On the Machine Learning Based Business Workflows Extracting Knowledge from Large Scale Graph Data

Mert Musaoğlu[1(✉)], Merve Bekler[1], Hüseyin Budak[1], Celal Akçelik[1], and Mehmet S. Aktas[2]

[1] R&D Department GTech, Istanbul, Turkey
{mert.musaoglu,merve.bekler,huseyin.budak,celal.akcelik}@gtech.com.tr
[2] Computer Engineering Department, Yildiz Technical University, Istanbul, Turkey
aktas@yildiz.edu.tr

Abstract. The data created by web users while navigating on a website constitutes graph data. Large-scale graph data is generated on websites many users visit with high frequency. Analyzing large-scale graph data using artificial intelligence techniques and predicting user behavior by creating models is an actively studied research topic. Within the scope of this research, a machine learning business process is proposed that will allow the interpretation of graph data obtained from web user navigation data. A prototype application was developed to demonstrate the usability of the proposed business process. The developed prototype application was run on graph data obtained from websites with intense user-system interaction. A comprehensive evaluation study was carried out on the prototype application. The results obtained from the empirical evaluation are promising and show that the proposed business process is used.

Keywords: Machine learning · Sequential pattern mining · Customer journey · Funnel analysis · Encoding

1 Introduction

A series of conversions, sales, and registrations are performed on websites that receive visitors daily. Funnel analysis is used to understand the traffic in this set of transactions. Funnel analysis is a method of understanding the steps required to reach a result within the website and how many users pass through these steps [1]. In other words, this analysis allows us to analyze the splits in the conversion path, understand which path has the splits, and interfere with these splits to impact the overall conversion rate. It is clear that improving conversion rates is essential, but funnel analysis is not just tracking conversion rates. Funnel analysis also allows us to understand how the conversion rate changes based on user characteristics or

TUBITAK supported this study under project ID# 3200698.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
O. Gervasi et al. (Eds.): ICCSA 2022 Workshops, LNCS 13381, pp. 463–475, 2022.
https://doi.org/10.1007/978-3-031-10548-7_34

behavior. Thus, it is understood which users are more likely to convert, the reasons for leaving, and what users who leave are doing. This feature improves the funnel performance and finds potential flaws in critical funnels and user flows. Depending on the complexity of a product or the question it is trying to answer, the funnel can be incredibly simple or overly complex. Therefore, it is crucial to correctly define a funnel to gain accurate insights from the analysis. Funnels, also called conversion funnels or sales funnels, are widely used in various marketing functions. They help identify the barriers that cause users to leave before reaching a conversion point. The navigation click flow data subject to the funnel analysis reveals evidence that will inform about the usability and functionality of the web application. The web crawl graph data in Fig.1(a-1) consists of navigation URLs served by corners and links. Each link between two vertices corresponding to navigation URLs refers to a single user's action, such as a visited URL [2]. These tours within site are evaluated at various stages of funnel analysis. As in the image in Fig.1(a-2) here, funnel analysis consists of various stages, including awareness, interest, interaction, action, and advocacy. One of the most effective points of funnel analysis as a marketing activity is that it gives information about which stage has the most room for improvement. This study argues that clickstream data contains an essential signal for predicting future user action. We argue that the conversion rate can be increased by estimating people who could not reach the "action" stage in funnel analysis for any reason but are likely to access it. We formulate the problem of simultaneously predicting future user actions given a user's clickstream history. To solve this new problem, we treat user click data on web pages as a classification problem by preparing it for modeling with preprocessing and embedding methods. We perform experiments on a real dataset and predict the person's next step using supervised learning-deep learning algorithms.

The structure of this work is as follows. Section 2 gives an overview of the fundamental concept, followed by literature. Section 3 describes the problem description. Our proposed methodology is explained in Sect. 4. Section 5 describes the prototype and experimental design, while Sect. 6 provides our experiments. Section 7 covers the critical conclusion related to the research summary and future research directions.

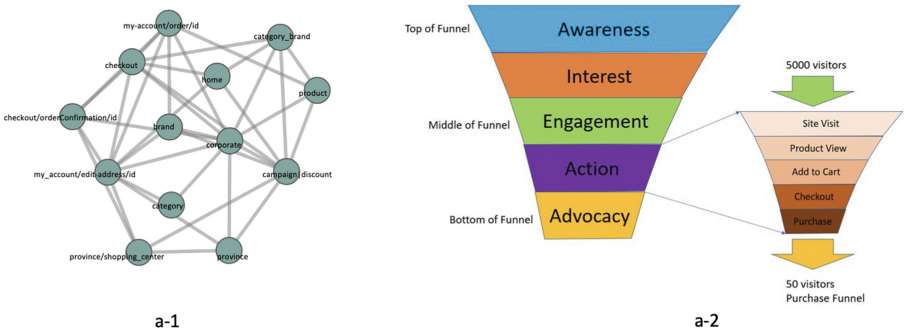

Fig. 1. a-1) Visualization of an example user browsing graph data, a-2) Conversion funnel

2 Fundamental Concept and Literature Review

2.1 Fundamental Concept

In this study, we propose an end-to-end business process that aims to predict the subsequent movements of users in the conversion path in the stages of the funnel analysis. Navigation sequence data generated from user navigations are first subjected to different data transformation applications and data pre-processing. Embedding methods are the focus of these data pre-processing processes. Although many embedding methods are used in the literature, various versions of Word2Vec, CountVectorizer, and TF-IDF with the help of N-Gram are frequently used, especially in studies on clickstream data. This study uses five different data transformations, including these three methods and their variants. After the data pre-processing process, the data is labeled according to the user movement to be predicted and made suitable for the relevant machine learning and deep learning algorithms. While training the mentioned algorithms, models are trying to predict users' subsequent movements in the conversion path. Many models are trained using different embedding methods and different estimation algorithms, and the most successful one is selected according to the relevant metric.

Embedding Methods: The textual clickstream data input to machine learning models is called word embedding. These methods provide the extraction of semantic information between words. Methods are gathered under two different representations as frequency-based and estimation-based.

TF-IDF (Term Frequency-Inverse Document Frequency) method, which is one of the frequency-based word embedding methods, is one of the frequently used embedding methods [3]. A statistical approach determines how much each word represents the document it contains. For each word, a score is calculated, indicating the value of the word in the relevant document and word dictionary (corpus) [4].

- Number of occurrences of the relevant word in the document, term frequency (TF).
- Reverse document frequency (IDF) of related word.

This score is calculated with the help of two basic metrics. Ramos observed that TF-IDF works much more successfully in documents containing information on similar topics in his study [5].

Term Frequency (TF) value is the frequency of each word in the document. It is a metric that is directly related to the length of the document. Since there is no relationship between the lengths of the documents and their order of importance, normalization is applied over the frequency value. As Yang and Korfhage [6] pointed out, the longer the documentation lengths, the better the performance of word conversions. In the next step, the documents are converted into vectors.

The vectorization process is applied over the whole word dictionary for documents. In this way, the change due to the difference between the document lengths is prevented [7] TF value takes a value between 0 and 1 according to the importance of the word in the document. If a word does not appear in the document at all, it takes the value 0, and if all the words in the document are the same word, the word's TF score will be the value 1.

TF(t,d) = number of occurrences of word t in document/number of words in document

Reverse Document Frequency (IDF) is the inverse of document density (DF); it is calculated by taking the logarithm of document density. Document frequency is calculated by dividing the number of documents by the number of documents containing the relevant word.

DF(t) = occurrence of t word in n documents

When calculating the IDF, if there are frequently repeated words such as stop words in the document, this value will receive a low score. While calculating the IDF value, one is added to the denominator to eliminate possible ambiguity, taking into account the possibility that the word will never occur in the document.

$$IDF(t) = log(N/(DF + 1))$$

Thus, the TF-IDF value can now be produced by combining the formulas from both sides.

$$TF - IDF(t, d) = TF(t, d) * log(N/(DF + 1))$$

The TF-IDF score can be used with N-Gram sequences to better measure associations. N-Gram series are defined as pieces of text that come together N times that are used together in texts or documents. The variable N here can be a character, word, or sentence [8].

CountVectorizer, another frequency-based embedding method, creates a new column in the matrix for each different word in the documents and represents the words with the help of columns. The matrix also includes other documents in its rows, each cell represents the number of times the relevant word occurs in the relevant document [9].

Word2Vec, one of the prediction-based embedding methods, is an unsupervised learning model that can detect the contexts and semantic dependencies between words, unlike other methods. As Church [10] stated in his study, Word2Vec was accepted by large groups in a short time due to its ease of use and accessibility and was used in most studies. The Word2Vec model detects the relationships between words and creates word vectors depending on them. An embedding matrix is made from these extracted vectors and given to deep learning models

as an embedding layer. Looking at the working logic, in the first step, as in the CountVectorizer, using the tokenizer creates a column in the matrix for each different word. In each document, 1 or 0 is assigned depending on whether the relevant word is included or not. One of the ways Word2Vec differs from other methods is the window size parameter. With the Window size parameter, information on how many words can be left and right of the relevant word as a bundle is given to the model [11].

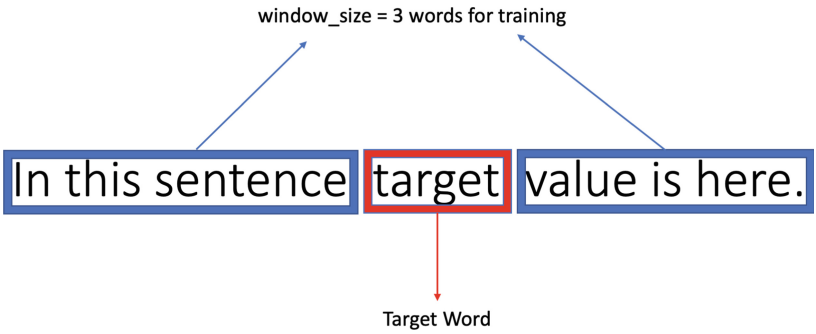

Fig. 2. Example of window size parameter

When we look at the preliminary stages of Word2Vec, Word2Vec models are divided into two structures. In models with CBOW and Skip-Gram. CBOW structure, the word in the center is selected as the target variable, and the other words in the window size are used as input. In Skip-Gram models, the related word predicts other words in window size. Words used as inputs are transferred to the modeling stage of Word2Vec as vectors. The model, which is in the artificial neural network structure, contains N neurons in the hidden layer, and the output module generates a vector of V length. The Word2Vec model does not include an activation function between the input and hidden layers. The softmax function is located between the hidden layer and the output layer. Related processes are applied to each document, and then the model is made ready by using a mapping.

2.2 Literature Review

Customers proceed on a purchasing journey to meet their needs [12]. The classic AIDA (Attention, Interest, Desire, and Action) model depicts the consumer journey as a buying funnel (Strong 1925; Howard and Sheth 1969) [13]. The journey in this funnel; includes stages such as awareness, interest, desire, and action. It also consists of a buying process in which customers' implicit purchasing tendencies can be influenced by their goals or situational factors in the shopping environment. Considering all aspects, it is critical not to lose the customer who comes

to the final stage of purchasing funnel. In this respect, we recommend a business process that supports the completion of the conversion path by finding people who are most likely to make the transition to the purchase stage. Within the proposed business process, studies were carried out using the leading websites of companies in the automotive and retail sectors. When we look at the literature, although no study performs this funnel analysis and directly deals with websites in the automotive and retail industries, some studies make sense of customer behavior through search data in many channels such as search engines, video screens, redirect engines. Moe ve Fader (2004) [14] used different types of visits to develop a model that predicts consumers' probability of purchasing, similar to the business process we propose. Moe (2003) [15] has worked on different visit goals that reflect different stages in the funnel. There exist studies that focus on the analysis of browsing graph-data for the purpose of user interface testing [2,16–20]. However, in this study, we focus on analyzing browsing graph data to understand the purchase behaviour.

In our proposed business process, clickstream data is used to determine the possibility of purchasing the next step. Traditional classification algorithms work on vectors consisting of real numbers. For this purpose, the focus will be on creating vectors that will most successfully represent dynamic URLs. Defining dynamic URLs is a current research topic in the literature. We see that Yuan et al. use the word2vec method to represent dynamic URLs as vectors (Yuan et al. 2018) [21]. In another study, Li et al. generate representation vectors by applying the word2vec method after further segmenting URLs into non-numeric characters (Li et al. 2019) [22]. These transformations are made with the general name of embedding methods. In the literature, many studies use machine learning methods to a group and make sense of user browsing behaviors involving similar click-data flows (Su 2015), (Ting 2005), (Wang 2013) (Sadagopan 2008). When the methods used to predict online shopping behavior are examined, it is seen that supervised machine learning methods and mainly deep learning are used. Kohn, Dennis Lessmann, Stefan Schaal, Markus (2020) [23] emphasizes that supervised machine learning is conceptually unsuitable due to the sequential nature of click data but that the iterative neural networks (RNNs) framework can unlock the full potential of clickstream data. In his studies, Logistic Regression, Gradient Boosting Machine, etc., from supervised learning algorithms. Furthermore, RNN comparisons were made with GRU, LSTM, which are deep learning algorithms. In the proposed business process, we use both supervised learning algorithms and deep learning algorithms compared to different embedding methods. In this way, we aim to collect the strengths of other studies in the literature. We observe studies that analyzes the graph data to extract information on the data lineage [24–27]. Different from these studies, we focus on analyzing browsing graph-data.

3 Problem Definition

The business process we recommend is intended to predict users' future actions and consider their clickstream history. For this process, customer navigation data

is divided into customer sessions needed. Various pre-processing and embedding methods have been applied to the navigation data. The new data became the input for machine learning and deep learning algorithms. As a result of the process, users who are most likely to enter the conversion path you want to go in the funnel are estimated from the users who continue to browse the site. For a better understanding, the following questions are asked.

1. What should a business process look like by modeling click-stream and showing the purchase target?
2. How can tagged sequence data be created and automatically tagged in the business process that models customer behavior?
3. What are the most appropriate embedding methods to model customer behavior?
4. Which machine learning–deep learning models give the best results for predicting sequences?

4 Proposed Methodology

The proposed business process is modularly designed. The end-to-end module-based flow from the data collection stage to the estimation results is illustrated in Fig. 3. At the same time, Fig. 3 image is the first of the research questions, "What should a business process look like by modeling click-stream and showing the purchase target?" answers the question.

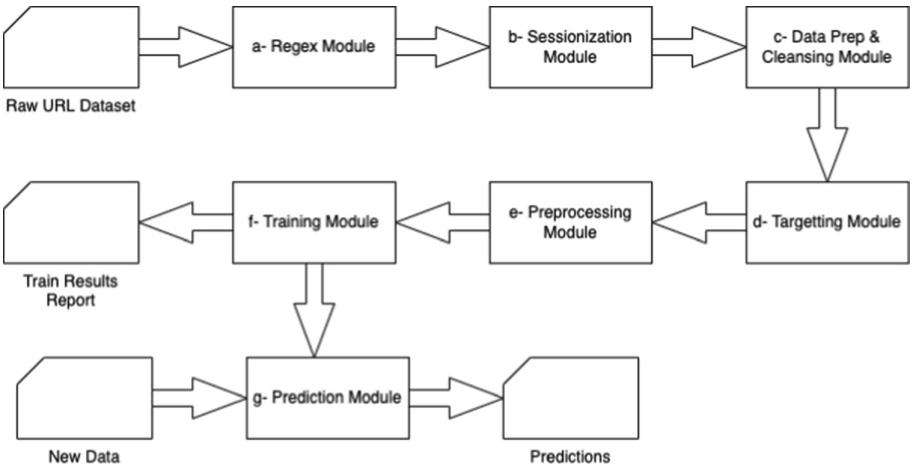

Fig. 3. Proposed supervised machine learning based business work flow

Modules a, b, c, d, e in the flow cover the data flow process to get the data ready for modelling. Other modules e and f include modeling and estimation studies.

The collected data set is passed to the first module, **a-regex module**, in order to mark predefined topics. Each page URL is grouped and tagged according to which topic-specific navigation it contains using the relevant regex. After the tagged URLs, they move to **b-sessionization module** and are grouped here as SessionID, Page URL, Time. At the same time, observations with missing data were excluded from the analysis. In **c-data prep & cleansing module**, the number of crawls in the same session is limited. The goal here is to get people's real browsing. In this restriction, if there is a long wait at the relevant URL, the session is divided into two. Afterward, the number of navigation steps is limited to 10. The reason behind this action is to avoid data loss. The last ten tours of the sessions with more than ten crawls are included in the analysis. Single trips in the same session were removed from the analysis. As a result of these sequential modules, a cleansed sequence dataset was obtained. In order to predict the next step, the last movements of the cleaned session-based data obtained are reserved as the target variable. With this study, we have answered the second of the research questions. In this study, which is included in **d-targetting module**, as seen in Fig. 4, label encoding is made for the selected target, and a 0-1 assignment is made. After the assignment, the variable set that will enter the training is inserted into the embedding methods to make the text data mathematical in the **e-preprocessing module** in order to make it suitable for modeling.

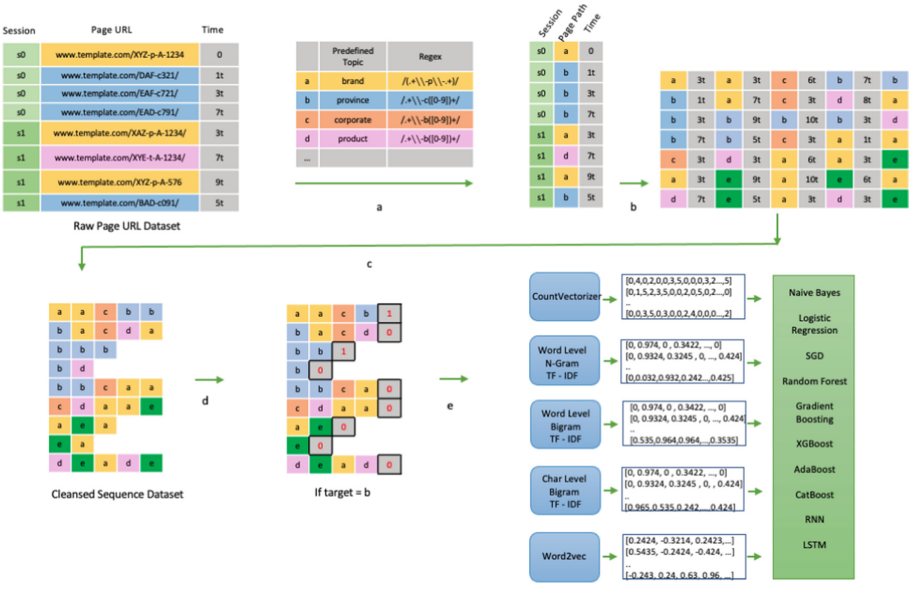

Fig. 4. Data flow

In this module, datasets are created using five (CountVectorizer, Word Level N-Gram TF-IDF, Word Level Bi-Gram TF-IDF, Char Level Bi-Gram TF-IDF, Word2Vec) embedding methods. As a result of this process, the data sets are ready for modeling. In **f-training module**, then different algorithms are trained. Eight algorithms are machine learning algorithms (Naive Bayes, Logistic Regression, Stochastic Gradient Descent, Random Forest, Gradient Boosting, XGBoost, AdaBoost, CatBoost), and two algorithms are deep learning algorithms (RNN, LSTM). Analyzes were carried out using different algorithms. In the proposed end-to-end process, different embedding methods and algorithms were compared. A design was made by finding the embedding method and estimation algorithm that gave the best results according to the metrics selected for the problem. These comparisons are recorded in reports based on success metrics and working times. The estimation of the new incoming user behavior flow is performed in **g-prediction module** with the algorithm and embedding method that gives the best results based on metrics.

5 Prototype and Experimental Study

5.1 Prototype

The modules described in the methodology section are implemented using various Python-3.9 libraries. While collecting navigation data on the website, javascript scripts were used. In the modeling and preprocessing part, libraries such as sklearn (1.0.2), TensorFlow (2.7.0), Keras (2.7.0), gensim (4.1.2), catboost (1.0.3), xgboost (1.5.1) were used. MongoDB was used for saving datasets.

5.2 Experimental Design

Within the scope of this study, a raw data e-commerce site data set of approximately 6 million lines have been studied. On this raw data, the pre-processes in modules a, b, c, d, and e in the modular business process described in the methodology section were completed, and approximately 160 thousand rows of data entered the modeling. It has been tried to find the probability that the next step of the people browsing the e-commerce site will be "sales." 70% of the dataset is reserved for training during the modeling phase, while 30% is reserved for testing. The training set, approximately 70 thousand, was divided into 5 folds by k-fold cross-validation. The success rate of the embedding methods and algorithms according to the error metrics and working times followed in the study is recorded as a table. The error metrics used in the study are as follows, respectively:

- Accuracy
- Precision
- Recall
- F1 Score
- Roc-Auc.

In the study, the embedding method, algorithm, target, and the desired fold number in the k-fold cross-validation are written dynamically, allowing the user to change it.

6 Experiments

As a result of the proposed work process, success scores were obtained based on the metrics explained in the experimental design section. As seen in Fig. 5, when looking at the F1 score values, it is seen that the TF-IDF embedding method is better at the character level, while XGBoost is in the first place as an algorithm, followed by CatBoost. Naive Bayes algorithm and CountVectorizer embedding method gave the lowest results. These results will vary depending on the target variable and selected data set. In addition, 3 and 4 of the research questions were answered together with Fig. 5.

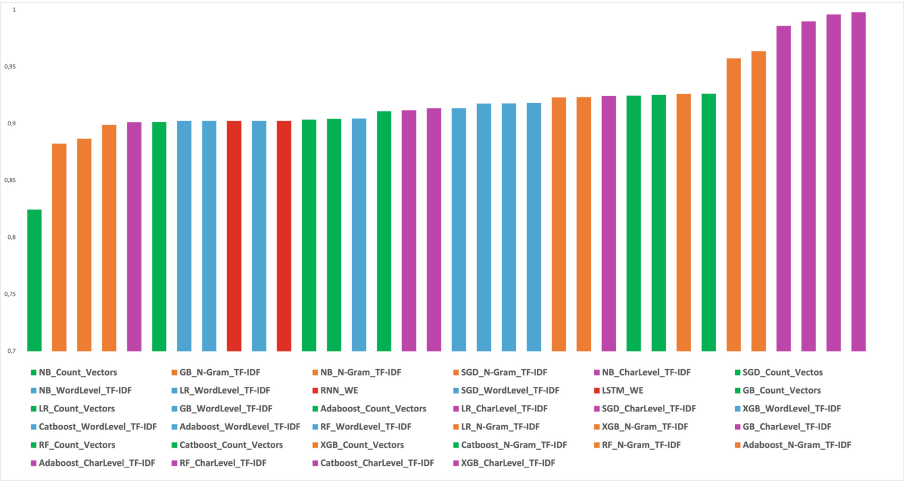

Fig. 5. F1-score results

When we look at the accuracy results in Fig. 6, it is seen that the TF-IDF embedding method is better at the character level in parallel with the F1 score, and tree-based models have higher success as an algorithm.

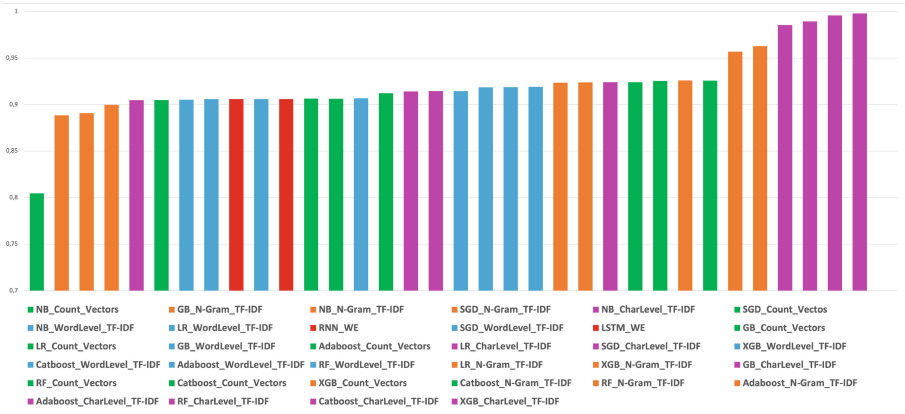

Fig. 6. Accuracy results

7 Conclusions and Future Research Directions

With the proposed business process, the website navigation data have been used to find which topic will be the next move in people's navigation. Different embedding methods and machine learning/deep learning algorithms represent textual data with mathematical expressions. As a result of the study, different success metrics and methods were compared. As a result, it has been seen that the character-level TF-IDF method is more successful in this data set, and tree-based algorithms are much better.

We carried out our studies on the data set we obtained from an e-commerce site in this study. We aim to expand the use of this study in other e-commerce sites. We focused on a study aiming at purchasing within the scope of the business process, and we will implement this study for different purposes in other studies. In addition, it is another aim of ours to increase the embedding methods we use in future studies.

References

1. Daugherty, T., Djuric, V., Li, H., Leckenby, J.: Establishing a paradigm: a systematic analysis of interactive advertising research. J. Interact. Adv. **17**(1), 65–78 (2017)
2. Olmezogullari, E., Aktas, M.: Representation of click-stream DataSequences for learning user navigational behavior by using embeddings. In: 2020 IEEE International Conference on Big Data (Big Data), pp. 3173–3179 (2020). https://doi.org/10.1109/BigData50022.2020.9378437
3. Aizawa, A.: An information-theoretic perspective of TF-IDF measures. Inf. Process. Manag. **39**(1), 45–65 (2003)
4. Zhang, W., Yoshida, T., Tang, X.: A comparative study of TF-IDF, LSI and multi-words for text classification. Exp. Syst. Appl. **38**(3), 2758–2765 (2011)

5. Ramos, J.: Using TF-IDF to determine word relevance in document queries, Technical report, Department of Computer Science, Rutgers University (2003)
6. Kusner, M., Sun, Y., Kolkin, N., Weinberger, K.: From word embeddings to document distances. In: Proceedings of the 32nd International Conference on Machine Learning, pp. 957–966 (2015)
7. Yang, J., Korfhage, R., Rasmussen, E.: Query improvement in information retrieval using genetic algorithms-a report on the experiments of the TREC project. In: Proceedings of the 1st Text Retrieval Conference (TREC-1), pp. 31–58 (1992)
8. Atanu, D., Mamata, J., Thakkar Jitesh, J.: Lexical TF-IDF: an n-gram feature space for cross-domain classification of sentiment reviews. In: International Conference on Pattern Recognition and Machine Intelligence, pp. 380–386 (2007)
9. Rodríguez, P., Bautista, M.A., Gonzàlez, J., Escalera, S.: Beyond one-hot encoding: lower dimensional target embedding. Image Vis. Comput **75**, 21–31 (2018)
10. Ward, C.K.: Word2Vec. Nat. Lang. Eng. **23**, 155–162 (2017)
11. Rong, X.: Word2vec Parameter Learning Explained, arXiv preprint (2014). arXiv:1411.2738
12. Li, A. Ma, L.: Charting the path to purchase using topic models. J. Mark. Res. **57**(6), 1019–1036 (2020)
13. Howard, J.A., Sheth, J.N.: The Theory of Buyer Behavior, pp. 83–114. John Wiley Sons, New York (1969)
14. Moe, W., Fader, P.: Dynamic conversion behavior at E-commerce sites. Manag. Sci. **50**, 326–335 (2004). https://doi.org/10.1287/mnsc.1040.0153
15. Moe, W.W.: Buying, searching, or browsing: differentiating between online shoppers using in-store navigational clickstream. J. Cons. Psychol.. **13**(1–2), 29–39 (2003)
16. Erdem, I., Oguz, R.F., Olmezogullari, E., Aktas, M.S.: Test script generation based on hidden Markov models learning from user browsing behaviors. In: 2021 IEEE International Conference on Big Data (Big Data), pp. 2998–3005. IEEE (2021)
17. Oz, M., Kaya, C., Olmezogullari, E., Aktas, M.S.: On the use of generative deep learning approaches for generating hidden test scripts. Int. J. Softw. Eng. Knowl. Eng. **31**(10), 1447–1468 (2021)
18. Oguz, R.F., Oz, M., Olmezogullari, E., Aktas, M.S.: Extracting information from large scale graph data: case study on automated UI testing. In: Euro-Par 2021: Parallel Processing Workshops. Euro-Par 2021, LNCS, vol. 13098. Springer, Cham (2021). https://doi.org/10.1007/978-3-031-06156-1_29
19. Olmezogullari, E., Aktas, M.S.: Pattern2Vec: representation of clickstream data sequences for learning user navigational behavior. Concurr. Comput. Pract. Exp. **34**(9) (2022)
20. Uygun, Y., Oguz, R.F., Olmezogullari, E., Aktas, M.S.: On the large-scale graph data processing for user interface testing in big data science projects. In: 2020 IEEE International Conference on Big Data (Big Data), pp. 2049–2056 (2020)
21. Yuan, H., Yang, Z., Chen, X., Li, Y., Liu, W.: URL2Vec: URL modeling with character embeddings for fast and accurate phishing website detection. In: 2018 IEEE International Conference on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom) (2018)
22. Li, B., Yuan, G., Shen, L., Zhang, R., Yao, Y.: Incorporating URL embedding into ensemble clustering to detect web anomalies. Fut. Gener. Comput. Syst. **96**, 176–184 (2019)

23. Köhn, D.L., Stefan Schaal, M.: Predicting online shopping behaviour from clickstream data using deep learning. Exp. Syst. Appl. **150**, 113342 (2020). https://doi.org/10.1016/j.eswa.2020.113342
24. Tufek, A., Gurbuz, A., Ekuklu, O.F., Aktas, M.S.: Provenance collection platform for the weather research and forecasting model. In: 2018 14th International Conference on Semantics, Knowledge and Grids (SKG 2018) (2018)
25. Baeth, M.J., Aktas, M.S.: Detecting misinformation in social networks using provenance data. In: 2017 13th International Conference on Semantics, Knowledge and Grids (SKG 2017) (2017)
26. Yazici, I.M., Karabulut, E., Aktas, M.S.: A data provenance visualization approach. In: 2018 14th International Conference on Semantics, Knowledge and Grids (SKG 2018) (2018)
27. Riveni, M., Baeth, M, Aktas, M.S., Dustdar, S.: Provenance in social computing: a case study. In: 2017 13th International Conference on Semantics, Knowledge and Grids (SKG 2017) (2017)