Text-Based Product Matching with Incomplete and Inconsistent Items Descriptions

Szymon Łukasik[1,2,3][0000−0001−6716−610X], Andrzej Michałowski[3], Piotr A. Kowalski[1,2][0000−0003−4041−6900], and Amir H. Gandomi[4][0000−0002−2798−0104]

[1] Faculty of Physics and Applied Computer Science, AGH University of Science and Technology, al. Mickiewicza 30, 30-059 Kraków, Poland
{slukasik,pkowal}@agh.edu.pl
[2] Systems Research Institute, Polish Academy of Sciences, ul. Newelska 6, 01-447 Warsaw, Poland
{slukasik, pakowal}@ibspan.waw.pl
[3] Synerise PLC, ul. Lubostroń 1, 30-383 Kraków, Poland
{szymon.lukasik, andrzej.michalowski}@synerise.com
[4] Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW 2007, Australia
gandomi@uts.edu.au

Abstract. In recent years Machine Learning and Artificial Intelligence are reshaping the landscape of e-commerce and retail. Using advanced analytics, behavioral modeling, and inference, representatives of these industries can leverage collected data and increase their market performance. To perform assortment optimization – one of the most fundamentals problems in retail – one has to identify products that are present in the competitors' portfolios. It is not possible without effective product matching. The paper deals with finding identical products in the offer of different retailers. The task is performed using a text-mining approach, assuming that the data may contain incomplete information. Besides the description of the algorithm, the results for real-world data fetched from the offers of two consumer electronics retailers are being demonstrated.

Keywords: product matching · assortment · missing data · classification

1 Introduction

Over the last ten years, the share of people ordering goods or services online increased significantly. In the United States between 2010 and 2019, e-commerce sales as a percent of total retail sales tripled and reached 12 percent [24]. The impact of machine learning and artificial intelligence in this context cannot be neglected. Over 70% of surveyed tech executives see the positive impact of AI on the retail industry, and 80% state that the use of AI allows maximizing profits [8]. Typically machine learning is currently being used for personalization of services, managing relationships with customers and enterprise resource planning [23, 26, 12].

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_8

Optimizing assortment corresponds to the problem of selecting a portfolio of products to offer to customers to maximize the revenue [3]. Similarly, price optimization represents a problem of finding the best pricing strategy that maximizes revenue or profit based on demand forecasting models [13]. Both are crucial for gaining a competitive advantage in the market. To solve the aforementioned issues, it is important to match products present in the offers of competitors to the ones sold by the analyzed retailer. Such product matching is not a trivial problem as it involves using the information provided by different retailers. Its accuracy and completeness varies. Therefore designed algorithms have to deal with different data types, missing values, imprecise/erroneous entries.

The task of this paper is to present a new algorithm of product matching. It is based on analyzing product descriptions with text-mining tools. The matching problem is formulated here as an instance of classification. Our approach uses a decision-tree ensemble with gradient boosting as the learning algorithm. Together with chosen text similarity measures, it allows us to successfully deal with missing values and imprecise product descriptions. It is important to note that we assume here that no structured data regarding technical product attributes are available for both retailers – which is frequently the case.

The paper is organized as follows. In the next Section, an overview of existing approaches to product matching is being provided. It is followed by a brief summary of text similarity measures and methods of word embeddings. These elements serve as the methodological preliminary to the description of the proposed algorithm presented in Section 3. It is followed by the results of the first studies on the algorithm's performance. Finally, the last Section of the paper contains a conclusion and plans for further research.

2 Related work

2.1 Product matching

Product matching, in essence, can be perceived as an instance of a much broader problem of entity matching. Given two sets of products A and B – also known as product feeds – coming from two different sources (retailers) S_A and S_B, the task of product matching is to identify all correspondences between products in A and B. Such correspondence means that two matched products listed in A and B represent the same real-world object. The match result is typically represented by a set of correspondences together with similarity values $s_{ij} \in [0, 1]$ indicating the similarity or strength of the correspondence between the two objects [17].

There are not many machine learning papers dealing strictly with product matching. In [16] authors perform sophisticated feature engineering along with identifying product codes to improve matching performance. In [22] an approach using Siamese networks standard neural network using fastText (on concatenated product descriptions) is being tested. It also uses global product identifiers (if available). In [1] one can find an interesting application of fuzzy matching – with basic similarity measures (e.g. Q-grams and exact string matching). Study [20] is

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_8

the most complete one of the analyzed here. It uses both textual information as well as images of products (feeding convolutional neural network). It uses standard Word2Vec embeddings and offers average matching performance. Finally, when overviewing literature on product matching it is worth noting here that there are no publicly available datasets, allowing us to test and compare various approaches to product matching. Most of the studies are using customized datasets – obtained or generated by authors themselves.

Product matching can be reformulated as a task of binary classification – its goal is to assign element x_{ij} to one of the two classes/categories with a known set of representative elements (known as the training set T) [5]. For product matching x_{ij} should represent a set of matching descriptors calculated for product i from feed A and product j from feed B. Such descriptors should essentially correspond to the similarity of products – calculated for different product attributes. Classification outcome for a given pair should be either crisp decision – match or no match, or matching probability value $p_{ij} \in [0, 1]$ corresponding to product similarity mentioned above.

There exists a variety of classification approaches that might be used in the aforementioned problem. Among others Support Vector Machines (SVM) [10], neural networks [9] or decision trees [18] could be named here.

As one of the typical features of product feeds is the incompleteness of product descriptions, the algorithms that are designed to deal with missing values are strongly preferred. Here we propose to use the XGBoost algorithm [6] based on decision-tree ensemble with gradient boosting. One of its essential features is treating the non-presence of attribute value as a missing value and learning the best direction to handle such a situation.

In the subsequent part of the paper, we will treat product description as a minimal set of features shown in Fig. 1 along with the exemplary product. The feature set contains the product name, brand, product type (or in other words category it belongs to). Besides these textual features, the product price is also available. Problems with the accuracy of such description can manifest in the form of:

1. errors in product's titles and overusing it to list product's attributes;
2. different structures of categories between two considered feeds;
3. missing brand;
4. hidden components of the price (e.g. costs of delivery).

To overcome these difficulties, product matching descriptors have to employ soft similarity measures allowing to compare not only the strict set of words used in the descriptions but also their semantic context. Possible solutions with this respect will be covered in the next subsection of the paper.

2.2 Text Similarity Measures

Finding similarities between texts is an essential task for a variety of text-mining applications. It can be perceived as a hierarchical procedure – with the similarity of individual words situated at the bottom level of this structure. It is fundamental for analyzing higher-level similarities: of sentences, paragraphs and,

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_8

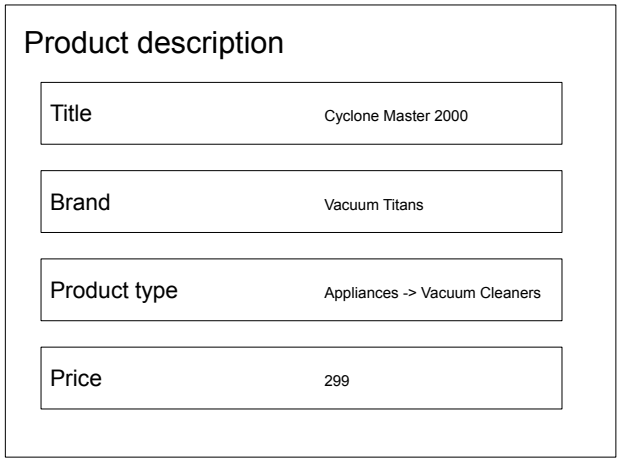

Fig. 1. Elements of product description

documents. Words can be similar in two aspects: lexical and semantic. The first corresponds to the occurrence of similar character sequences. On the other hand, words are similar semantically if they are associated with the same/opposing concepts or representing different kinds of objects of the same type. Lexical similarity can be analyzed with string-based approaches, while semantic one is usually tackled with corpus-based and knowledge-based algorithms. For a more detailed outlook on the text similarity measures, one could refer to the survey paper [11]. Here we will concentrate on lexical similarity as semantic context will be captured with the additional tool – covered in the next subsection.

The string-based similarity is calculated either on a character basis or using terms forming a string. The first approach tries to quantify differences between the sequences of letters. One of the well-known technique of this type is Levenshtein distance. Levenshtein distance between two words corresponds to the minimum number of single-character edits (insertions, deletions or substitutions) required to change one word into the other [27]. Damerau-Levenshtein distance represents a generalization of this concept – by including transpositions among its allowable operations [7].

As one of the most popular techniques of term-based similarity evaluation Jaccard similarity could be named [14]. It is calculated by:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| + |A \cap B|}. \tag{1}$$

It means that to obtain its value for text matching, one has to divide the number of words present in both words by the number of words present in either of the documents.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_8

2.3 Word Embeddings

The aim of word embedding is to find a mapping from a set of words to a set of multidimensional – typically real-valued – vectors of numbers. A useful embedding provides vector representations of words such that the relationship between two vectors mirrors the linguistic relationship between the two words [21]. In that sense, word embeddings can be particularly useful for comparing the semantics of short texts.

The most popular approach to word embeddings is Word2Vec technique [19], which can build embeddings using two methods: CBOW model, which takes the context of each word as the input and tries to predict the word corresponding to the context and skip-gram model. The latter takes every word in large corpora and also takes one-by-one the words that surround it within a defined window to then feed a neural network that after training will predict the probability for each word to actually appear in the window around the focus word [25].

fastText is a more recent approach based on the skipgram model, where each word is represented as a bag of character n-grams. A vector representation is associated with each character n-gram, with words being represented as the sum of these representations. What is important from the point of product matching fastText allows us to compute word representations for words that did not appear in the training data [4]. fastText, by default, allows us to obtain an embedding with 300 real-valued features, with the models for a variety of languages already built.

3 Proposed approach

The algorithm introduced here employs only textual features listed in Fig. 1 – which are available for almost all product feeds. Matching descriptors are formed of real-valued vectors consisting of 6 features:

1. The first element of the feature vector corresponds to the Jaccard similarity obtained for titles of tested products from Feed A and Feed B. It is aimed at grasping the simplest difference between products – in their names.
2. The second is calculated as the cosine similarity between 300-dimensional fastText embeddings of product types, with prebuilt language models corresponding to the retailer's business geographic location. In this way, synonyms used by different retailers in product categories are being properly handled.
3. The third feature corresponds to the Damerau-Levenshtein similarities calculated for the brand of two products under investigation. It allows us to ignore any typos or misspellings.
4. Fourth element is calculated as a relative price difference:

$$\frac{|priceA - priceB|}{\max(priceA, priceB)} \tag{2}$$

and it is inspired by the fact that the same product sold by two different retailers would be priced similarly for both of them.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_8

5. Finally, two last features are built using 2-dimensional embedding with Principal Components Analysis [15] for the product type. It is obtained from the 300-dimensional fastText embeddings. It allows us to differentiate the impact of other features between different product categories.

Constructed features serve as an input to classifier – pretrained using manually processed dataset containing feature values and corresponding matching labels. As a classifier extreme gradient boosting ensemble (XGboost) is being used. Thanks to the sparsity-aware split finding algorithm, it is able to accurately handle missing values (denoted as NaN), which might be present in the dataset [2].

Figure 2 summarizes the detailed scheme of the algorithm.

4 Experimental Results

We have preliminarily tested the proposed approach on the real-world instance of product matching. To achieve this goal, we have been using two product feeds – obtained by web-scraping – from two EU-based consumer electronics retailers.

Feed A consists of 130 000 products, Feed B - of 30 000 items. Such an imbalance of feeds sizes makes the problem even more challenging. In addition to that, their category trees are not consistent. Examples of observed inconsistencies were shown in Tab. 1. It can be seen that the same group of products are not located for Feed A and Feed B under the same path in their category trees.

The matching model was trained using a set of 500 randomly chosen and manually labeled (as match or no match) pairs of products. The testing dataset consisted of 100 randomly chosen products from feed A. For the purpose of subsequent analysis, matchings obtained from the classifier for this set of items were again analyzed manually.

Table 1. Examples of inconsistencies in category trees

Feed A	Feed B
Small appliances > Kitchen - cooking and food preparation > Kitchen utensils > Breadboxes	Small appliances > Kitchen accessories > Breadboxes
Audio-video devices and TV sets > Car Audio and Navigation > Sound > Speakers	Audio-video devices and TV sets > Car audio > Speakers
Parent Zone > Feeding > Breast pumps	Small household appliances > For children¿ Breast pumps
Audio-video devices and TV sets > HiFi Audio > DJ controllers	Audio-video devices and TV sets > Party > DJ controllers
Small household appliances > Home - cleanliness and order > Vacuum cleaners > Standard vacuum cleaners	Small household appliances > Cleaning > Standard vacuum cleaners

Figure 3 demonstrates feature importances during the training phase reported by XGBoost. It can be seen that naturally, product title is a very important factor to take into account while matching products. Still, other descriptors, such as price ratio or category of product being matched, seem to have a non-negligible impact.

Table 2 contains selected pairs with the highest matching probabilities calculated with respect to the product from the main feed A. The first four rows demonstrate examples of a perfect match – when the algorithm is able to recover

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_8

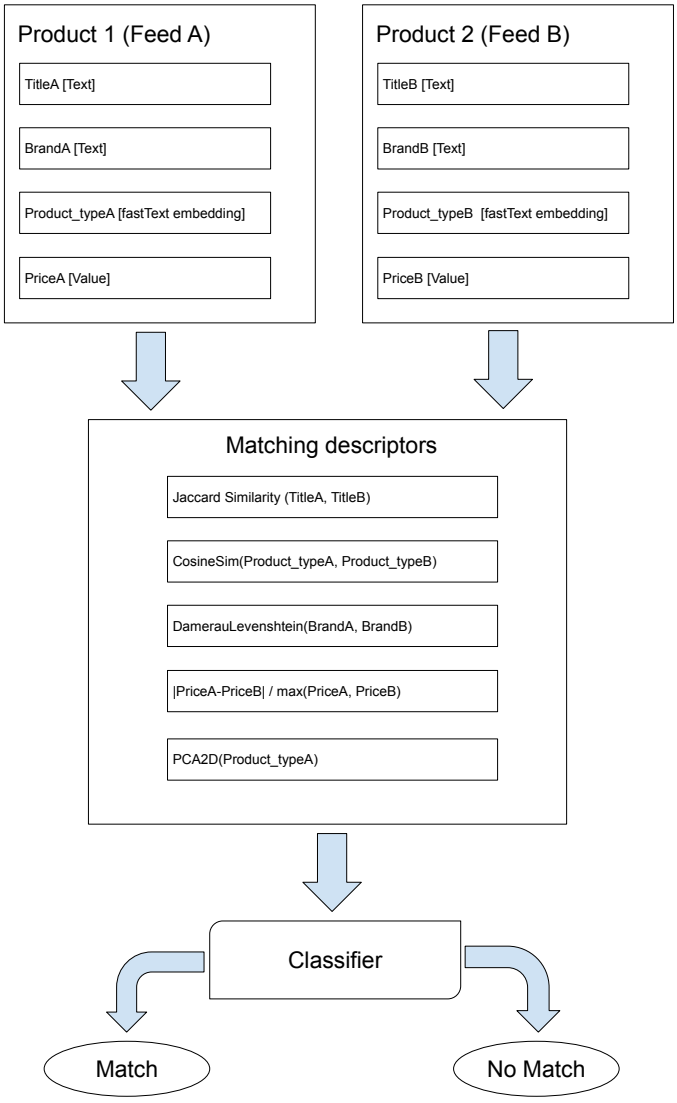

Fig. 2. Proposed product matching algorithm scheme

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_8

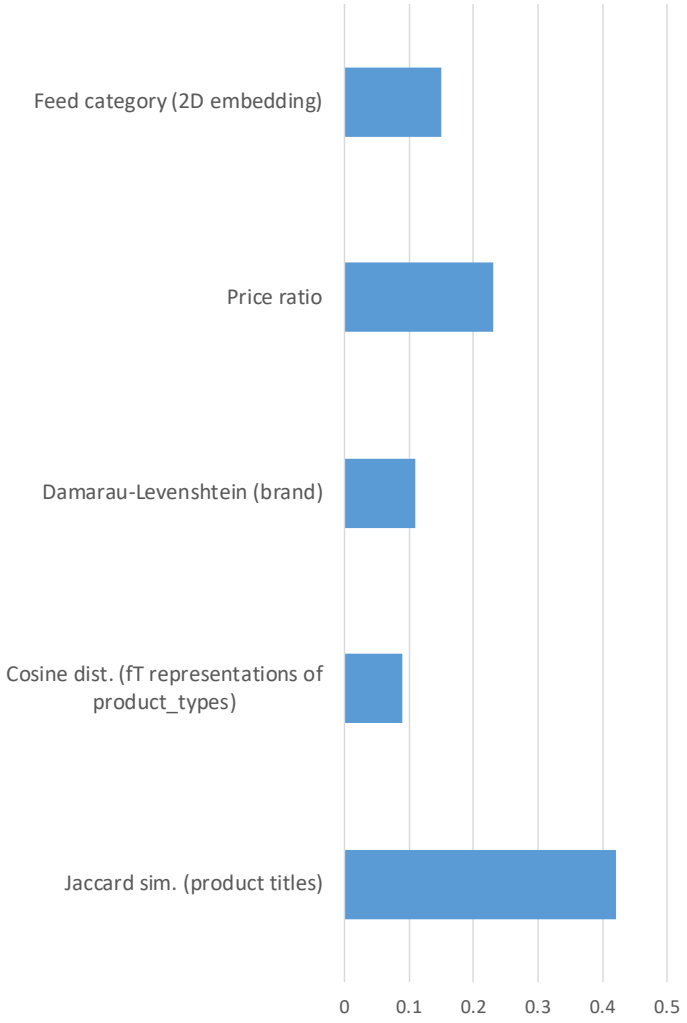

Fig. 3. Importance of features obtained while fitting model to the training dataset

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_8

product similarities very well, even though in some cases, the compatibility for product names is not perfect. The rest of the table presents cases where the algorithm (as a matching probability threshold, a natural value of $p_{ij} > 0.5$ could be chosen) was able to detect a no-match situation. In all of these cases, rejection of matching was not so evident as product descriptions contained a lot of similar elements.

Table 2. Selected matching results including both closest matched pairs and matching probabilities

Product name (feed A)	Product name (feed B)	Matching probability
METROX ME-1497 Black	Coffee grinder METROX ME-1497	0.86
ACER Nitro 5 AMD RYZEN 5 2500U/15.6"/8GB/256GBSSD/AMDRX560X/W10	ACER Nitro 5 (NH.Q3REP.021) Laptop	0.96
CANDY BIANCA BWM4 137PH6/1-S	CANDY BWM4 137PH6 Washing Machine	0.90
BOSCH ADDON MUZ5CC2	Bosch Cube cutter MUZ5CC2	0.83
LAVAZZA Pienaroma 1kg	LAVAZZA Qualita Oro 1kg Blended Coffee	0.28
BOSCH MSM 67190	BOSCH MSM 6S90B Blender	0.45
SANDISK CRUZER ULTRA 16 GB USB 3.0 SDCZ73-016G-G46	SANDISK USB/OTG Ultra Dual 256 GB USB 3.0 Memory	0.11
PHILIPS HD 9126/00	PHILIPS HD 4646/70 Electric Kettle	0.34
APPLE Leather Folio iPhone XS, Peony Pink	APPLE Leather Folio for iPhone X Black	0.50

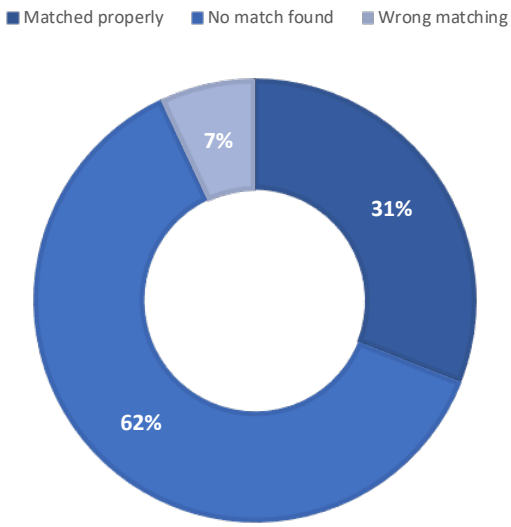

Fig. 4. Statistics of matching results

Figure 4 illustrates the results of matching quantitatively. It can be seen that the majority of products from feed A have not found a match from a correspond-

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_8

ing feed B. It is a natural consequence of inequality in the number of products between those feeds. At the same time, of 38 obtained matches (with probability > 0.5), 31 of them were correct (which accounts for 81% of all matches). It demonstrates the robustness of the method and its reasonable performance also when compared with other studies in the field of product matching (e.g. see [20]).

5 Conclusion

The paper examined the possibility of using text-based machine learning algorithms on sets of product feeds to obtain product matchings. Our goal was to introduce the approach using the minimal amount of information from the product description, at the same time tolerating inaccurate and missing information provided in the product feeds.

Through the first round of experiments performed on the real-world data the algorithm proved to be effective and robust. Further studies will involve using additional, optional matching descriptors such as product images and their technical features. Experiences from the field of product matching will be used for building new category matching algorithms – which are also important from the data presentation standpoint. The existing framework of obtaining product descriptors could also be used for other tasks such as product grouping and categorization.

Acknowledgment

The work was supported by the Faculty of Physics and Applied Computer Science AGH UST statutory tasks within the subsidy of MEiN.

References

1. Amshakala, K., Nedunchezhian, R.: Using fuzzy logic for product matching. In: Krishnan, G.S.S., Anitha, R., Lekshmi, R.S., Kumar, M.S., Bonato, A., Graña, M. (eds.) Computational Intelligence, Cyber Security and Computational Models. pp. 171–179. Springer India, New Delhi (2014)
2. Aulia, D., Murfi, H.: Xgboost in handling missing values for life insurance risk prediction. SN Applied Sciences **2** (08 2020). https://doi.org/10.1007/s42452-020-3128-y
3. Bernstein, F., Kök, A.G., Xie, L.: Dynamic assortment customization with limited inventories. Manufacturing and Service Operations Management **17**, 538–553 (2015)
4. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. CoRR **abs/1607.04606** (2016), http://arxiv.org/abs/1607.04606
5. Charytanowicz, M., Niewczas, J., Kulczycki, P., Kowalski, P.A., Łukasik, S.: Discrimination of wheat grain varieties using x-ray images. In: Pietka, E., Badura, P., Kawa, J., Wieclawek, W. (eds.) Information Technologies in Medicine. pp. 39–50. Springer International Publishing, Cham (2016)

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_8

6. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 785–794. KDD '16, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2939672.2939785, https://doi.org/10.1145/2939672.2939785
7. Damerau, F.J.: A technique for computer detection and correction of spelling errors. Commun. ACM **7**(3), 171–176 (Mar 1964). https://doi.org/10.1145/363958.363994, https://doi.org/10.1145/363958.363994
8. Edelman: 2019 Edelman AI Survey. Whitepaper, Edelman (2019)
9. Faris, H., Aljarah, I., Mirjalili, S.: Training feedforward neural networks using multi-verse optimizer for binary classification problems. Applied Intelligence **45**(2), 322–332 (Sep 2016). https://doi.org/10.1007/s10489-016-0767-1, https://doi.org/10.1007/s10489-016-0767-1
10. Gaspar, P., Carbonell, J., Oliveira, J.: On the parameter optimization of support vector machines for binary classification. Journal of integrative bioinformatics **9**, 201 (07 2012). https://doi.org/10.2390/biecoll-jib-2012-201
11. Gomaa, W., Fahmy, A.: A survey of text similarity approaches. international journal of Computer Applications **68** (04 2013). https://doi.org/10.5120/11638-7118
12. Ismail, M., Ibrahim, M., Sanusi, Z., Cemal Nat, M.: Data mining in electronic commerce: Benefits and challenges. International Journal of Communications, Network and System Sciences **08**, 501–509 (01 2015). https://doi.org/10.4236/ijcns.2015.812045
13. Ito, S., Fujimaki, R.: Large-scale price optimization via network flow. In: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems 29, pp. 3855–3863. Curran Associates, Inc. (2016), http://papers.nips.cc/paper/6301-large-scale-price-optimization-via-network-flow.pdf
14. Ivchenko, G., Honov, S.: On the jaccard similarity test. Journal of Mathematical Sciences **88**(6), 789–794 (1998)
15. Jolliffe, I.: Principal component analysis. Springer Verlag, New York (2002)
16. Köpcke, H., Thor, A., Thomas, S., Rahm, E.: Tailoring entity resolution for matching product offers. In: Proceedings of the 15th International Conference on Extending Database Technology. p. 545–550. EDBT '12, Association for Computing Machinery, New York, NY, USA (2012). https://doi.org/10.1145/2247596.2247662, https://doi.org/10.1145/2247596.2247662
17. Köpcke, H., Rahm, E.: Frameworks for entity matching: A comparison. Data and Knowledge Engineering **69**(2), 197 – 210 (2010). https://doi.org/https://doi.org/10.1016/j.datak.2009.10.003, http://www.sciencedirect.com/science/article/pii/S0169023X09001451
18. Liu, L., Anlong Ming, Ma, H., Zhang, X.: A binary-classification-tree based framework for distributed target classification in multimedia sensor networks. In: 2012 Proceedings IEEE INFOCOM. pp. 594–602 (March 2012). https://doi.org/10.1109/INFCOM.2012.6195802
19. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality (2013)
20. Ristoski, P., Petrovski, P., Mika, P., Paulheim, H.: A machine learning approach for product matching and categorization: Use case: Enriching product ads with semantic structured data. Semantic Web **9**, 1–22 (08 2018). https://doi.org/10.3233/SW-180300

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_8

21. Schnabel, T., Labutov, I., Mimno, D., Joachims, T.: Evaluation methods for unsupervised word embeddings. In: Proceedings of the 2015 conference on empirical methods in natural language processing. pp. 298–307 (2015)
22. Shah, K., Kopru, S., Ruvini, J.D.: Neural network based extreme classification and similarity models for product matching. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers). pp. 8–15. Association for Computational Linguistics, New Orleans - Louisiana (Jun 2018). https://doi.org/10.18653/v1/N18-3002, https://www.aclweb.org/anthology/N18-3002
23. Srinivasa Raghavan, N.R.: Data mining in e-commerce: A survey. Sadhana **30**(2), 275–289 (Apr 2005). https://doi.org/10.1007/BF02706248, https://doi.org/10.1007/BF02706248
24. US Census Bureau: Quarterly retail e-commerce sales. News report CB19-170, US Census Bureau (2019), Nov 19th
25. Vieira, A., Ribeiro, B.: Introduction to Deep Learning Business Applications for Developers: From Conversational Bots in Customer Service to Medical Image Processing. Apress (2018), https://books.google.pl/books?id=K3ZZDwAAQBAJ
26. Yu, G., Xia, C., Guo, X.: Research on web data mining and its application in electronic commerce. In: 2009 International Conference on Computational Intelligence and Software Engineering. pp. 1–3 (Dec 2009). https://doi.org/10.1109/CISE.2009.5363366
27. Yujian, L., Bo, L.: A normalized levenshtein distance metric. IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(6), 1091–1095 (June 2007). https://doi.org/10.1109/TPAMI.2007.1078

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_8