ECONOMICS AND CULTURE 20(1), 2023

DOI: 10.2478/jec-2023-0007

ITEM MATCHING MODEL IN E-COMMERCE: HOW USERS BENEFIT

Olga Cherednichenko [1], Oksana Ivashchenko [2], Ľuboš Cibák [3], Marcel Lincenyi [4]

[1]Bratislava University of Economics and Management, Bratislava, Slovakia, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine, olga.cherednichenko@khpi.edu.ua, ORCID: orcid.org/0000-0002-9391-5220

[2]Bratislava University of Economics and Management, Bratislava, Slovakia, Oksana.Ivashchenko@vsemba.sk, ORCID: orcid.org/0000-0003-3636-3914

[3]Bratislava University of Economics and Management, Bratislava, Slovakia, lubos.cibak@vsemba.sk, ORCID: orcid.org/0000-0003-3881-7924

[4]Bratislava University of Economics and Management, Bratislava, Slovakia, marcel.lincenyi@vsemba.sk, ORCID: orcid.org/0000-0002-9076-026X

Received 4 April 2023; accepted 29 May 2023

Abstract

Research purpose. During the last decades, e-commerce sales have been rocketing, and this tendency is expected to increase over the following years. Due to the digital nature of e-commerce, one actual item can be sold on various e-commerce platforms, which leads to the exponential growth of the number of propositions. At the same time, the title and description of this item might differ. All these facts make more complicated for customers the process of searching on online platforms and change business approaches to the development of competitive strategy by e-commerce companies. The research question is how we can apply a machine learning algorithm to detect, based on the product information such as title and description, whether the items are actually relevant to the same product.

Methodology. We suggest an approach that is based on a flexible textual data pipeline and the usage of a machine-learning model ensemble. Each step of the data processing is adjustable in dependence on domain issues and data features because we can achieve better results in solving the item-matching task.

Findings. The item-matching model is developed. The proposed model is based on the semantic closeness of text descriptions of items and the usage of the core of keywords to present the reference item.

Practical implications. We suggest an approach to improving the item searching process on different e-commerce platforms by dividing the process into two steps. The first step is searching for the related items among the set of reference items according to user preferences. The reference item description is created based on our item-matching model. The second step is surfing proposals of similar items on chosen e-commerce platforms. This approach can benefit buyers and sellers in various aspects, such as a low-price guarantee, a flexible strategy of similar products shown, and appropriate category-choosing recommendations.

Keywords: E-commerce; Item matching; Model; Entity resolution; Business Management; Marketing; Digitisation.

JEL codes: M30; M19; O31

Introduction

E-commerce has been an essential part of our lives in recent years. According to the Trend report "eCommerce in Europe" (eCommerce in Europe, 2023) value of this market in Europe is expected to grow by 12% per year between 2020 and 2026. A lot of e-commerce shops work in this market, and they suggest a wide range of goods of different categories and brands, such as foods, electronics, shoes, clothing, etc., produced by different manufacturers. At the same time, the same real-world products are sold by different e-shops. These tendencies impact both customers and sellers.

© 2023 Authors. This is an open-access article licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0)

On the one hand, customers spend a lot of time searching for necessary items because they have to compare items from different online platforms, their descriptions, and photos; also, they have to solve the issue if these items are the same or only similar, and only after that customers are able to make a decision about buying the item which they were looking for. On the other hand, sellers spend a lot of time analysing the e-commerce market to assess the demand and supply of goods, and the price policy of their competitors in an attempt; as a result, to be competitive and to save their market share. So, the task of developing an item-matching model is relevant to nowadays requirements of sellers and customers of e-commerce platforms.

The product offers on e-commerce platforms usually include titles, text descriptions and images. To identify specific product items, attributes should be extracted for further processing. One of the main features of product descriptions on different online platforms is their heterogeneity with different levels of detail. However, some retailers have started to provide product offers with structured or semi-structured specifications and semantic annotations (i.e., Microdata markup). On the one hand, it makes the process of item-matching easier, but on the other hand, new challenges of entity resolution are rising. As one of the product characteristics, image processing can also be involved in the item-matching process.

As has been stated above, the problem of identifying similar products is vital for all participants in the e-commerce market. The main aim of our research is to develop a method which makes the search process on e-commerce platforms less complicated for both customers and sellers. To answer the research question, we propose a flexible textual data pipeline, which counts the speciality of domain and data, and a set of machine learning models to provide data processing. As a result, we developed an item-matching model based on the semantic connection of item text descriptions and running the tag core to represent reference items.

Literature Review

Product matching is a specific part of the Entity Resolution problem. The main goal of product matching is finding the most suitable item according to preference records. This issue has been explored for more than 30 years. Usually, researchers focus on the type of task (classification, clustering, etc.), the similarity metrics and the data features (unstructured, annotated, etc.).

In Köpcke et al. (2010), for similarity calculation, the authors used different non-learning approaches (COSY, PPJoin+) and learning-based match approaches (support vector machine, decision tree) for bibliographic and e-commerce data entities. Obtained results showed low match quality for the e-commerce domain and proved the necessity of further improvements by using other similarity measures or changing characteristics of the learning-based approach.

Product titles identify the product, but on different online platforms, the title of the same product can vary, and, as a consequence of that, traditional similarity measures may not show satisfactory results for matching item titles. Gopalakrishnan et al. (2012) proposed an unsupervised approach for solving item-matching problems based on the similarity of item titles. They proposed to enrich item titles with missing words which frequently appear in search engine results and calculate token weights based on their ability to enrich titles from search results; for matching-title pairs, authors proposed to use the Cosine similarity. In their experiment, they explored titles of cameras and appliances and such missing tokens as brand and model number; the following works planned to explore titles of products which may not consist of such representative tokens. Akritidis et al. (2018) also proposed clustering item titles to solve the item-matching problem; however, their approach did not depend on external search engines and was based on morphological analysis of items titles and consisted of two phases: construction of k-combinations of words and providing same statistics; usage forward index and statistics to define representative cluster tag of the item. Obtained results approve the necessity of solving the problem of incorrect division of identical items in different subsets.

Some researchers considered the item-matching problem a classification issue (Kannan et al., 2011, Köpcke et al., 2012). Kannan et al. (2011) used an approach based on exploring the semantics of item descriptions and finding the probability of items. To provide this approach, the authors used two datasets with structured and unstructured data, with the main goal of matching records from the unstructured

dataset with one or several structured records. The developed system is able to use any domain with offers and a storehouse of structured data. In contrast to the approach (Kannan et al., 2011), some researchers proposed a matching approach based on pattern extraction of Universal Product Code to identify similar items (Köpcke et al., 2012). Results showed low quality, so the authors planned to investigate the usage of other methods to improve the suggested approach.

Some online platforms use specific product identifiers (GTINs, EANs). The work suggested using schema.org annotations to cluster items from different sellers related to the same item and learn matcher using this clustering as a distant control (Peeters et al., 2020).

Four deep learning (DL) solutions (SIF, RNN, Attention, and Hybrid) are provided for matching text (Mudgal et al., 2018), the so-called framework Deepmatcher. The authors focused on the fact that DL could better solve this task instead of classical learning-based entity matching due to its ability to obtain good results on raw data. Four DL models and the Magellan EM system (decision tree, random forest, Naive Bayes, SVM and logistic regression) were applied to compare obtained results. They made experiments on 32 datasets of different sizes and from different domains: 11 of them are structured datasets, 6 datasets are textual and 6 dirty datasets. Experiments showed several challenges: DL models need a large quantity of training data to obtain good results, and limited capability of DL models to understand semantics from different domains. Primpeli et al. (2019) planned to repeat the results obtained by Mudgal et al. (2018). For the experiment, the researchers created a training dataset from schema.org annotated item descriptions; the total amount of records was 26 million, and 16 million clusters were created touching on the same items. Upon evaluating matching results, the gold standard dataset was created. The authors concluded that their results proved the results obtained by the previous researchers (Mudgal et al., 2018) about the advantage of using DL methods on textual data.

Other researchers also suggest using deep learning models, e.g., the task of product matching was solved (Peeters et al., 2020) by applying the BERT model, and obtained results proved higher efficiency in comparison with the Deepmatcher results (Mudgal et al., 2018).

Pre-processing is an important part of the successful solution of the product-matching task. The most challenging step is vectorising of data representation because all approaches of item-matching require numeric data, but the product description is text and image. Vectorising is used in text pre-processing to make any text machine understandable. Changing textual information in purposeful vectors allows using these word embeddings in further text processing, such as text classification, text clustering, sentiment analysis, etc.

There are some powerful approaches to vectorise the text. Mikolov et al. (2013) set to explore the combination of RNN and NLMs to obtain improved word representations for morphologically complicated words. The main goal of Singh et al. (2019) is to recognise unifiable news articles. So, the authors investigated different vectorisation techniques and obtained document embeddings for further document classification. They noticed that vectorisation methods, such as Word2Vec, TF-IDF, GloVec, SenteceToVec, and Doc2Vec, capture only semantic and syntactic meaning of words which are suitable for the task of clustering and classification but not very useful for more complicated tasks. More context-dependable embeddings can be obtained using ELMo, and BERT. Other researchers (Darú et al., 2022) used TF-IDF technique in pre-processing stage for performing the classification of item descriptions in Portuguese. Other authors (Krzeszewska et al., 2022) estimated if the choice of vectorisation technique influenced results of classification results in large text datasets; they made the conclusion that the accuracy of classification depended on the method of vectorisation. Researchers in the paper by Yang et al. (2022) proposed using the topic model (Yang et al., 2020) and BERT model to calculate the similarity between texts. The dataset used for the experiment consists of 1,02 GB of new text data. The authors compared the results of similarity estimation by using their proposed topic model and traditional TF-IDF on the step of keyword extraction. They proved their hypothesis that the new approach would perform better than the traditional technique because of its reflection of semantic relationships between words.

In our previous works, the issue of product matching was explored. In the paper (Cherednichenko et al., 2018), we collected data about mobile phones from different e-commerce platforms and grouped them by similar features; this experiment showed rather good results because most phone characteristics are

quantitative. However, many other items sold on online platforms often do not have such exact characteristics, so in another attempt of ours (Cherednichenko et al., 2020a), we provided an approach of item-matching based on the customer perception of items and their similarity. The results of our experiment showed a good perspective of combining text description and assessment of images with building an item-matching model. Also, taking into account the heterogeneity of text description of goods and the vast amount of propositions of goods, in our subsequent work (Cherednichenko et al., 2021), the reinforcement learning approach was explored. The developed model makes a search query from the obtained core of tags combined with different keywords.

Further experiments showed the dependency of results on the main platform during the core of tag building. Our further research has proved the necessity of creating some flexible methodology which allows changing setups depending on searching goals, types of data, etc. In Cherednichenko et al. (2022), we proposed constructing a flexible processing data pipeline for product categorisation. The main feature of this developed methodology is its flexibility, so new methods and the order of steps of processing data can be changed to obtain the goal.

Thus, we can summarise that despite the great interest in the product-matching task, there are a lot of open issues. Our previous research found some valuable techniques for item grouping, tagging, pipeline construction, etc. As the state-of-the-art shows, the machine learning models are promising enough to solve product-matching tasks. Therefore, the research question is how we can apply a machine learning algorithm to detect, based on their product information such as title and description, whether the items are relevant to the same product.

Research Methodology

The product-matching task is an essential step in e-commerce platforms, as it helps to streamline the buying and selling process and provides a better user experience for all parties involved. It involves matching products based on their characteristics, such as their name, description, category, and other attributes. This task is commonly performed in e-commerce platforms, where multiple sellers may offer the same product but with slightly different names, descriptions, or prices. Product matching aims at identifying and grouping these similar products, providing a better user experience for buyers and helping sellers to compete on a level playing field.

We suggest a step-by-step approach to identify groups of similar products, create the tag core for each group, and search for relevant items on websites. We can summarise our previous results and suggest a novel method for product matching. Our method involves three key stages: first, we use machine learning algorithms to determine the similarity between items, enabling us to group them into relevant product categories. This step is critical in minimising the number of items that need to be processed, making subsequent steps more efficient.

Next, we create tag cores by extracting the most relevant keywords from the item descriptions. This allows us to capture the essence of each item and facilitate more accurate and efficient searching.

Finally, we use machine-learning models to search for items on e-commerce platforms based on the extracted tag cores. This step enables us to efficiently search for items that match the desired criteria while minimising the number of irrelevant search results.

So, firstly the text descriptions of items should be processed, vectorised and clustered. Based on our experience, we would like to highlight that the outputs strongly depend on input data. That is why designing the only model is unrealisable.

In general, text mining deals with issues of the inference of structured information from collections of unstructured input texts. Approaches that are applied in text mining require task-specific text analysis processes that may consist of several interdependent steps and are realised with text analysis pipelines (Wachsmuth, 2015). One of the main problems is that text analysis pipelines are mainly constructed manually because their design requires expert knowledge. Wachsmuth et al. (2013) suggested creating an automatic pipeline for annotation text, which can be presented:

$$\Pi = \langle A.\pi \rangle ,$$

where A is a setting of algorithms;

π is a schedule of performing each algorithm.

Each algorithm needs specific input data and time for performing, and it ends up with specific output data. Time for performing and some restrictions must be considered in the schedule. In their work (Wachsmuth, 2015), the authors continued their exploration of pipeline development; their goal was to construct an efficient pipeline. In this case, they focus their attention on the optimisation schedule. Some authors provided a data cleansing pipeline which joins several interrelated stages of creation training set for further processing to solve the item-matching issue (Peeters et al., 2020; Primpeli et al., 2019).

We can say that the aim of the first stage is to build a multi-class classifier which can predict the product category and create groups of similar items. The high-level steps are presented in Figure 1.

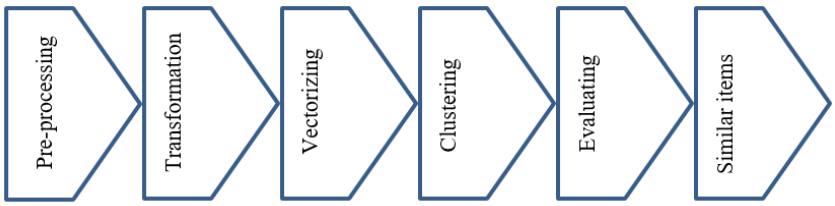


Fig.1. Pipeline of similar items' group creating

As shown in Figure 1, we suggest the following general steps:

1. Data pre-processing (extract text descriptions of items)

2. Data transformation

a. Removing articles (a, an, the), prepositions (in, of, at), conjunctions (and, or, nor), etc. from the item descriptions

b. Stemming

3. Vectorising

4. Clustering

a. Choosing the model

b. Experimenting

5. Evaluating the results of clustering

6. Creating groups of similar items.

The main goal is to make the pipeline as flexible as possible and allow users to change it (e.g., add their own functions, set their own order of action, and use other methods for data processing). As we found out, the clustering of item descriptions is related to the product category, language, data source, etc. The flexible pipeline gives the tool for researching the data and creating groups of similar items.

In one of our previous works, we suggested the creation of a flexible pipeline for solving the item-matching task. A pipeline is a chain of interdependent actions under data in which instructions transform data; the output data of one step is the input data of other steps. Our paradigm of the pipeline is as follows: given a set of unstructured item text descriptions, which are processed to obtain structured information. The main features of our pipeline are its flexibility and automation.

As similar items are created, we can describe each group as a set of keywords. Thus, the second step is to build the tags core for similar items.

Following the results in (Cherednichenko et al., 2020b), we hypothesise that if we know item groups with similar objects, then: 1) similar items have the same core of tags; 2) the group of a new item can be easily found if the core for a group of objects is known.

Having the cleaned tag list sorted by descending tag frequency, we can create the tags core based on the algorithm suggested in our previous work. This algorithm is based on the word2vec model and allows us to consider the semantic similarity of words.

The data pipeline of tag core creation is depicted in Figure 2.

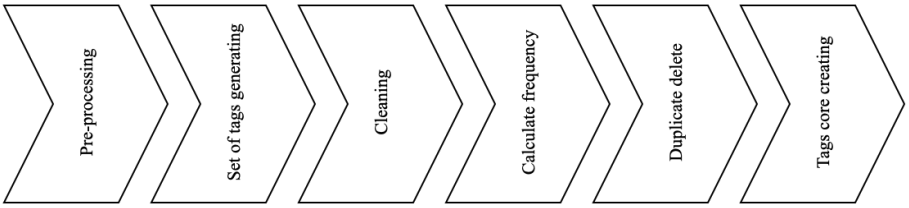


Fig.2. Pipeline of tags core creating

As shown in Figure 2, we suggest the following general steps:

1. Data pre-processing (represent each item as a set of keywords).

2. Generate the set of tags for the group of similar items.

3. Data cleaning (remove punctuation signs, numbers, out-of-vocabulary words, etc.).

4. Calculate the frequency for each tag to order the set of tags.

5. Delete duplicates.

6. Create the core of tags.

Thus, we have the tag core as a description of the group of similar items. It is the basis for searching for new items and categorising them. To build a similar item group, we suggest their text descriptions are similar in terms of semantics and apply some NLP techniques in the data pipeline. Despite the developed data pipeline, its flexibility is an important issue due to text data, as item descriptions are sensitive to each step of their processing.

Finally, the tag core can be implemented in the item-searching algorithm. We research the model using Reinforcement Learning (RL) approach (Cherednichenko, 2021). Reinforcement Learning has become increasingly popular in various domains owing to its successful applications. It is a machine learning type involving agent learning through interactions with an environment.

In our specific task, the agent interacts with an e-commerce platform by sending search queries. The environment responds with a list of items that match the query, ranging from a few to tens of thousands, making their processing quite challenging. The objective of the agent is to minimise the size of the list of items while obtaining a complete and accurate set of items that can be grouped into a single product category based on varying characteristics such as price, size, colour, and shipping, among others. The final-step pipeline is shown in Figure 3.

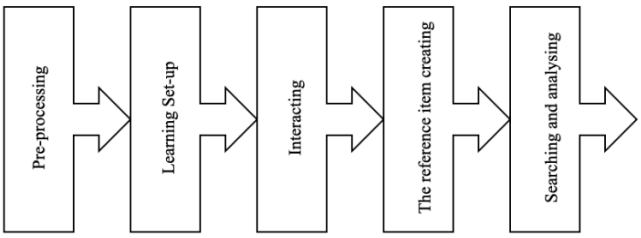


Fig.3. Item Searching

To implement the ideas of RL, we suggest the following steps:

1. Data Pre-processing (cleaning, tags core choosing, etc.).

2. Learning Setup:

a. Define the initial query.

b. Define the reward function.

c. Define the reference tags.

3. Interacting (query and evaluate the response).

4. The reference item creation.

5. Searching and analysing.

Overall, we propose a novel three-stage method that utilises machine learning algorithms for effectively identifying item similarity, creating a tag core, and searching items on e-commerce platforms. Our approach leverages a flexible textual data pipeline and an ensemble of machine learning models, allowing for adaptable data processing that can be tailored to specific domain issues and data features. By leveraging the power of machine learning, we can achieve accurate and efficient item matching, improving the overall user experience on e-commerce platforms.

Research Results

Let us provide the experimental results to show how the suggested approach can be applied and investigate how machine learning is beneficial.

Firstly, we experimented with similar item group creation based on vectorising the text description and clustering algorithm. As mentioned above, the flexible pipeline is a solution for this step. So, we consider the dataset based on items from Men's Shirts category. We collected the set of 381 item descriptions from Amazon.com (Amazon, 2023). After pre-processing text data (empty values drop, lowercasing, punctuation removal, etc.), we receive the set of tokens (pre-processed words) representing the items. Overall estimation of how pre-processing and cleaning work is depicted in Table 1.

Table 1. Overall estimation (Source: authors' contribution)

Number of the step	Step name	Dataset size	Size difference
1	Initial dataset	47 743,00	0,00
2	Non-alphabetic values drop	45 009,00	-2 734,00
3	Stop words drop	42 006,00	-3 003,00
4	Single characters drop	39 397,00	-2 609,00
5	Fully consonants values drop	32 745,00	-6 652,00
6	Fully vowels values drop	32 562,00	-183,00
7	Unique values count	1 299,00	-31 263,00

Vectorising is a crucial step that can significantly influence the outcomes of clustering. We experimented with different vectorisers and implemented the K-means algorithm to group items. Table 2 shows the experimental results with vectorisers. We can notice that the vectorising model defines the outcomes, but the choice of the model should be made only based on a series of experiments. The suggested pipeline allows playing with different models to find the best one.

Table 2. Experimental results with vectorisers (Source: authors' contribution)

Vectorisation model name	Cluster id	Elements number Total
spaCy Tov2Vec model	1	229
	2	152
Custom-trained Word2Vec model	1	354
	2	19
	3	8
Google trained Word2Vec model	1	24
	2	125
	3	83
	4	37
	5	112

To perform the next step of our pipeline, we have chosen Google trained Word2Vec model with five clusters. As an example, descriptions and photos of similar items from cluster 2 due to chosen clustering model are shown in Fig. 4.

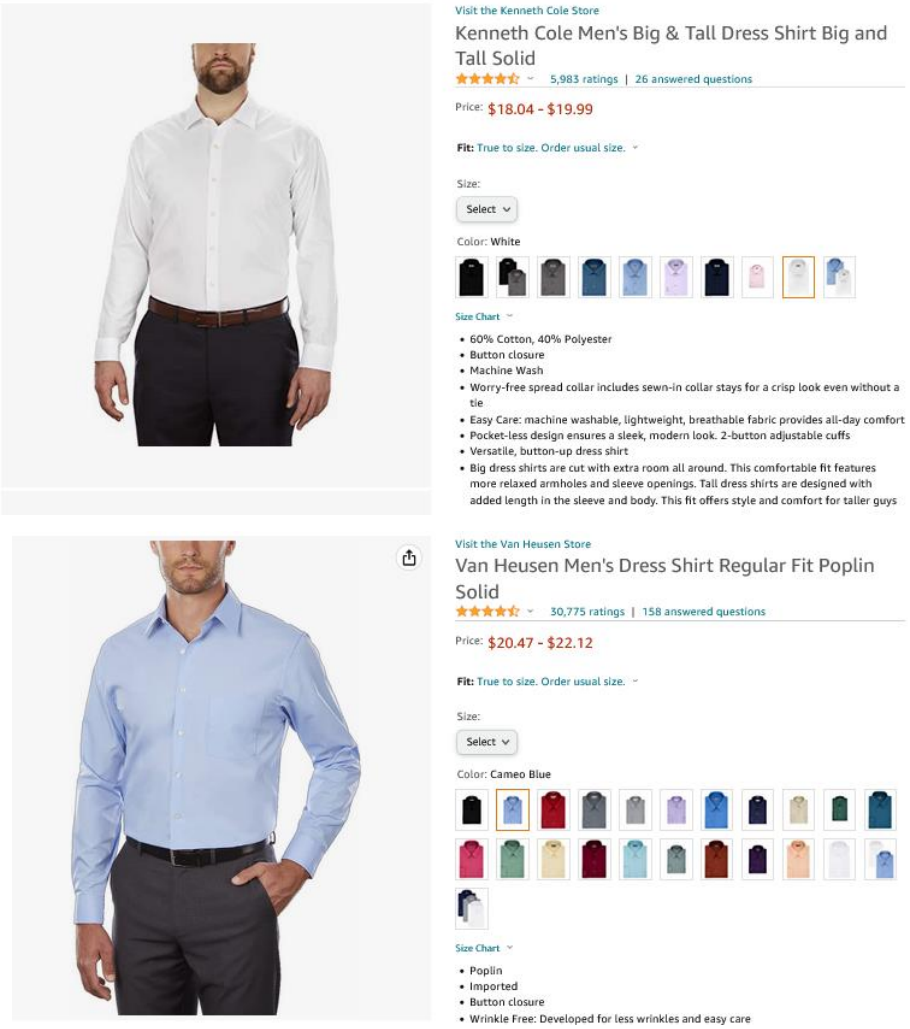


Fig.4. The examples of similar items (Source: extracted from Amazon, 2023)

The next step of the suggested method is the creation of the tag core for the cluster with similar items. In this step, we use the algorithm based on the word2vec model and the frequency of obtained tags in previous steps. The results of performing the second stage are represented in Table 3. The tag core was created for each group of similar items. As a similarity measure, we applied a cosine metric to compare vectors which represent keywords of the item description. We have played with different levels of thresholds, and the outputs obtained with a threshold equal to 0,75 are depicted in Table 3.

Table 3. The results of performing the second stage (Source: authors' contribution)

Clustering method	Cluster Number	The tag core (for similarity value > 75)
Google trained Word2Vec model	1	Snap, fit, shirt, button, flannel, comfort
	2	men, solid, dress, shirt, long, sleeves, button, regular
	3	short, sleeve, shirt, button, polo, casual
	4	western, fit, shirt, button, down, denim,
	5	long, sleeve, shirt, zip, casual, athletic, golf, men

Thus, the result of the second step of the proposed method is the sets of tag cores for each group of similar items. For further experimenting, we chose the tag core of the second cluster with the largest quantity of elements. The tag core we experimented with is 'men, solid, dress, shirt, long, sleeves, button, regular'. We presupposed that the tag core describes reference items for further searching on different e-commerce platforms. So, we try to search for items which are similar to the reference item.

Searching for items in accordance with customers' demands is the next step of our pipeline. To perform it, we chose the most popular online shops Anna Baluch (Baluch, 2023), Amazon (2023), Walmart (2023), eBay (2023) and AliExpress (2023). Searching was performed on the first seven pages of mentioned online platforms to check the relevance of created tag core. In our experiments, we use the obtained tag core to create a search query. The total match percentage and match percentage for the first pages of every considered online platform are shown in Table 4. In order to illustrate the results of this experiment, we made screenshots of the search results from different e-commerce platforms. We evaluate the results manually based on descriptions and images obtained from websites. The search results with photos and product titles are presented in Figures 5 to 8.

Table 4. The total match percentage and match percentage for the first pages of every considered online platform (Source: authors' contribution)

Online platform	Obtained results	The 1st-page match, %	Total match, %
Amazon	420	93,33%	79,76%
AliExpress	420	76,67%	70,71%
Walmart	280	92,50%	68,57%
eBay	420	90,00%	78,33%

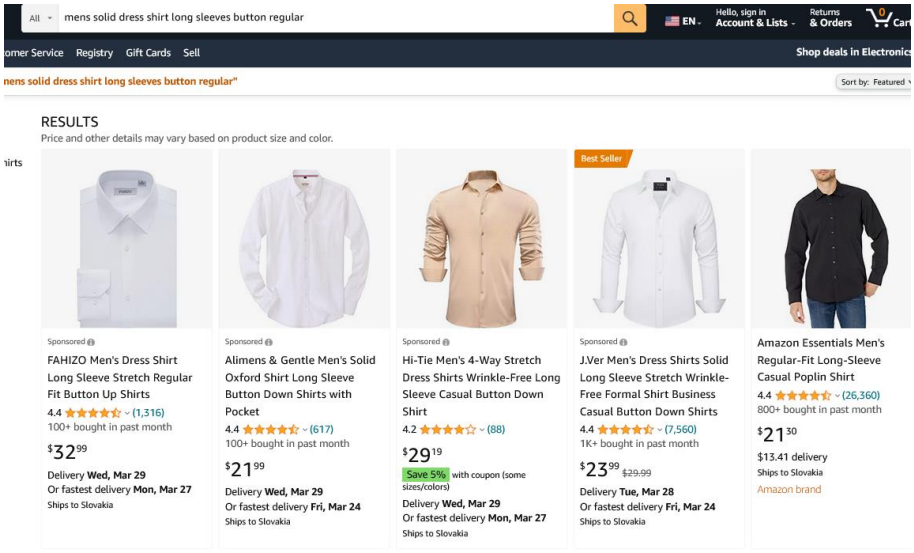


Fig.5. Results of searching query on Amazon (screenshot from Amazon, 2023)

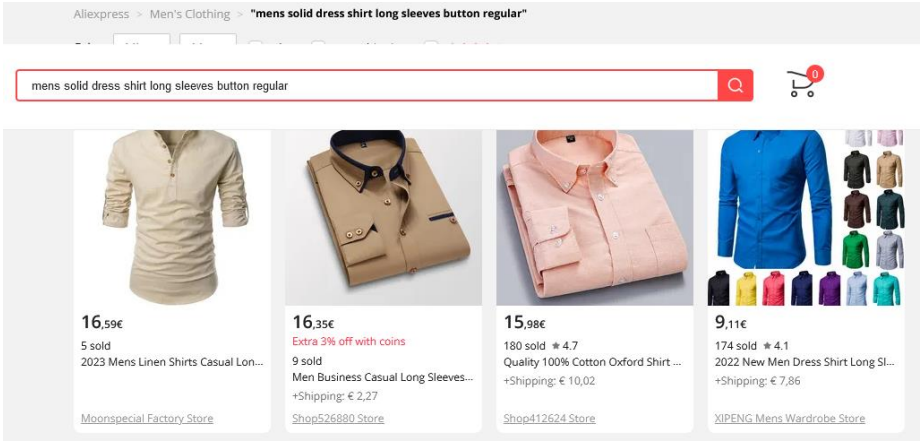


Fig.6. Results of searching query on AliExpress (screenshot from AliExpress, 2023)

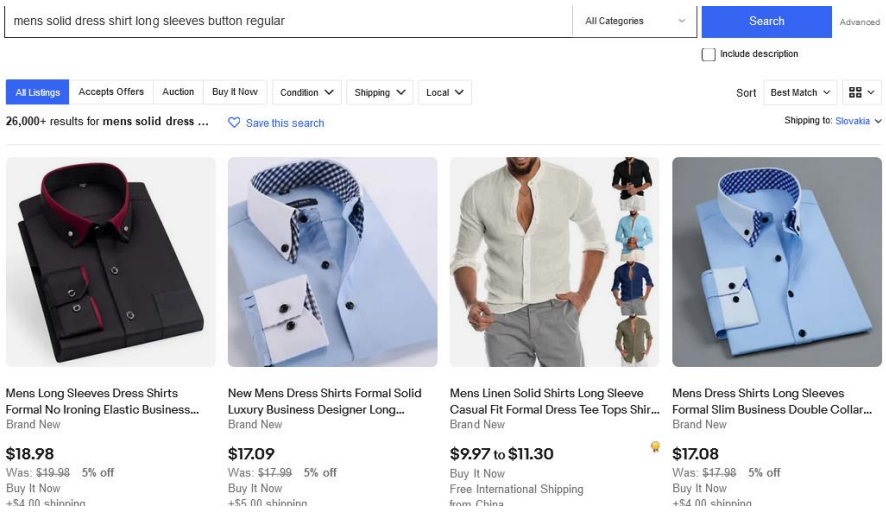


Fig.7. Results of searching query on eBay (screenshot from eBay, 2023)

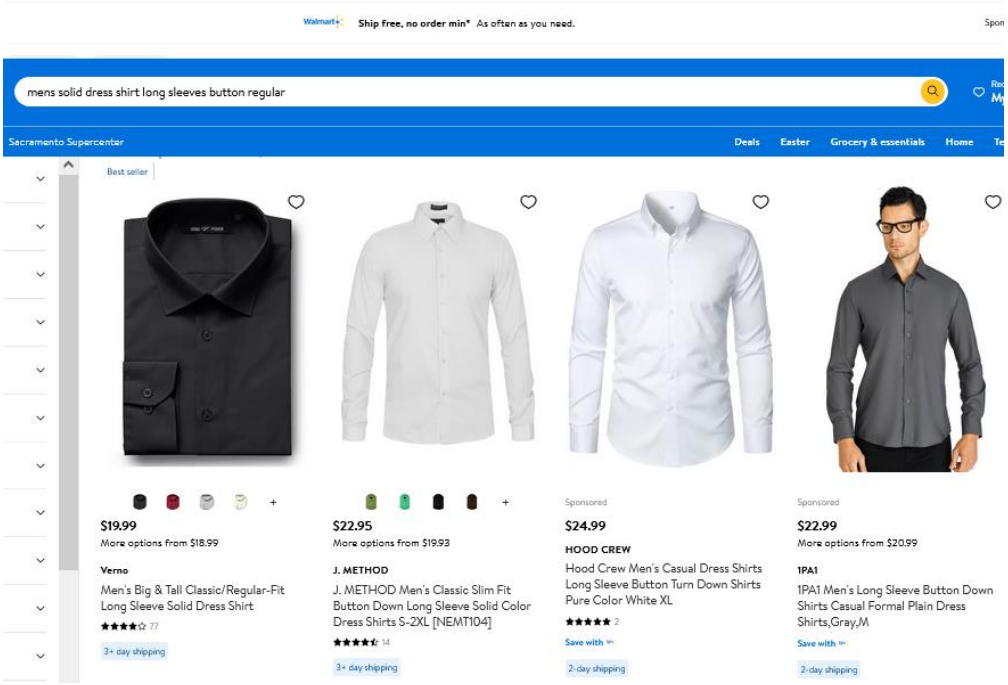


Fig.8. Results of searching query on Walmart (screenshot from Walmart, 2023)

We can see a high percentage, more than 90%, of results, match the first pages of every online platform except AliExpress. The total match percentage for every online platform is lower than for the first page. It is not easy to reach high quality because of the diversity in item descriptions. However, it is 75% on average, which proves the relevance of created tag core for searching for relevant items on different e-commerce platforms. Like the previous experiment (Cherednichenko et al., 2021), the current experiment proves that the best results we obtained on the e-commerce platform where we had collected data.

Discussion and Conclusions

In this work, we propose a three-stage pipeline to identify similar items. As a result of experimenting, we obtained a high match percentage of performing search queries in the most popular online shops (Baluch, 2023) Amazon (2023), Walmart (2023), eBay (2023) and AliExpress (2023). To perform product matching, various techniques can be used, including natural language processing (NLP), machine learning, and fuzzy matching algorithms. NLP techniques can be used to extract relevant information from product names and descriptions, such as keywords and synonyms, while machine learning models can be trained to identify similarities between products based on their text descriptions. Fuzzy matching algorithms can help to match products based on similar names or descriptions, even if they are not exact matches. So, we can summarise that we need to apply different machine-learning techniques to implement the suggested pipeline and improve the results of product matching.

Different e-commerce platforms offer varying methods for sellers to publish their products on the platform. For instance, Amazon, eBay, and others allow users to become sellers and add products themselves, which can increase the number of products available. However, inconsistencies may arise when adding new products and assigning categories. Misclassifying a product can lead to difficulty finding the correct item, making proper categorisation critical for all e-commerce platforms. This ensures efficient product search and better user experience by highlighting the correct categories.

To address these issues, an automatic tool capable of classifying any product based on its title within the product taxonomy is needed. This approach not only facilitates human work but also enhances the consistency of product categorisation on e-commerce websites.

The idea of creating a pipeline for text processing is not new. The analysis of information resources shows a lot of successful implementations, but most of them are not flexible or require special skills from users. Some researchers developed a pipeline that prepared a training dataset for further use of different methods to solve the item-matching problem (Peeters et al., 2020; Primpeli et al., 2019). The Magellan Entity Matching system (Konda, 2018) was developed to solve entity-matching problem, but it was not automated and focused on skilled users. Also, SpaCy (2023) provides users with the opportunity of creating their own pipeline based on default or customer-created functions. The Scikit-learn pipeline (Scikit-learn, 2023) is instrumental when users want to chain multiple estimators into one. In both mentioned pipelines, users are considered to have developer skills. In comparison, the suggested approach consists of all stages of identifying item similarity and searching reference items on different e-commerce platforms; also, it is oriented toward e-commerce managers who are not specialised in programming.

Our approach has the potential to benefit all players in the e-commerce market. For consumers, the proposed method can significantly reduce the time required to find the desired item on different online platforms. The quality of search results will also be significantly improved compared to manual searching, as the machine learning algorithms can effectively identify and group similar items, reducing the number of irrelevant search results.

For sellers, the proposed pipeline can provide valuable insights into their competitors' offers and pricing strategies. By using machine-learning models to search for similar items on competitive online platforms, sellers can easily compare prices and offers to optimise their own marketing strategy. This can help sellers attract new customers and increase sales by providing better value propositions and more appealing designs.

Furthermore, the ability to quickly adapt to changing market trends and customer preferences is critical for e-commerce sellers to remain competitive. With our approach, sellers can easily adjust their marketing strategy to capitalise on new opportunities and entice new customers. This can lead to increased customer loyalty and brand recognition, helping sellers to establish themselves as leaders in their respective markets.

Thus, our proposed method has the potential to significantly enhance the e-commerce experience for both consumers and sellers, making it easier and more efficient to find and sell items online.

In our following works, we plan to explore other algorithms to identify similar items to study the influence of their results on the quality of performing the last step of our pipeline, searching items on e-commerce platforms.

Acknowledgements

The research study presented in this paper is funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project No. 09I03-03-V01-00078 and the project No. 09I03-03-V01-00080

References

AliExpress (2023). *Aliexpress*. https://www.aliexpress.com/

Amazon (2023). *Amazon.* https://www.amazon.com/ref=nav_logo

Akritidis, L., & Bozanis, P. (2018). Effective unsupervised matching of product titles with k-combinations and permutations. *2018 Innovations in Intelligent Systems and Applications (INISTA)*, 1-10. https://doi.org/10.1109/INISTA.2018.8466294

Baluch, A. (2023). *38 Ecommerce Statistics of 2023*. https://www.forbes.com/advisor/business/ecommerce-statistics/

Cherednichenko, O., Ivashchenko, O., & Vovk, M. (2022). Towards pipeline construction for product matching task. *Proceedings of the 41th International Conference on Organizational Science Development,* 165-177. https://doi.org/10.18690/um.fov.3.2022.12

Cherednichenko, O., Vovk, M., Ivashchenko, O., Baggia, A., & Stratiienko, N. (2021). Improving item searching on trading platform based on reinforcement learning approach. *COLINS*. https://ceur-ws.org/Vol-2870/paper106.pdf

Cherednichenko, O., Vovk, M., & Ivashchenko, O. (2020a). Item matching based on collection and processing customer perception of images. https://ceur-ws.org/Vol-2732/20200329.pdf

Cherednichenko, O., Vovk, M., Kanishcheva, O., & Godlevskyi, M. (2018). Towards improving the search quality on the trading platforms. *Information Systems: Research, Development, Applications, Education. SIGSAND/PLAIS 2018. Lecture Notes in Business Information Processing*, 333. https://doi.org/10.1007/978-3-030-00060-8_2

Cherednichenko, O., Yanholenko, O., & Kanishcheva, O.V. (2020b). Developing the key attributes for product matching based on the item's image tag comparison. *MoMLeT+DS.* https://ceur-ws.org/Vol-2631/paper18.pdf

Darú, G. H., Motta, F. D. da M., Castelo, A., & Loch, G. V. (2022). Short text classification applied to item description: Some methods evaluation. *Semina: Ciências Exatas E Tecnológicas, 43(2),* 189–198. https://doi.org/10.5433/1679-0375.2022v43n2p189

eBay (2023). *eBay.* https://www.ebay.com/

eCommerce in Europe (2023). *Trend report.* https://ecommercedb.com/reports/ecommerce-in-europe/589

Gopalakrishnan, V., Iyengar, S. P., Madaan, A., Rastogi, R., & Sengamedu, S. (2012). Matching product titles using web-based enrichment. *Proceedings of the 21st ACM International Conference on Information and Knowledge Management,* 605-614. https://doi.org/10.1145/2396761.2396839

Kannan, A., Givoni, I. E., Agrawal, R., & Fuxman, A. (2011). Matching unstructured product offers to structured product specifications. *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 404-412. https://doi.org/10.1145/2020408.2020474

Konda, P. V. (2018). *Magellan: Toward building entity matching management systems.* [Doctoral thesis, University of Wisconsin-Madison]. University of Wisconsin-Madison.

Köpcke, H., Thor, A., & Rahm, E. (2010). Evaluation of entity resolution approaches on real-world match problems. *Proceedings of the VLDB Endowment, 3,* 484 - 493. https://doi.org/10.14778/1920841.1920904

Köpcke, H., Thor, A., Thomas, S., & Rahm, E. (2012). Tailoring entity resolution for matching product offers. *Proceedings of the 15th International Conference on Extending Database Technology,* 545-550. https://doi.org/10.1145/2247596.2247662

Krzeszewska, U., Poniszewska-Maranda, A., & Ochelska-Mierzejewska, J. (2022). Systematic comparison of vectorization methods in classification context. *Applied Sciences, 12,* 5119. https://doi.org/10.3390/app12105119

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems, 26*. https://papers.nips.cc/paper_files/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf

Mudgal, S., Li, H., Rekatsinas, T., Doan, A., Park, Y., Krishnan, G., & Raghavendra, V. (2018). Deep learning for entity matching: A design space exploration. *Proceedings of the 2018 International Conference on Management of Data,* 19-34. https://doi.org/10.1145/3183713.3196926

Peeters, R., Bizer, C., & Glavas, G. (2020). Intermediate training of BERT for product matching. DI2KG. https://ceur-ws.org/Vol-2726/paper1.pdf

Primpeli, A., Peeters, R., & Bizer, C. (2019). The WDC training dataset and gold standard for large-scale product matching. *Companion Proceedings of the 2019 World Wide Web Conference*, 381-386. https://doi.org/10.1145/3308560.3316609

Singh, A. K., & Shashi, M. (2019). Vectorisation of text documents for identifying unifiable news articles. *International Journal of Advanced Computer Science and Applications, 10(7),* 305–310. https://doi.org/10.14569/ijacsa.2019.0100742

Scikit-learn (2023). *Scikit-learn. Machine learning in Python*. https://scikit-learn.org

SpaCy (2023). *Industrial-strength natural language processing*. https://spacy.io

Wachsmuth, H. (2015). *Text analysis pipelines. Towards ad-hoc large-scale text mining.* Springer Cham. https://doi.org/10.1007/978-3-319-25741-9

Wachsmuth, H., Rose, M., & Engels, G. (2013). Automatic pipeline construction for real-time annotation. *Computational Linguistics and Intelligent Text Processing. CICLing 2013. Lecture Notes in Computer Science, 7816.* https://doi.org/10.1007/978-3-642-37247-6_4

Walmart (2023). *Walmart.* https://www.walmart.com/

Yang, X., Yang, K., Cui, T., Chen, M., & He, L. (2022). A study of text vectorization method combining topic model and transfer learning. *Processes, 10(2),* 350. https://doi.org/10.3390/pr10020350

Yang, Y., Liu, Y., Lu, X., Xu, J., & Wang, F. (2020). A named entity topic model for news popularity prediction. *Knowledge-Based Systems, 208,* 106430. https://doi.org/10.1016/j.knosys.2020.106430