "Text Matching"

1. INTRODUCTION
Product matching is generally referring to an automated learning technique which able to look up exact same or similar items against range of known items. This is not only commonly used in retail, but also applicable on other business usages, such as market price benchmarking, competitor analysis, product recommendation engine etc.

Matching algorithm falls under a subcategory of Deep Learning model "Natural Language Processing", this model is flexible and able to apply in wide range of product applications.

Nowadays, items on different e-commerce platforms are commonly segmented by order id, title, image, price, product description, shop name, etc. Product matching is a technique commonly used in e-commerce websites and this deep learning technique can denote matching products by different retailers within search option.

This strategy is favorable to both end users and e-commerce retailer. For customers, this creates the flexibility for choosing best price offer compared to all available selling options, it is cost efficient and intelligent buying. As for retailer, this is not only able to establish rational price ratio as well as to improve the business competitive power, by based on item diversity and price range, retailers can learn from competitors and create better marketing strategy to achieve long term business success. [1]

There are often duplicated items within dataset are sold by different retailer and under different product description name, this insight can easily differentiate by human manual review, but length of action can be extremely long, and it is not realistic for real world business as tasks are needed to be completed within short timing period, and that is why machine learning can be an effective and powerful solution to modern businesses.

In this challenge, we are going to leverage the occurrence of text data and computational power to highlight any matching sell items which under specific retail seller and comparing text similarity under matched item title.

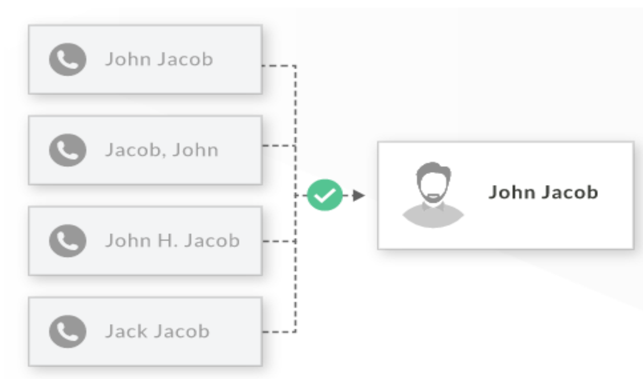

Figure 1. Matching algorithm example

2. LITERATURE REVIEW
Application which relates to Natural language processing (NLP) are now everywhere, it is now critically important to the modern information age. People are closely communicating with each other 95% with different languages and all the text or audio data are generated from the use of internet search, emails, language translation, virtual personal assistant, daily medical device, customer service chatbot etc.

Natural language processing (NLP) is a sub category of Machine Learning/Artificial Intelligence which focusing on the computational understanding to process human language, it is an unsupervised learning technique which able to predict word meaning based on provided information, this technique has rooted from the year of 1950s, this is started from a published article "Computing Machinery and Intelligence" as known as Turing test that established by Alan Turing, and it was the initial phrase of calling artificial intelligence in action, this test involves many tasks that resolved by computational interpretation and basic natural language processing techniques. [2]

The evolution of Natural language processing (NLP) can simply segment as three stages:

Symbolic *(1950 - early 1990s)*

The initial ideology of symbolic in Natural language processing (NLP) was founded by John Searle, based on the popular Chinese room exercise [3], this experiment is conducted by a set of rules with a phrasebook, questions and potential matching answers in Chinese, computer will then automatically apply NLP techniques by applying set of rules to the data import from.

Statistical *(1990-2010s)*

Up until the mid-1980s, majority of Natural language processing (NLP) models were conducted by many complex manual written rules. In the early 1990s, due to the steadily increase in changing and emerging computational power, corpus linguistic techniques have emerged to the market, it can quickly speed up time when processing big data, it is a rapidly evolving and powerful methodology by leverage the use of statistical analysis to provide large collection of text or audio data to calculate the linguistic occurrence. [4]

Neural (*2010-Present)*

In recent years, there is further development in Natural language processing (NLP), computed models can now action in wide range of tasks, thanks to combination usage of feature learning and deep neural network are emerging, we can now be able to develop complex models with multiple neural layers with the ability to identify detailed data features with high accuracy rate, including functions such as machine translation, text to audio reading, product matching, voice recognition etc. [5]

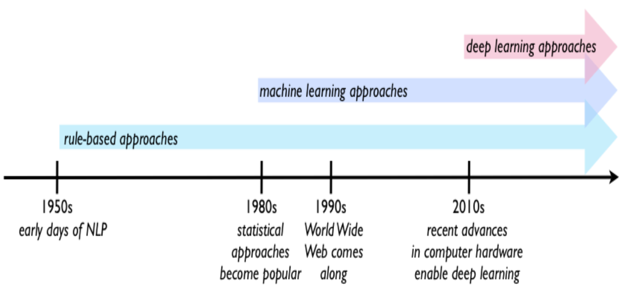

Figure 3. Natural Language Processing Evolution Timeline

3. DATASET

Zalando had provided two datasets to conduct the challenge:

Offers_training	102884 rows and 10 columns
Offers_test	106741 rows and 10 columns

Two datasets are clean, without any missing values, each is respectively split into 70/30 portion for training and testing purpose.

Label	Description
offer_id	unique identifier for an offer of a product (i.e. a product x shop combination, where
shop	"zalando", "aboutyou"
lang	"de" (German)
brand	e.g. "Nike" - note that different shop s might have different brand nomenclature
color	e.g. "blue" - note that there could be more than one and different shop s might have
title	e.g. "White Nike tennis top"
description	a long product description that can may contain material composition, cleaning inst
price	price in euro without any discount
url	url of the product description page
image_urls	list of product images such as stock photo, with model, lifestyle photo, or close up

Figure 2. Provided Dataset columns

4. METHODOLOGY

Data preprocessing

Data preprocessing is an essential step for any data analytic and machine learning projects, by identifying any existed abnormal data which might be irrelevant, incomplete, incorrect, or missing from the working dataset, according to different conditions we can then whether delete or replace the subset, accordingly, cleaned data is not only can deliver clearer exploratory data analysis and visualization result, but it can also produce better action productivity, prediction result and reduce errors to occur.

According to both training and testing datasets provided from Zalando, there is no missing data involved but we needed to remove all excessive punctuation and meaningless characters.

```
[ ] ## Remove punctuation and meaningless characters
    import re
    def preprocess(description):
      # Actually not required as what we have is titles which usually doesn't
        description=description.lower()
        description=re.sub('[-\n\t]+',' ',description)
        description= re.sub(r"won\'t", "will not",description)
        description=re.sub(r"can\'t", "can not",description)
        description=re.sub(r"n\'t", " not",description)
        description=re.sub(r"\'re", " are",description)
        description=re.sub(r"\'s", " is",description)
        description=re.sub(r"\'d", " would",description)
        description=re.sub(r"\'ll", " will",description)
        description=re.sub(r"\'t", " not",description)
        description=re.sub(r"\'ve", " have",description)
        description=re.sub(r"\'m", " am",description)
        description=re.sub('[^a-z0-9]+',' ',description)
        description=re.sub('\s+',' ',description)
        return description.strip()

clensed_train=[preprocess(title) for title in tqdm(testing70.title.values)
```

Figure 4. Remove punctuation and meaningless characters

All punctuation and meaningless characters are now removed, and data are now cleaned and ready for training.

```
#Cleaned data that is ready for training
clensed_train

['schal',
 'plisseerock',
 'tasche',
 'winterjacke ashani puffy',
 'averie shorts stoffhose',
 'sweatshirt holly',
 'kabukipinsel nothe essential kabuki',
 'blouse solid with tape detail bluse',
 'sneaker is serendipity',
 'shorts',
 'breaker pants jogginghose',
 'shortsleeve workwear jumpsuit',
 'kleid iserena',
 'rucksack pop quiz',
 'uhr',
```

Figure 5. Cleaned data

Exploratory Data Analysis

Based on the testing data provide by Zalando, we have divided the entire dataset into 70/30 proposition, and we have identified several basic ground truths about the testing dataset.

Based on title, there are many duplicated product title combinations, and the top five items are “Shirt”, T-Shirt”, “Kleid”, “Hose”, “Hemd” & “Sweatshirt”. Indeed, some items are equally the same to human eyes but they are unique when output from a general visualization and computer is seeing them as distinct items. We can use product matching algorithms to resolve this problem.

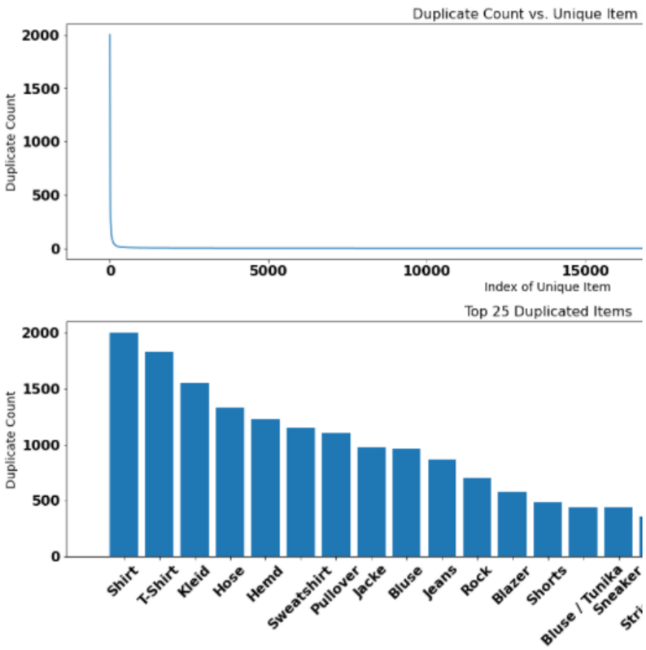

Figure 5. Top 25 duplicated items sorted by titles

To display the level of specific wording occurrence in titles, here we have computed a word cloud to exercise the outputs, and there are several commonly duplicated item names are displayed on the graph

Figure 6. Word cloud occurrence based on title

Similarity Matching

Usually, product titles are consisted of a brief description which includes the product name, and other information about size and colors, often different retailer are promoting same item under different product title, by performing product matching we can get to understand what are the matched product titles that is available on market and easy to give product recommendation to customers to choose from.

Product title similarity is a product matching technique which are commonly used in e-commerce retailer market, which allows the algorithm to compare mutual offers from quantifying any similar product titles and explore a similarity score between 1 to 100. The highest score which indicates higher similarity of both product titles are.

For matching, we will extract the targeted column “title” as input, implement the Ngrams, TF-IDF techniques and vectorize all text into wording encoders, based on the cosine distance we can then compute the final similarity score on all matched items. [6]

Ngrams

It is a commonly used text mining technique which can effectively use in finding any co-occurring text which appeared on a sentence, in this project we have set the N=3, which means every 3 words will bring a bag of word.

TF-IDF

The complete name is term frequency inverse document frequency, which is commonly used in any information retrieval task and mining process. It is suitable for evaluating the importance of specific given word to the entire documentation, as well as recalls the word occurrence counts.

Embedding Algorithm

This is an algorithm which vectorize all text into different dimensional encoders, generally used in any sort of clustering, language classification tasks. It is a popular framework for any sentence embedding purpose.

Vectorization

It is an algorithm under the Scikit Learn package, transform split word (post Ngrams) into a workable token, data will then vectorize into numerical features and allow computer to perform any complex models on all type of data (text, image and audio).

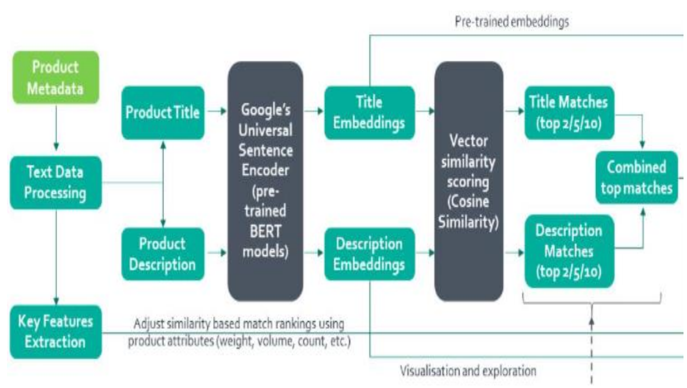

Figure 7. Product Matching Process

5. RESULTS

There are two different results are exported throughout this challenge.

Method 1.

there are matched items founded based on “order_id”, there are some specific items are having more than one product matches, as well as the retailer’s name is also displayed on the third column.

1	c065bcd3-0669-4061-b1ce-fd6467f81b91	c065bcd3-0669-4061-b1ce-fd6467f81b91 bfc5dcf7-00d5-47d8-b5d9-9079bdfa09bd 43b347fb-7ce4-4cb2-8138-ec51564a10d5 a55bb1e0-6171-4269-bbc0-e0da85f582b0 b0a7ed62-ff5a-4529-9248-1ece68a918a3 d3998af7-b484-4254-b984-8b6f1b8a3ad3 d633fb62-fee7-4457-ac1d-febddca5e31f 4eb858e0-97d7-44ba-a6b0-47bd94367b2f 9f9a1ff0-a240-4cbc-94de-6d697c1d3076 25a1d01b-4c70-4dfe-bc72-288c0dbc4fb8 bca91096-cd8c-4b88-856b-af8f7b30f51c d3953b22-1849-4c2d-90a8-531481290189 706c177b-198c-4f4f-821d-a6ca5847b3d0 c02c3b3c-804f-4718-8138-6df36fa2780c 89ac4d87-b3a0-4da8-ab7d-dab13600c02e ef42dd96-120e-4fa0-b03e-b2cae7d00420 81b40b75-fbe1-47be-8d92-24fb16cd8d54 9f411113-a7f5-4b4c-b00f-565c3b57be84 d9c90546-5bd7-45dd-9f59-f006ae3cee33 604f5758-d15f-44e2-9db1-e1c45c43c7e0 4ca6dc46-0136-452c-b77a-d5e9aba6e557 1c0a989a-c3f1-4177-a5f9-723b155f42ee 435bd784-b0df-45d3-93be-c3f3aa49ad4f 9644679a-7799-43f9-b6c6-9da2ab8cc620 cd0e3f8a-c92c-4d68-8cfc-5597876b5cd3 fce40dc7-6b2f-42d3-b0ed-398f857130a9 031d8372-25ad-4043-9dd9-70efd6daa806 955c2ec3-2a5a-4ed0-a393-67079d1c74fb a2962af3-7087-431f-ab39-32fcb2ed544a 172e4903-80fb-449b-a128-b089706a2c3a 31e34ec7-3bb6-46dc-9802-40d0a71a3fd9 4a835144-5e30-4c39-9c4b-200fe6b1b4f6 c50b0484-4137-488a-827c-f70ac3285218 7d39f395-1b7e-4f20-a79d-a3c131b4681a 41dd9546-d5b3-44ad-b102-5ac4eda7695f 15014aba-c843-4964-8be1-5292f9351ddf 7631e4dc-e913-47d1-970e-f019d02f9b23 4038b762-33f2-44dc-af27-577cbfbe418c 856601ec-269b-4e1a-8424-52879ff64384 a982a5d2-ab0d-49a9-a0a5-8358f3b3e5cd 2a0be397-eb5b-45aa-ac0f-54eb48e61856 f62c4b9b-e77e-4a30-b5f6-4a4a9c7f9fd0	aboutyou
4	649b7eeb-30be-4bee-9b45-b01f378ba11a	649b7eeb-30be-4bee-9b45-b01f378ba11a 7cf774f5-fd83-4e1c-a555-8cffa2037a53 d3daa54a-c2d9-4285-a5c1-86ea0ff73993	zalando

Figure 8. “Order_id” product matching

Method 2.

In the method 2, apart from calculating the cosine distance, we have also implemented the threshold score to 0.80, which means any similarity below 80% will not display on the chart.

```
[ ] import time
    t1 = time.time()
    matches = awesome_cossim_top(tf_idf_matrix, tf_idf_matrix.transpose(), 10, 0.8)
    t = time.time()-t1
    print("SELFTIMED:", t)

    SELFTIMED: 14.779923677444458
```

Figure 9. Threshold set up

The matched titles are displayed along with similarity score on the side, final outputs are displayed in ascending order.

	left_side	right_side	similairity
23386	Concealer 'Can't Stop Won't Stop'	Concealer Can't Stop Won't Stop	0.991327
671	LOUISA High Heel Sandalette Keilsandalette Plateausandalette	PRUE WEDGE Keilsandalette Plateausandalette High Heel Sandalette	0.988736
17402	KATYAA Keilsandalette Plateausandalette High Heel Sandalette	LOUISA High Heel Sandalette Keilsandalette Plateausandalette	0.988736
17404	KATYAA Keilsandalette Plateausandalette High Heel Sandalette	High Heel Sandalette Keilsandalette Plateausandalette	0.988736
17403	KATYAA Keilsandalette Plateausandalette High Heel Sandalette	High Heel Sandalette Keilsandalette Plateausandalette	0.988736
17405	KATYAA Keilsandalette Plateausandalette High Heel Sandalette	LOUISA High Heel Sandalette Keilsandalette Plateausandalette	0.988736
17406	KATYAA Keilsandalette Plateausandalette High Heel Sandalette	LOUISA High Heel Sandalette Keilsandalette Plateausandalette	0.988736
85198	Sportunterwäsche	Sportunterwäsche 'IVESDALE'	0.983543
61785	Schlafshirt 'EMELIE'	Schlafshirt	0.982200
61789	Schlafshirt 'EMELIE'	Schlafshirt	0.982200

Figure 10. Product title matching

6. DISCUSSION

Product matching indeed is a challenging and complex task.

Key barriers:

- **Limited computational processing capacity**: There were multiples computational crashed incidents in Google Collaboratory due to low GPU and RAM capacity
- **Cell continuously executing**: Previously was tried to implement KNN model in identify potential matching items but cell was keep running and did not execute until end of run time.
- **Model limitation**: Due to the metadata size volume, there are limitation on certain classification models selection.
- **Not real time model**: Product matching is an effective business solution in matching offer matched items for product recommendation purpose, but the model action on real time data and there are underlying lagging indicators.
- **Inconsistent product information**: Many selling items title are manually inserted by manpower and each retailer might insert slightly different to another.

Key advantages:

- **Optimize product insight knowledge and improve overall competitive power**: Companies will get to know platform item offer price insights, easily to conduct benchmark with other market competitors.
- **Offer rational price policy**: Able to understand what the average market price is for per item, help in segment and create a rational price policy.
- **Product recommendation**: According to platform dynamic prices changes to adjust any promotions efficiently.
- **Minimize and control overstocking condition**: By understanding what the on-trend products are, companies can smartly slow down the procurement order on low sale revenue product to avoid overstocking situation.
- **Monitor each product lifecycle**: By accessing to product data, organization can accurately understand the product performance [7].

7. CONCLUSION

The contrastive learning model has spread and has seen increasing benefits to information retrieval tasks in recent years.

Product matching algorithms can bring many positive outcomes to be diverse business retailers. By reviewing the export matrix of item similarity levels, e-commerce organizations can quickly understand and identify their market strength and opponent position.

There are number of barriers throughout this challenge as due to computational capacity, it is indeed an effective and powerful algorithm, but requires an efficient computer power and resources to compute the output effectively.

8. REFERENCE

[1] A. Danielkievich, "Forbytes," 14 December 2021. [Online]. Available: https://forbytes.com/blog/product-matching-in-ecommerce/. [Accessed 26 03 2022].

[2] A. Turing, "Computing machinery and intelligence," *Oxford academic* , no. 01 October 1950, pp. 433-460, 1950.

[3] D. Cole, "The Chinese Room Argument," *Standord Encyclopedia of Pholosophy* , no. Thu Feb 20, 2020, 2020.

[4] J. F. a. J. Hale, "Corpus Methods," Department of Linguistics, Unviersity of Georgia , [Online]. Available: https://linguistics.uga.edu/research/content/corpus-methods. [Accessed 28 03 2022].

[5] N. D. L. H.-Y. S. Ming Zhou, "Progress in Neural NLP: Modeling, Learning, and Reasoning," ScienceDirect, 14 12 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2095809919304928. [Accessed 30 03 2022].

[6] A. Danielkievich, "Product Matching in Ecommerce: How to Use Deep Learning for Making Informed Offers," Forbytes , 14 12 2021. [Online]. Available: https://forbytes.com/blog/product-matching-in-ecommerce/. [Accessed 29 03 2022].

[7] "Product Matching AI: The Secret Weapon Powering Successful eCommerce Teams," Intelligence Node, [Online]. Available: https://www.intelligencenode.com/blog/product-matching-ai-for-ecommerce/. [Accessed 30 03 2022].