

PRODUCT NAME MATCHING

Jan Přichystal¹

¹Department of Informatics, Faculty of Business and Economics, Mendel University in Brno, Zemědělská 1, 613 00 Brno, Czech Republic

DOI: https://doi.org/10.11118/ecos2026_028

ABSTRACT

The problem of product name matching belongs to the area of entity matching which is one of the data integration tasks. It plays an important role in the analysis of data available on the internet and it combines various techniques and data preprocessing. Gathering and analyzing customer reviews about products published on the internet is challenging task, because particular e-shops offer the products under a bit different names and it could be complicated to match identical products. The article aims at selecting the identical products identified by their names and introduces methods for product name matching based on machine learning using the combination of token-based and character-based functions.

Keywords: customer reviews, product name, entity matching, machine learning

JEL: L86

1 INTRODUCTION

The string-matching problem appears in many applications, e.g. systems for the automated detection of possible plagiarism, data cleaning, information retrieval or comparison of DNA/protein sequences. Such systems describe Wise (1993), Pandit *et al.* (2020) or Moussiades (2005). In this article we focus on finding similar strings in set of product names offered by e-shops. The motivation is we need to union the reviews of products published by the customers. There we try to identify discussed topics and opinion and help other customers to select optimal product without the need to read all customers reviews on the internet. Similar problem is discussed by Yao (2016). Methods of finding string similarities are listed in Deng (2013). Bergsma and Kondrak (2007) propose an alignment-based discriminative framework for string similarity. They gather features from substring pairs consistent with a character-based alignment of the two strings. Yu (2016) present an extensive set of algorithms for string similarity search and join. Wang (2011) propose a new similarity metrics, called “fuzzy token matching based similarity”, which extends token-based similarity functions (e.g., Jaccard similarity and Cosine similarity) by allowing fuzzy match between two tokens. Kopcke *et al.* (2010) investigate the degree the machine learning helps semi-automatically determine suitable match strategies with a limited amount of manual training effort. The product pairing problem on web portal Zboží.cz discusses Bumba (2021). Their approach is based on usage of the CatBoost library developed by the machine learning team at Yandex. This library deals with the generation of



This article is licensed under CC BY-SA 4.0 Attribution-ShareAlike 4.0 International license. <https://creativecommons.org/licenses/by-sa/4.0/deed.en>

so-called gradient boosted decision forests. Generating decision forests is supervised machine learning method, so they need to have enough annotated data to train the model. Then they generate signals (features) from this data. The trained model is using 77 different signals, which arise from various comparisons of the attributes of the offer and the product candidate, such as title, description, product number, price, EAN, ISBN, manufacturer and others. Although the approach looks promising, the disadvantage is obvious – it is not automatic but supervised.

2 DATA

The data was downloaded from various web portals including Czech and international shops. We work with two types of collections. First collection contains the complete list of products in certain segment – the reference set. The second collection is the set of downloaded reviews for each product in the segment identified by the product name. To get the widest possible data set of products and reviews we focused on the segment of cell phones, digital cameras and lenses. The primary reference set containing the list of all introduced products was obtained from mobilmania.cz (cell phones), digicamdb.com (digital cameras) and dpreview.com (lenses). The reviews of products were downloaded from czc.cz, alza.cz, heureka.cz, fotoguru.cz, mobilmania.cz, michalkupsa.cz. The data was downloaded and saved to NoSQL document-based database MongoDB¹ because of different structures of web pages. Together with the product names and reviews we obtain the product properties such as weight, dimensions and others as described in example below. The idea was to use such properties to simplify the process of matching both sets. Unfortunately, we found the properties are not filled properly on many e-shops, we cannot rely on their accuracy and as a result they rather complicate the pairing process.

An example of record we save into the MongoDB database could be seen below:

```
{
  "name": "Samsung Galaxy A10s",
  "brand": "Samsung",
  "dimensions": "157×76×7.8",
  "weight": "168",
  "battery_capacity": "4000",
  "screen_size": "6.2",
  "screen_resolution": "720×1520",
  "introduced": "2019"
}
```

Problems with matching product names arise mainly because different sellers offer products under slightly different names. It is always a single product, but is identified by different names that omit or add certain product features. This problem could be expressed by the examples listed below. The first set shows the product “Samsung Galaxy S10+” different variants that could be found in various e-shops. The last two items show different products, but with similar product name, they can cause the confusion and wrong assignment. For example, the similarity of “Samsung Galaxy S10 Plus black” could be higher to “Samsung Galaxy S10 128 GB black” than “Samsung Galaxy S10+” which is in the reference set.

¹<https://www.mongodb.com/>

- Samsung Galaxy S10+
- Samsung Galaxy S10 Plus
- Samsung Galaxy S10 Plus black
- Samsung Galaxy S10+ 128GB
- Samsung Galaxy S10+ Dual SIM černá
- Samsung Galaxy S10+, 8GB/128GB, černá
- Samsung Galaxy S10+ 128GB+8GB RAM SM-G975F/DS Dual Sim 6.4" LTE
- Samsung Galaxy S10 128 GB black
- Samsung Galaxy S10e

As can be seen in list above the product name consists of separate parts. If we focus on category of mobile phones the product name usually starts with the name of the manufacturer. In our example it is “Samsung”. Then follows the product line – “Galaxy”, the product itself – S10. Then there could be product model, here it is symbolized by “Plus” or “+”. The vendors sometimes add more information about color – “black”, memory size – “128GB” or “8GB/128GB”, “Dual SIM”, product code – “SM-G975F/DS”, display size – “6.4” or network capability – “LTE”.

It is obvious the product name could vary and the same product may have significantly different product name and on the other hand the different products may have almost identical names. For example, if we use simple character-based function to compare such strings it finds higher similarity between “Samsung Galaxy S10+” and “Samsung Galaxy S10e” than between “Samsung Galaxy S10+” and “Samsung Galaxy S10+, 8GB/128GB, černá”. This causes problems in identifying and matching products.

The example below shows the same problem again. The similarity of “Apple iPhone Xs 64 GB gold” will be higher to “Apple iPhone X 64GB gold” than to “Apple iPhone Xs”.

- Apple iPhone Xs
- Apple iPhone Xs 64 GB gold
- Apple iPhone X 64GB gold

Sometimes we face the problem of identifying different product versions expressed by the year. The example below shows the different products (different versions for two years) but with the same name. The problem could appear when the e-shop doesn’t reflect the strategy of Samsung company and uses only the name Samsung Galaxy A3. Then identifying of right customers reviews could be problematic.

- Samsung Galaxy A3 (2016)
- Samsung Galaxy A3 (2017)

Such problems are more emphasized in the digital camera lenses as could be seen in the example below where each item is different product. The last item is the general product name sometimes used in e-shops. Then the assignment is also complicated.

- Canon EF-S 18-55mm F4-5.6 IS STM
- Canon EF-S 18-55mm f/3.5-5.6 IS STM
- Canon EF-M 18-55mm f/3.5-5.6 IS STM
- Canon EF-S 18-55mm f/3.5-5.6 III
- Canon EF-S 18-55mm f/3.5-5.6 IS II
- Canon EF-S 18-55mm f/3.5-5.6 IS
- Canon EF-S 18-55mm f/3.5-5.6 II
- Canon EF-S 18-55mm f/3.5-5.6

The last example shows the list of digital cameras where the first item is particular product but sometimes used for the rest of products listed below. The names distinguish the evolution of product during the years.

- Sony Cyber-shot DSC-RX100
- Sony Cyber-shot DSC-RX100 II
- Sony Cyber-shot DSC-RX100 III
- Sony Cyber-shot DSC-RX100 VA
- Sony Cyber-shot DSC-RX10

3 METHODS AND RESULTS

Existing studies mainly use the following functions to quantify similarity of two strings (Wang, 2011):

- **Token-based similarity functions:** They first tokenize strings as token sets (“bag of words”), and then quantify the similarity based on the token sets, such as Jaccard similarity and Cosine similarity. Usually if two strings are similar, their token sets should have a large overlap. Token-based similarity functions have a limitation that they only consider exact match of two tokens and neglect fuzzy match of two tokens.
- **Character-based similarity functions:** They use characters in the two strings to quantify the similarity, such as edit distance which is the minimum number of single-character edit operations (i.e., insertion, deletion, and substitution) needed to transform one to another. In comparison with token-based similarity, edit distance is sensitive to the positions of the tokens in a string.

The similarity of strings could be also measured as the longest similar substring – **sequence-based similarity**. Such methods are longest common subsequence similarity, longest common substring similarity or Ratcliff-Obershelp similarity. (Mayank, 2019)

The algorithms and approaches mentioned in the introduction doesn’t provide sufficient results. The problem appears in minor differences of distinct product names and major differences in names of the same product. Also, the constantly changing set of reference products plays important role and prevents the creation of training set.

We decided to propose new algorithm avoiding listed problems. The algorithm first identifies suitable candidates and then chooses the right one.

Before matching the product names, it is necessary to preprocess the data to increase the chance of match. We apply removing of the special characters e.g. commas or apostrophes, replacing the slash with space and lowering the case of text. The “Samsung Galaxy S10+, 8GB/128GB” is transformed to “samsung galaxy s10+ 8gb 128gb”.

An algorithm itself consists of these steps:

- (1) Finding 100% match – if the product name from review set is completely equal the algorithm assigns it directly.
- (2) Tokenization and Sorensen-Dice (Sørensen, 1948) – the product names of both sets are tokenized by the spaces and the Sorensen-Dice score is calculated for each product from reference set. Only the products of certain company (e.g. Samsung) are selected. Top 5 products with the highest score are selected.
- (3) Longest common substring similarity (LCSTR) – for the purpose of score determination the spaces in the string are removed to avoid false similarity. The longest match of the product

from review set and products selected in previous step is calculated. The product with the longest match is picked. If there are two or more products with the same longest match value, the product with higher Sorensen-Dice score is selected.

3.1 Results

The table below shows the partial results of process of identification of similar products. In the first column there is a title of product from the reference set saved in the database we use to match product name from downloaded review. The second column shows the similarity score of reference title and the name of product being compared calculated by Sorensen-Dice in 2. step. The third column shows the longest substring of product name from downloaded review matched to reference title in 3. step. The last column presents the score calculated in 3. step using LCSTR function. The product name from downloaded review we try to match is “Xiaomi Mi 8 Pro 128GB transparentní”.

Reference title	Sorensen-Dice score	Result substring	Substring score
xiaomi mi 8 pro	0.80	xiaomimi8pro	12
xiaomi mi 8 128gb	0.80	xiaomimi8	09
xiaomi mi 8 lite 128gb	0.73	xiaomimi8	09
xiaomi mi 8	0.67	xiaomimi8	09
xiaomi mi 8 lite	0.60	xiaomimi8	09

The matched product title from reference set is therefore “Xiaomi Mi 8 Pro” – the first row in table with the highest substring score.

The second example shows matching of the product name downloaded from review “Tamron AF 18-200mm f3.5-6.3 Di II VC pro Nikon”.

Reference title	Sorensen-Dice score	Result substring	Substring score
tamron 18-200mm f3.5-6.3 di ii vc	0.800	18-200mmf3.5-6.3diiivc	22
tamron 18-200mm f3.5-6.3 di iii vc	0.660	18-200mmf3.5-6.3diii	20
tamron 18-270mm f3.5-6.3 di ii vc pzd	0.625	0mmf3.5-6.3diiivcp	18
tamron 18-400mm f3.5-6.3 di ii vc hld	0.625	00mmf3.5-6.3diiivc	18
tamron af 18-200mm f3.5-6.3 xr di ii ld aspherical (if) macro	0.600	af18-200mmf3.5-6.3	18

The matched product title from reference set is therefore “Tamron 18-200mm f3.5-6.3 Di II VC” with the highest substring score.

4 DISCUSSION

The proposed algorithm has more than 90% efficiency when all items in reviews set are also in reference product set. If there appear the products that are not listed in reference set we face the problem with false positives. It means when the product is not listed another (with similar name) product is assigned. For example, to the “Aligator S6000 Duo” the “Aligator s5070 Duo” is assigned.

The algorithm was implemented using the Python programming language using the `textdistance`² library.

We still make experiments with the combination of methods and their settings. Now we focus to third step of our algorithm, where the longest common substring does not provide sufficient results. The aim is to have different weights for different parts of string, because similarity of certain parts of string is less important (e.g. “mm” or “GB”).

The problem also appears in situation where the product specification influences the tokenization of product name e.g. “Xiaomi Mi9” vs “Xiaomi Mi 9”.

We also experiment with the information gathered from e-shop aggregators like Heureka³, where the list of different product names for each product can be found on individual e-shops.

Acknowledgements

This paper was supported by the project CZ.02.1.01/0.0/0.0/16_017/0002334 Research Infrastructure for Young Scientists, this is co-financed from Operational Programme Research, Development and Education and the project EG16_084/0010315 – Multicriterial Text Analysis Software (2017–2020, MPO/EG).

REFERENCES

- BERGSMA, S., KONDRÁK, G. 2007. Alignment-based discriminative string similarity. In: *Proceedings of the 45th annual meeting of the association of computational linguistics*, pp. 656–663.
- BUMBA, M. 2021. Párování zboží pomocí CatBoost a praktické zkušenosti ze Zboží.cz. *Root.cz*. [Accessed: 14 January 2021] <https://www.root.cz/clanky/parovani-zbozi-pomoci-catboost-a-prakticke-zkusenosti-ze-zbozi-cz/>. 2021
- DENG, D. *et al.* 2013. Top-k string similarity search with edit-distance constraints. In: *2013 IEEE 29th International Conference on Data Engineering (ICDE)*. IEEE, pp 925–936.
- KOPCKE, H., THOR, A., RAHM, E. 2010. Learning-based approaches for matching web data entities. *IEEE Internet Computing*. 14(4), 23–31.
- MAYANK, M. 2019. *String similarity — the basic know your algorithms guide!* [Accessed: 11 November 2020] <https://itnext.io/string-similarity-the-basic-know-your-algorithms-guide-3de3d7346227>. 2019
- MOUSSIADES, L., VAKALI, A. 2005. PDetect: A clustering approach for detecting plagiarism in source code datasets. *The computer journal*. 48(6), 651–661.
- PANDIT, A. A., TOKSHA, G. 2020. Review of Plagiarism Detection Technique in Source Code. In: *International Conference on Intelligent Computing and Smart Communication 2019*. Springer, Singapore, pp. 393–405.

²<https://pypi.org/project/textdistance/>

³<https://www.heureka.cz>

- SØRENSEN, T. *et al.* 1948. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons. *K. Dan. Vidensk. Selsk. Biol. Skr.* 5, 1–34
- YAO, Y., SUN, A. 2016. Mobile phone name extraction from internet forums: a semi-supervised approach. *World Wide Web*. 19(5), 783–805.
- YU, M. *et al.* 2016. String similarity search and join: a survey. *Frontiers of Computer Science*. 10(3), 399–417.
- WANG, J., LI, G., FE, J. 2011. Fast-join: An efficient method for fuzzy token matching based string similarity join. In: *2011 IEEE 27th International Conference on Data Engineering*. IEEE, pp. 458–469.
- WISE, M. J. 1993. *String similarity via greedy string tiling and running Karp-Rabin matching*. Online Preprint, Dec, 1993, 119(1), 1–17.

Contact information

Jan Přichystal: e-mail: jan.prichystal@mendelu.cz