Extracting and Grouping Furniture Products Using Transformer Models

George-Vlad Manolache

1. Problem Statement

The goal of this project is to **extract furniture product names** from text content on e-commerce websites and **group URLs based on product similarity**. This involves building a pipeline that:

1. **Extracts product names** accurately from noisy website content using **Named Entity Recognition (NER) models**.
2. **Clusters similar products** by computing **sentence embeddings** and identifying similarities across different URLs.

The project aims to demonstrate how **transformer-based models** such as **BERT** and **RoBERTa** can be fine-tuned for Named Entity Recognition tasks, and how **Sentence Transformers** like **all-mpnet-base-v2** or **all-MiniLM-L6-v2** can be used to **group similar products** by averaging embeddings of the furniture products found and computing similarity scores.

2. Dataset presentation and Analysis

The proposed dataset contains a file with **704 URLs** from different furniture stores, however, many of them either do not feature any products or are non-functional, presenting a significant challenge in identifying sufficient product entities in order to train the model.

To address this, a thorough search for **550 valid URLs** was conducted for achieving a good **F1 score**, with **manual annotation** performed to identify furniture product entities across each URL. The annotation was carried out using **RELATE**, a Romanian language technology platform designed for processing Romanian text and not only. Of the **550 URLs examined**, approximately **186 were deemed useful**, yielding a total of **3.367 identified furniture products**, while the others were either inactive or empty of products.

Before proceeding with the annotation, it was essential to **extract content from the web pages**. For this task, **Selenium** and **BeautifulSoup** were employed to gather and parse dynamic data from each URL. This was followed by a preprocessing phase to **eliminate unnecessary spaces** and **separate concatenated words**.

Figure 1 illustrates the correlation between the percentage of the initial **550 URLs processed** and the overall number of products identified.

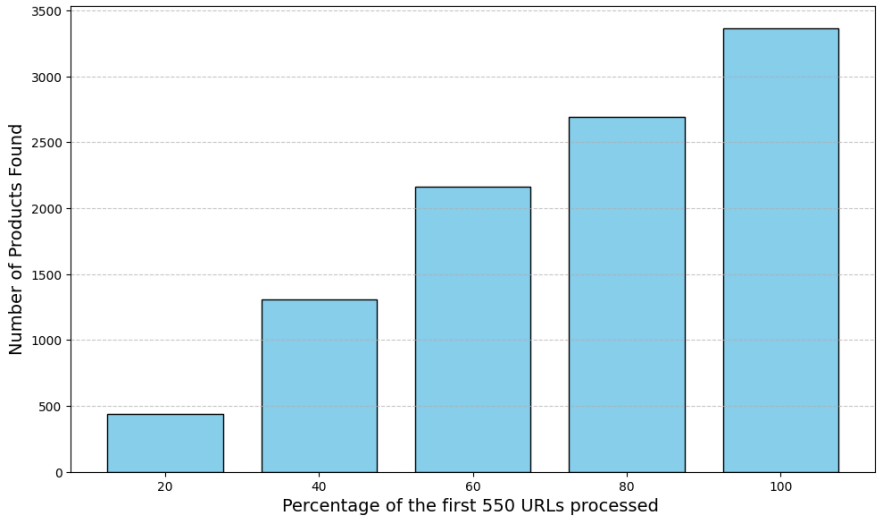


Figure 1

Figure 2 displays a snippet of the processed content from a URL, accompanied by some annotated products. Most products tend to be listed individually on separate lines, frequently near key phrases such as **'You may also like,' 'Product details,' 'Add to cart,'** and **pricing information**, which facilitates easier identification by the model. However, some products are found within more complex contexts, such as **customer reviews** or **product descriptions**, as shown in **Figure 3,** or simply are found in the **first line** of the URL content

/URLs/url_content_125.txt

```
181 Compare
    ORG
182 Tommy Barstool
183 Regular price
184 $ 430.95
185 $ 507.00
186 Sale price
187 Add to Wishlist
188 Add to Cart
189 Quickshop
        ORG
190 The Sally outdoor collection is characterised by a frame with wood look...
191 Compare
        ORG
192 Sally Outdoor Collection
193 Regular price
194 $ 3,057.45
195 $ 3,597.00
196 Sale price
197 Add to Wishlist
198 Add to Cart
199 Quickshop
    ORG
200 Stone Armchair is defined by the curved shape of its back. An...
201 Compare
    ORG
202 Stone Accent Chair
203 Regular price
204 $ 1,522.35
205 $ 1,791.00
206 Sale price
```

Figure 2

Figure 3

Of the **186 content files extracted from URLS**, **80%** were selected for training and **20%** for validation, resulting in **2692 product entities for training** and **675 for validation**.

To ensure the data is properly formatted for the **NER model**, a series of preprocessing steps were applied. First, each word in the files was tagged using the **BIO (Beginning-Inside-Outside)** tagging scheme. This means the **first word of a product entity** was labeled with the **B-PROD** tag, and subsequent words, if the product name consisted of multiple words, received the **I-PROD** tag. Words that were not part of any product entity were assigned the **O tag**.

Since treating individual lines as independent data points could limit the context available to the model, an **ENDLINE** tag was added at the end of every line and lines were **concatenated up to 63 words** to create more context. Lines exceeding 63 words were **split into sublines**, ensuring that multi-word product entities remained within the same subline. This decision was made to provide the model with **neighboring context**, making it easier to recognize products when surrounded by relevant keywords. At the same time, it prevented sentences from becoming too long, which would risk **token truncation**. Since the model has a **512-token limit** for each input (due to subtokenization, where 63 words can result in over 250 subtokens), this approach ensured no information was lost by exceeding the model's token limit and maintained an **acceptable number of data points** for both the training and validation sets. This resulted in a total of **3.860 data points for training** and **836 for validation**.

The final preprocessing step involved using the **specific tokenizer** of the model to tokenize the dataset for training and validation.

3. Model Architectures and Hyperparameters

To perform the **Named Entity Recognition (NER) task**, two **transformer-based models** were employed: **DistilBERT** and **DistilRoBERTa**. These are lightweight, distilled versions of **BERT** and **RoBERTa**, retaining most of the original models' performance while being more efficient.

BERT (Bidirectional Encoder Representations from Transformers) is based on stacked transformer encoder-only layers architecture (see **Figure 4**) pretrained on large corpora using two key objectives:

- **Masked Language Modeling (MLM)**: a portion of words (about 15%) is masked and the model is trained to predict the original values of these masked words
- **Next Sentence Prediction (NSP)**: the model is trained to predict whether the two sentences follow the correct order or not.

RoBERTa (Robustly Optimized BERT Approach) builds upon BERT by making several key improvements:

- **Removing the Next Sentence Prediction (NSP) objective**
- **Dynamically changing the masking pattern**

- **Training on a larger corpus with more data.**

Once pretrained, the models are **fine tuned** for different **NLP tasks**, in this scenario, for **Named Entity Recognition.** (**Figure 5**)

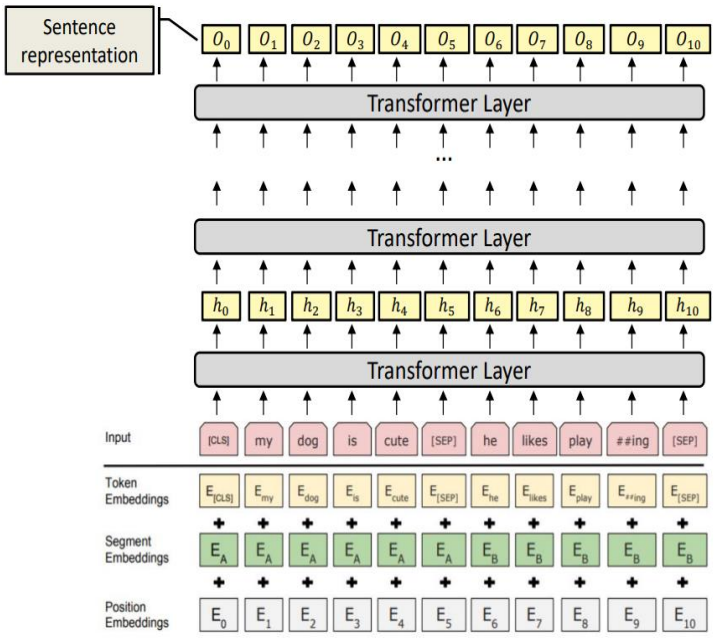


Figure 4

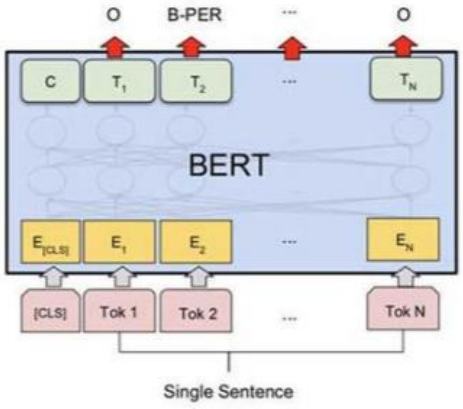


Figure 5

The **hyperparameters** used for training the models are the following:

- **Learning Rate**: 1e-4
- **Batch Size**: 16 (for both training and evaluation)
- **Optimizer**: AdamW (default in **HuggingFace**)
- **Learning Rate Scheduler**: Cosine with Restarts
- **Warmup Steps**: 160 (**approx. 10-11%** of the training set size)
- **Weight Decay**: 1e-5
- **Number of Epochs**: 7

Both models were **fine-tuned** using the **AutoModelForTokenClassification** class from the **HuggingFace** library, which simplifies model loading for token classification tasks. The models were trained to optimize the **evaluation loss** as the primary metric, ensuring that the best-performing model with the lowest evaluation loss is saved and used in the final evaluation.

4. Quantitative and Qualitative Analysis

To **quantitatively** evaluate the performance of **DistilBERT** and **DistilRoBERTa** metrics like **Precision**, **Recall** and **F1 Score** are employed to assess how well each model extracts product entities. Results can be seen below in the following graphs:

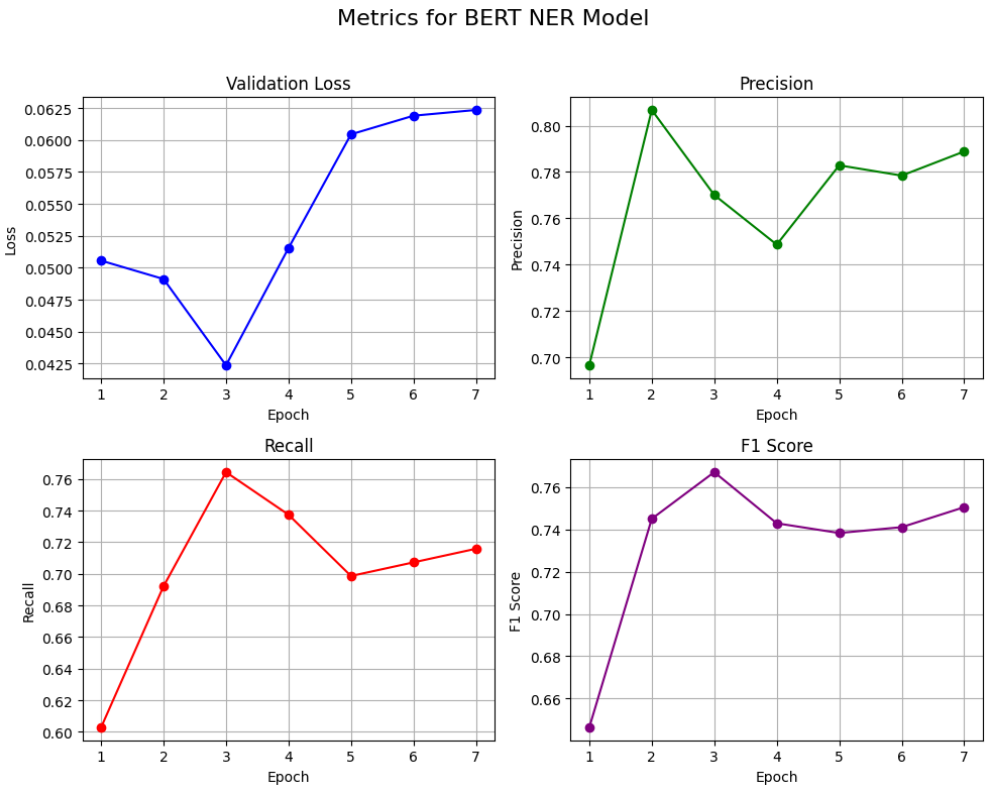
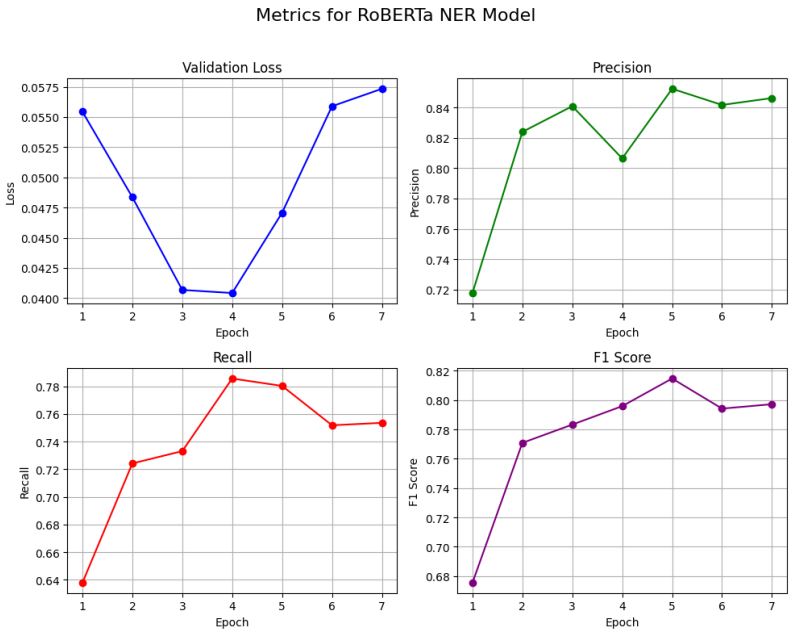


Figure 6

DistilBert achieved at its best **76,71% F1 Score** with **77% Precision** and **76,42% Recall** at epoch 3.



DistilRoBERTa achieved its highest **F1 Score** of **81.46%** at **epoch 5**. However, the model version from **epoch 4** was chosen instead, as it had the lowest **validation loss**, helping to mitigate the risk of **overfitting**. The **F1 Score** at **epoch 4** was **79.58%**, with a **Recall** of **78.55%** and a **Precision** of **80.63%**.

Figure 7

Model	F1 Score	Precision	Recall
DistilBERT	**76,71%**	**77%**	**76,42%**
DistilRoBERTa	**79,58%**	**80.63%**	**78,55%**

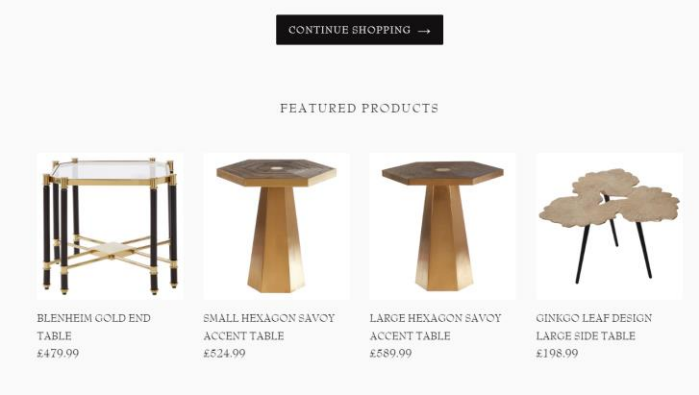
These results indicate that **DistilRoBERTa** offers slightly better performance than **DistilBERT** achieving higher scores for both Precision and Recall, likely due to the model's enhanced pretraining strategies and robust language understanding.

Despite the high imbalance between **O tags** and product related tags **B-PROD** and **I-PROD**, both models maintained high precision and recall, demonstrating their ability to handle class imbalance effectively.

Taking the above results into consideration, **DistilRoberta** was selected to perform the final predictions

The **qualitative analysis** complements the quantitative results by examining specific prediction patterns, model errors, and behavior in diverse contexts. After analyzing some cases of the **DistilRoBERTa**' predictions, the following conclusions were made:

- The model performs very well in indentifing products listed individually on separate line followed by certain keywords and price information:



Content fragment: " ...Continue shopping \n Featured Products \n Featured Products \n **BLENHEIM GOLD END TABLE** \n **BLENHEIM GOLD END TABLE** \n Regular price \n £479.99 \n Sale price \n £479.99 \n Sale..."

Figure 8

Model prediction: [' BLENHEIM GOLD END TABLE', ' SMALL HEXAGON SAVOY ACCENT TABLE', ' LARGE HEXAGON SAVOY ACCENT TABLE', ' GINKGO LEAF DESIGN LARGE SIDE TABLE']

url: https://uniquehomefurnishing.co.uk/products/hzh330

- Most of the time the model can also capture the product from customer reviews or descriptions:

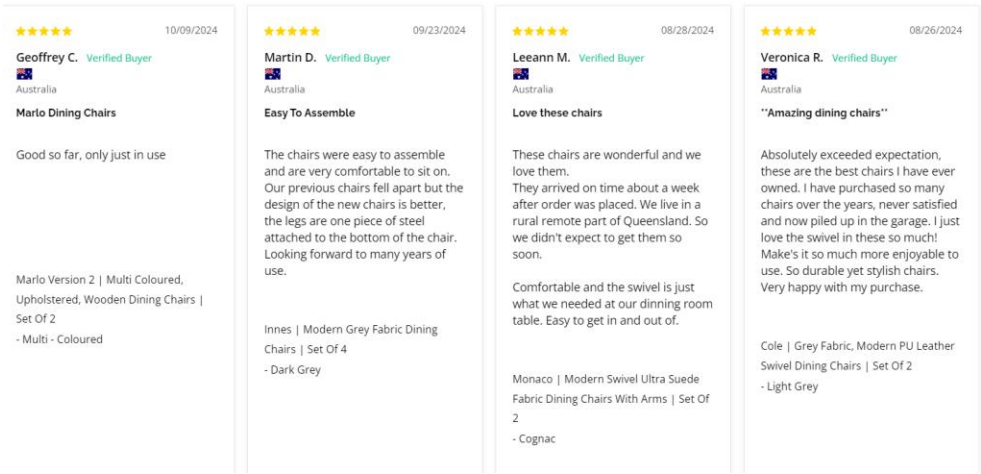


Figure 9

Content fragment: "... 08/28/2024 LM Leeann M. Australia Love these chairs These chairs are wonderful and we love them.\n

They arrived on time about a week after order was placed. We live in a rural remote part of Queensland. So we didn't expect to get them so soon. \n
Comfortable and the swivel is just what we needed at our dinning room table. Easy to get in and out of. **Monaco | Modern Swivel Ultra Suede Fabric Dining Chairs With Arms | Set Of 2 Cognac** Was this helpful? 0 0 ..."

Model prediction : [..., ' Marlo Version 2 | Multi Coloured, Upholstered, Wooden Dining Chairs | Set Of 2 Multi - Coloured', ' Innes | Modern Grey Fabric Dining Chairs | Set Of 4 Dark Grey', ' Monaco | Modern Swivel Ultra Suede Fabric Dining Chairs With Arms | Set Of 2 Cognac', ' Cole | Grey Fabric, Modern PU Leather Swivel Dining Chairs | Set Of 2 Light Grey', ...]

url: https://www.onlydiningchairs.com.au/products/wooden-velvet-retro-grey-dining-chair

Content fragment:"... **Sculptural Ebonized Credenza with Silver Leafed Front** \n Regular price \n $9,800,00 .... This **nine drawer credenza** has had a total groun up restoration with the case begin... Model prediciton: [' Sculptural Ebonized Credenza with Silver Leafed Front', ' nine drawer credenza',...]

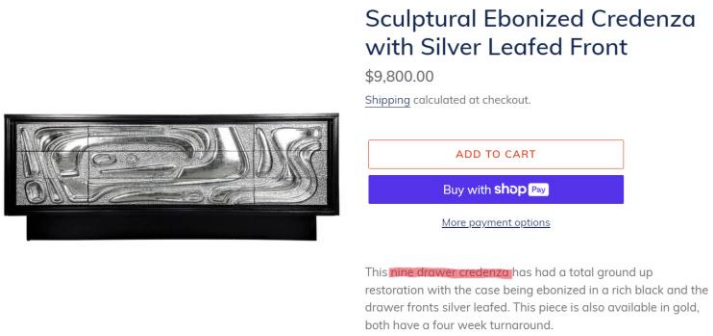


Figure 10

url: https://midcenturymasters.com/products/sculptural-ebonized-credenza-with-silver-leafed-front

- There are situations where the model isn't able to capture the product, due to the noise caused html parser used to extract the content:
  Content fragment: "... the elegantly industrial design represents both a new material and a new designer for GUBI.BAGDAD PORTABLE LAMPOriginally designed in 1954, the **Bagdad Lamp** is a brilliantly offbeat showcase of Mathieu Matégot's material..."

  Here, the model successfully identifies "**Bagdad Lamp",** but misses "**BAGDAD PORTABLE LAMP"**, even if **DistilRoBERTa** use subtokenization.

  url: https://shop.gubi.com/products/epic-dining-table

- Sometimes the model tags only **subtokens** instead of the full token (e.g., tagging "**im Hammock**" instead of "**Denim Hammock**"), tags words completely out of context (e.g., "**SATISFIED HOME OWNERS**", "**200**", "**"**"), or treats parts of the same product as separate entities (e.g., tagging "**Hanging**" and "**Chair**" as two products instead of "**Hanging Chair**").

  A possible solution when the model tags only the final subtokens of a full token is to apply the **B-PROD/I-PROD** tag to all subtokens of the corresponding token. However, this approach can lead to issues if the token is not part of a product entity, potentially increasing **false positives**. A way to address misclassification is by re-labeling tokens in the following manner: If a token labeled as **O** is between a **B-PROD/I-PROD** and an **I-PROD**, it is very likely that it should also be labeled as **I-PROD**. Another way is to filter out the elements that are just numbers or are solely lowercase words.

In general, **increasing the number of training examples** or exploring **augmentation techniques** (e.g., using synthetic product examples) and using a more robust **HTML parser** could help the model to mitigate these issues.

5. Grouping URLs based on similar products

To group URLs based on similar products, two **Sentence Transformer** models were utilized: **all-MiniLM-L6-v2** and **all-mpnet-base-v2**.

A **Sentence Transformer** generates dense vector representations (embeddings) of sentences or texts, capturing their semantic meaning. These embeddings can then be used for tasks such as clustering or computing similarity between texts. Such models are particularly effective in converting sentences into fixed-size vectors that can be compared using similarity metrics like **cosine similarity**.

all-MiniLM-L6-v2 model is compact and efficient, designed for fast computation, offering a good trade-off between speed and accuracy.

all-mpnet-base-v2 is a more advanced model with deeper contextual understanding, providing higher-quality embeddings, ideal for tasks that require precise semantic representations.

all-mnpnet-base-v2 was chosen to perform the task. For each URL, embeddings were computed for all the products listed, and the **mean embedding** was taken to represent the overall product content of the URL. A **similarity matrix** was then constructed using **cosine similarity** to measure the relationship between the embeddings of different URLs. The **top 3 most similar URLs** were selected for each URL, effectively grouping them based on product similarity.

The method seems to provide good results. For example, this website has as products multiple types of tables, https://livingedge.com.au/products/tables/dining, and the recommendations made are the following:

- https://rutherford-romaguera2611.myshopify.com/products/teddy-rug%20via%20@//twitter.com/undergrndmedia: "[' Dunn Dining Table', ' Rume Dining Table', ' Sierra Table', ' Wyn Waterfall Side Table', ' Alexei Outdoor Daybed', ' Cement Nesting Tables', ' Wyn Waterfall Coffee Table']"
- www.coolstuffandaccessories.com/products/white-wood-dining-table: "[' White Wood Dining Table', ' white wood dining table', ' Metal Dining Table/Grey', ' Modern Rectangular Dining Table', ' modern rectangular dining table', ' Rectangular Dining Table 32""x48""/White', ' Round Dining Table 48""/Black', ' Sia Wood Dining Table', ' Sia dining table.']"
- www.arighibianchi.co.uk/collections/new-in-homepage/products/birds-coat-rack: "[' Roslyn Dining Table', ' Lily Dressing Table', ' Hulk Dining Table', ' Indiana Dining Table', ' Indiana Dining Table.', ' Indiana...', ' Kingston Bedstead High Footboard', ' Sofas Side board Mirror Rugs Dining table']"

6. Conclusion

This project demonstrates the potential of transformer models for NER and product grouping tasks in real-world e-commerce applications. **DistilRoBERTa** emerged as the superior NER model, while the **all-mpnet-base-v2 Sentence Transformer** effectively grouped URLs by product similarity. Despite challenges related to noisy input data and class imbalance, the models provided reliable performance. The methods and results in this report highlight the feasibility of using transformer-based NLP solutions to streamline product extraction and clustering in online retail environments. Future work focusing on **data collection and improved parsing techniques** would further enhance the system's performance and robustness.

This pipeline can serve as a foundation for building **recommendation systems** or **search tools** that suggest related products or group similar items, ultimately enhancing the customer experience in online retail.