Annals of Operations Research (2024) 339:329–348
https://doi.org/10.1007/s10479-022-04834-w

ORIGINAL RESEARCH

Identifying purchase intention through deep learning: analyzing the Q&D text of an E-Commerce platform

Jing Ma[1] · **Xiaoyu Guo**[1] · **Xufeng Zhao**[1,2]

Accepted: 10 June 2022 / Published online: 1 July 2022

Abstract
Identifying purchase intention by analyzing the Query and the Document of the product description (Q&D) text is one of the most important means of promoting Purchase Rate (PR). In view of that customers sometimes cannot describe their purchasing intention in queries, this paper aims to identify purchase intention from implicit queries by computing semantic similarity between Q&D and proposes a novel model based on Word2Vec algorithm, Long Short-term Memory (LSTM) and Deep Structured Semantic Model (DSSM). Besides, an empirical analysis is conducted through the Keras framework and based on the factual retrieval data of the Home Depot, an E-commerce website selling building materials in America. The results show that the proposed model has achieved improving F1-score on test dataset compared with other existing models. The novel model combines Word2Vec and LSTM to extract text features and applies DSSM to further fetch high-dimension representations by maximizing semantic similarity between the user query and the description of the correct merchandise. Our proposed model can be used to remove or minimize subjective factors in extracting features, improves the performance of purchasing intention identification, and also improves the customer experience of online shopping.

Keywords Intention identification · Long Short-term Memory (LSTM) · Deep structured semantic model(DSSM) · Deep learning

✉ Xufeng Zhao
zx.peak@outlook.com

Jing Ma
majing5525@126.com

Xiaoyu Guo
xiaoyu.guo@nuaa.edu.cn

1 College of Economics and Management, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, Jiangsu, China

2 College of Mechanical and Electrical Engineering, Wenzhou University, Wenzhou 325035, Zhejiang, China

1 Introduction

Identifying purchase intention accurately can promote PR in the world of E-Commerce. According to a study by Kwek et al. (1970), web-retailers will be able to develop effective and efficient web-shopping operations to attract new and potential customers with a good understanding of the web shopper's online purchase intention. Customers depend on search results provided by E-Commerce platform to find their desirable products. Products with higher semantic similarity top the research results on E-Commerce platform like Taobao and Home Depot, which essentially improves the chance of making a deal. Customers may switch to another platform if they cannot find the product they want in the first page of searching results.

Balakrishnan and Dwivedi (2021) note that artificial intelligence(AI) enhances purchase intention through digital assistants. However, the very first step for enhancing purchase intention is to identify the query intention (Luo et al., 2018). As E-Commerce has gained scale, large quantities of goods are sold on E-Commerce platforms and it is harder for consumers to search for their desirable products than before. Additionally, one inevitable tendency, which is different from earlier online shopping experiences, is that customers will utilize natural language instead of key words when searching for the products they want to buy (Luo et al., 2018), which makes it harder to identify purchase intention from the user query. Even worse, in some vertical E-Commerce platforms, such as the building materials online shop, the name of a product is often uncommon and customers have no idea of what is the name of the product they want to buy. Therefore, they type what they want to do in the query box or ask the assistant for help. If the E-Commerce platform cannot identify purchase intention timely, it would produce a bad effect on shopping experience and thus easily lost the customers. Therefore, E-Commerce platforms intend to enhance their customers' shopping experience by addressing the problem of identifying purchase intention (Qian et al., 2017; Peng, 2018).

As the number of data is growing day by day, a number of studies explore big data driven and machine learning methods to improve the level and efficiency in management. For example, in 2018, Kumar et al. (2018) develop a big data analytics framework, which optimizes the maintenance schedule, improves the performance of remaining life prediction and leads to reduction in maintenance cost. In 2020, Kumar et al. (2020) design a big data driven framework to improve the accuracy of demand forecasts. In 2021, Qayyum et al. (2021) propose a depth-wise dense network to identify the COVID infected lungs X-rays effectively. In the same year, Sengupta et al. (2021) examine the predictors of successful Airbnb bookings with a machine learning-based variable-importance scheme and design customized recommendations for P2P accommodation platforms. In order to identify purchase intention accurately, a number of studies also have been done with big data driven and machine learning methods. User's purchase intention includes explicit intention and implicit intention (Fu & Liu, 2016). Explicit intention means that the intention is included in the query text while implicit intention is the intention that is not expressed externally but having in one's mind (Lee et al., 2015). For explicit intention identification, Luo et al. (2018) proposed a neural network model with bidirectional Long Short-term Memory (bi-LSTM) and attention mechanism, aiming to find the semantic similarity between natural language context words and central intention term, and Liao et al. (2020) build an intention recognizing model to obtain intentions of E-Commerce consumers and provide a better shopping experience for users. For implicit intention identification, to be studied in this paper, the present procedures of finding the purchase intention from ambiguous search queries are: (1) save the users' daily records, including behaviors of searching, clicking and purchasing; (2) check the purchasing records after searching; (3) build a map between the query and the commodity bought by the consumer. If the problem of ambiguous query occurs, E-Commerce platform would conclude a purchase intention and list commodities according to the established mapping relation. Based on these rules, Lee et al. (2015) conduct an experimental paradigm with collecting and analyzing the eye tracking data and response time, building a standard to recognize implicit shopping intention. Fu and Liu (2016) regard implicit consumption intention recognition as a multi-label classification task, which combines multiple features based on follower's behavior, intention behavior, retweets behavior, and user profiles. Jia et al. (2020) analyze a large number of text data with purchasing intention published by Weibo users. Fu and Liu (2016) and Jia et al. (2020) identify implicit intention on social media platform rather than text of user query, which is the problem we are intended to address. Li et al. (2017) notice that the user query also contains implicit shopping intention and leverage the Encoder-Decoder model to "translate" the implicit intention into the corresponding explicit intention by using the parallel corpora built on the social data.

Fewer research puts effort on mining user's query to find out the implicit intention, but implicit intention is very common on E-Commerce platform, especially in vertical E-Commerce platform. Thus, the main objective of this paper is to devise a model to identify purchase intention from user's query based on semantic similarity computation, inspired by information retrieval task. The key technology to calculate semantic similarity is deep learning, which has achieved the great performance in most Natural Language Processing (NLP) tasks (e.g., (Devlin et al., 2018)). As we devised, our model would utilize deep learning methods to predict the products, which possess the closest similarity with user's query.

To identify the purchase intention from natural language queries, we propose a novel Intention Identification Method based on Word2Vec, LSTM and DSSM (WL-DSSM). In our research, we organically integrate Word2Vec, LSTM, and DSSM, which is built on top of the good performance of Word2Vec-LSTM in natural language processing (e.g., (Xiao et al., 2018)) and LSTM-DSSM in information retrieval (Palangi et al., 2014). First, we use methods of natural language processing to pre-process the data. After pre-processing, we train the unique Word2Vec model using the cleaned corpus to give a semantic representation for each word. Finally, we use LSTM-DSSM for further extraction of text features.

The rest of this paper is organized as follows: In Sect. 2 we discuss related works. In Sect. 3 we provide the details of the proposed WL-DSSM method. In Sect. 4 we present the experimental process and results on assessing the performance of WL-DSSM and discuss their implications. And we conclude the paper in the last section.

2 Related works

Purchase intention identification mainly includes works of two different areas, which include semantic representation of text (including word-level and sentence-level representation) and semantic similarity calculation.

2.1 Semantic representation of text

2.1.1 Semantic representation of words based on one-hot and Word2Vec

One-hot encoding is often used to convert labels into a matrix to make sure the distances within labels are same (Mai & Le, 2020). Also, one-hot encoding transformation is used to convert categorical features into continuous and Boolean dummy variables with 0 or 1 for each of their values (Liao et al., 2020; Tchuente & Nyawa, 2021). Actually, one-hot encoding is also an easy way to transform natural language to matrix or array, which can be processed by computer directly. However, the word vectors represented by one-hot encoding are independent and the semantic meaning cannot be mapped. Additionally, the sparse and high-dimensional vectors would cause curse of dimensionality easily when the size of dictionary is big enough (Seger, 2018; Rodríguez et al., 2018). On the basis of one-hot encoding, Mikolov et al. (2013) proposed Word2Vec models to embed words into vectors space semantically. Furthermore, Mikolov et al. (2013) and Rong (2014) illustrated the details about Word2Vec, including the training procedures. On the one hand, Word2Vec could produce word vectors semantically with a lower dimension and thus improves one-hot encoding. On the other hand, Word2Vec could transform a word into a vector more efficiently than tremendous pre-training language models, such as Bidirectional Encoder Representation from Transformers (BERT) (Devlin et al., 2018), Generative Pre-Training (GPT) (Radford et al., 2018) and Xlnet (Yang et al., 2019), which all have millions of parameters to optimize. Consequently, Word2Vec is more suitable for real-time feedback platforms because of the efficiency.

2.1.2 Long short-term memory for sentence representation

Recently, deep learning technology has been applied in more and more fields, especially in the field of text processing (e.g., (Kumar et al., 2020)). Convolution Neural Network (CNN) could learn text semantics through convolution extraction of text features and has a certain capability of anti-noise (Zhou et al., 2017). In the real world, sequence is vital to natural language. For example, “look after” and “after looking” have different meaning and CNN cannot tell any differences from each other. That is to say, CNN did not take sequence factor into account. In order to solve this problem, Mikolov et al. (2010) proposed Recurrent Neural Network Language Model (RNNLM) to give a better representation for sentence data. Recurrent neural network (RNN) introduces constant circulation into its model so that it could process sequence information. RNN shows remarkable performance in processing many tasks concerning sequence including tasks of NLP (e.g., (Sun et al., 2021)), but RNN has a Long-Term Dependencies problem (Bengio et al., 1993) when it processes longer texts. The LSTM architecture, proposed by Hochreiter and Schmidhuber (1997) , addresses this problem of Long-Term Dependencies by introducing a memory cell that is able to store state information over long period of time into RNN.

Due to its great performance, LSTM has recently been used in information retrieval field for extracting sentence-level semantic vectors (Palangi et al., 2016) and context-aware query suggestion (Sordoni et al., 2015). Additionally, Eachempati et al. (2021) apply deep neural networks with LSTM to capture the sentiment from disclosure information aiming to assess the asset prices’ impact and Kumar et al. (2022) regard the LSTM as a base classifier to detect fraudulent review.

2.2 Semantic similarity calculation

Generally, cosine function is widely applied to calculate the value of similarity (e.g. (Kumar et al., 2022)). However, semantic similarity calculation depends on semantic representation of text to a great extent. Xia et al. (2020) use bi-LSTM to get the sentence vector and then calculate the similarity between the two vectors of question and answer by cosine function to decide whether the answer matches the question or not. An et al. (2016) propose a deep learning model based on LSTM, aiming to calculate the semantic similarity between questions in question-answering system, with State Of The Art(SOTA) performance under the situation without external information resources. Also, Nassif et al. (2016) build a neural network model based on LSTM in order to calculate the semantic similarity between questions. The two models proposed by An et al. (2016) and Nassif et al. (2016) only adapt to the situation of short text, but there are both short text and long text when shopping at E-Commerce platforms, with query in the form of short text and product description in the form of long text. Das et al. (2016) apply Siamese network (Chopra et al., 2005) to calculate semantic similarity and to search for similar questions. Siamese network concatenates two same networks, sharing parameters with each other. As a result, Siamese network only works well for sentence pairs which have similar text length. However, this paper focuses on calculating semantic similarity between the query and the product description, which is different from calculating similarity between questions. Because the text lengths of questions are similar while the text lengths of query and product description have a big gap, it is hard to get a good performance for the problem to be solved in this study with Siamese network theoretically.

Additionally, some models use semantic similarity to further improve the performance of text representation. Deep Structured Semantic Models (DSSM), mapping queries and documents into the same semantic space under the constraint condition of cosine similarity, are proposed for further feature extracting in the field of information retrieve (Huang et al., 2013). Shen et al. (2014) and Palangi et al. (2014) add CNN and LSTM into DSSM, namely CNN-DSSM and LSTM-DSSM respectively, and improve the performance of retrieve results. DSSM, CNN-DSSM and LSTM-DSSM use the method of Word Hashing to get the word vector representation, which cannot represent a word semantically.

Inspired by the existing literature, this paper mines and analyzes Q&D text data from the E-Commerce platform using WL-DSSM, which possesses advanced text processing, deep learning, and artificial intelligence technology. Based on the great performance of DSSM in information retrieval task, we introduce Word2Vec and LSTM at the same time in order to extract word-level and sentence-level feature semantically. Our research attempts to mine implicit purchase intention from search queries, to provide more intelligent service for consumers in online shopping.

3 Methodology

Aiming at identifying purchase intention through deep learning via analyzing the Q&D text on an E-Commerce platform (Home Depot), this paper proposes a purchase identification model based on Word2Vec, LSTM and DSSM. We show the framework of the WL-DSSM method in a schematic diagram in Fig. 1.

First, we pre-process the corpus via natural language processing tools and train the Word2Vec model using the processed corpus. Then, we get words vectors of Q and D from Word2Vec model after segmenting. Finally, we send vectors to LSTM sequentially and extract high-level feature via LSTM-DSSM under the constraint condition of cosine function.

3.1 Definition of the problem and notations

In order to identify purchase intention, we calculate the similarity score between Q&D. Therefore, our model is defined as:

$$S(Q, D) \in [0, 1] \tag{1}$$

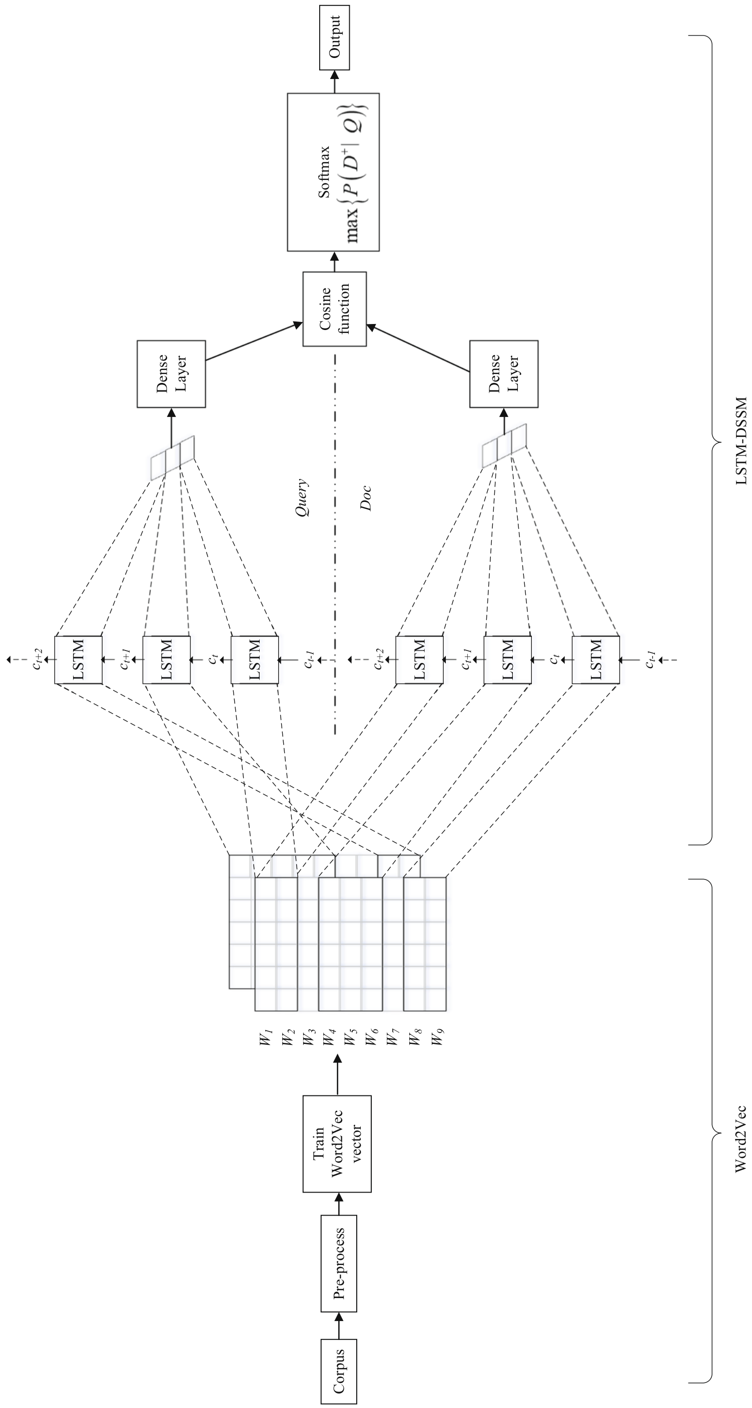

Fig. 1 The framework of WL-DSSM

where S indicates the semantic similarity between Q&D, S is a real number between 0 and 1, Q denotes a user query, and D denotes a document of the product description. Q and D are the inputs of the model and S is the output of the model. After calculating all the S, we select a maximum one and pick up the corresponding D as the intention product.

Additionally, let ϕ be a dictionary and V be the number of words in ϕ. Q_{vec} and D_{vec} indicate the vector representation of Q and D, respectively. D^+ is the intention of Q while D^- is not. Accordingly, D^+_{vec} and D^-_{vec} denote the vector representation of D^+ and D^-, respectively. Let d be the dimension of word vector, W_i be the word i, and W^i_{vec} be the vector representation of W_i. mql denotes maximum length of Q, mdl denotes maximum length of D, and c denotes the size of window. O^i is the one-hot representation of word i and ω is the current word.

3.2 Word vectorization based on Word2Vec

We first need to map words into semantic vectors so that the semantic similarity can be calculated. DSSM uses the method of Word Hashing to transform words into arrays. Word Hashing could reduce dimensionality, but it would cause that the words which have same spelling but different meaning would have the same n-gram representation (Huang et al., 2013). In this paper, inspired by the great performance of Word2Vec (Mai & Le, 2020; Xia et al., 2020), we train Word2Vec vectors using CBOW (Continuous Bag-of-Words) model.

The input layer of CBOW is the one-hot encodings of the words before and after target word ω. As is shown in Fig. 2, take sentence “Radiant Barrier is easy to use and install” as an example. We transverse each word in the sentence and regard it as the to be predicted word ω. If $\omega = is$ and $c = 2$, then the words before ω are $W^1 = Radiant$ and $W^2 = Barrier$; the words after ω are $W^4 = easy$ and $W^5 = to$. Therefore, $Context(\omega) = \{W^1, W^2, W^4, W^5\}$. Furthermore, take the one-hot encoding O^1, O^2, O^4, O^5, which have the same dimension V, as the input of CBOW.

The output of CBOW is a Huffman Tree, the leaf node of which is created by words appeared in the corpus and the weight of which is defined by the frequency of words. As the conventional customs of CBOW, if the node is classified to the left, this node belongs to the negative class and represents with 1; otherwise belongs to the positive one and represents with 0. Let θ be the undetermined coefficient. According to the equation of sigmoid function, the probability of classifying a node to the positive class is

$$\sigma\left(X_\omega^T\theta\right) = \frac{1}{1 + \mathrm{e}^{-\mathbf{X}_\omega^T\theta}}. \tag{2}$$

Therefore, the probability of classifying a node to the negative class is $1 - \sigma\left(X_\omega^T\theta\right)$. Let the $j - 1^{th}$ classification result be $p\left(d_j^\omega \mid \boldsymbol{X}_\omega, \theta_{j-1}^\omega\right)$, and

$$p\left(d_j^\omega \mid \mathbf{X}_\omega, \theta_{j-1}^\omega\right) = \left[1 - \sigma\left(X_\omega^T\theta_{j-1}^\omega\right)\right]^{d_j^\omega}\left[\sigma\left(X_\omega^T\theta_{j-1}^\omega\right)\right]^{1-d_j^\omega}. \tag{3}$$

For each word, there is only a path from root to node ω in Huffman Tree, with $l^\omega - 1$ times binary classification totally. The probability of classifying correctly is

$$p(\omega \mid \text{ Context } (\omega)) = \prod_{j=2}^{l^w} p\left(d_j^\omega \mid \boldsymbol{X}_\omega, \theta_{j-1}^\omega\right). \tag{4}$$

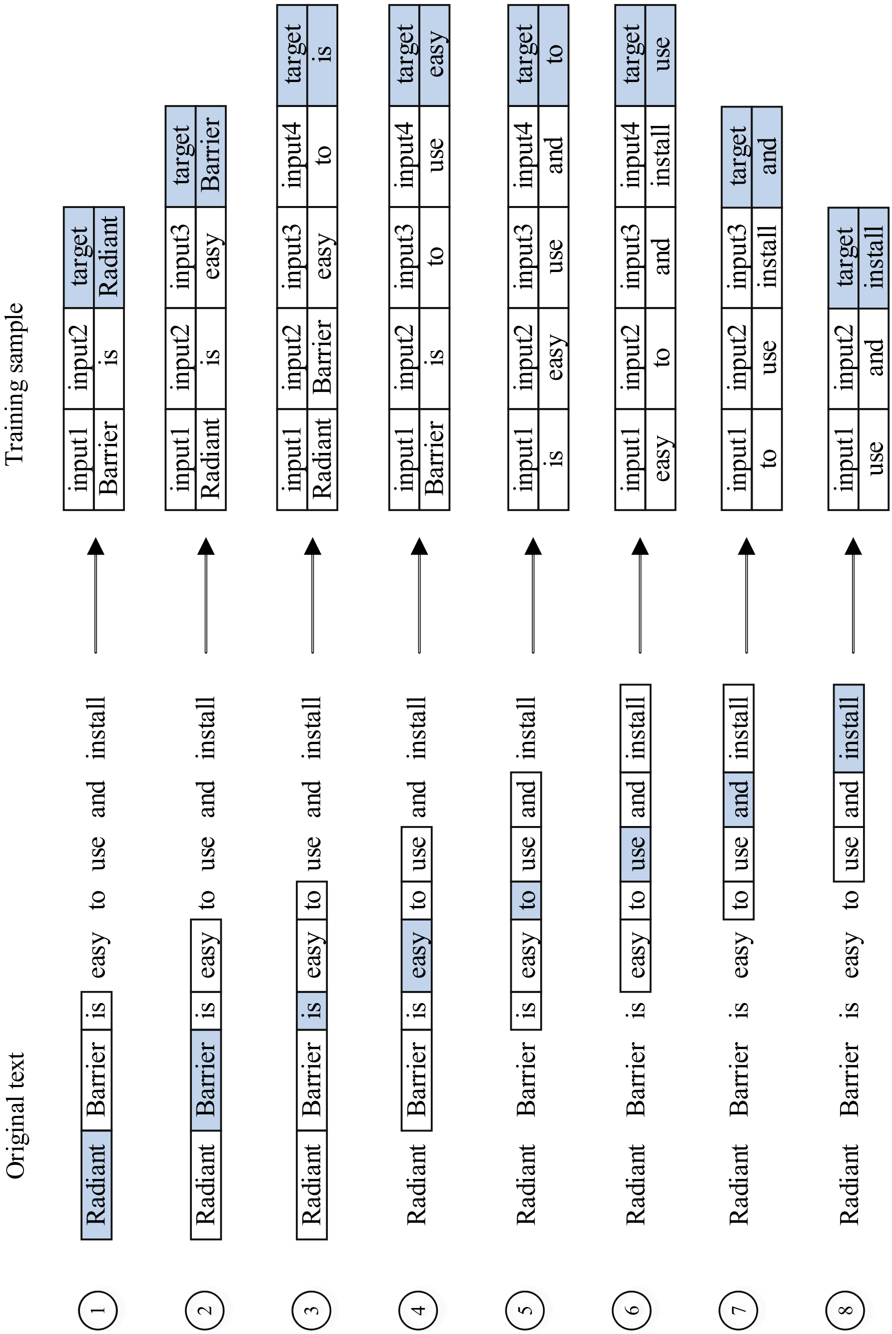

Fig. 2 Example of CBOW input

Consequently, the probability of every word in corpus ζ classified correctly is

$$P = \sum_{\omega \in \zeta} p(\omega \mid \text{ Context } (\omega)). \tag{5}$$

The objective function of CBOW is to maximize P. For this reason, take the logarithm of $p(\omega \mid \text{Context } (\omega))$ as the following equation:

$$\begin{aligned} L &= \sum_{w_t \in \zeta} \log p(\omega \mid \text{Context}(\omega)) \\ &= \sum_{w_t \in \zeta} \log \prod_{j=2}^{l^w} p\left(d_j^\omega \mid \mathbf{X}_\omega, \theta_{j-1}^\omega\right) \\ &= \sum_{w_i \in \zeta} \log \prod_{j=2}^{l^w} \left[1 - \sigma\left(X_\omega^T \theta_{j-1}^\omega\right)\right]^{d_j^\omega} \left[\sigma\left(X_\omega^T \theta_{j-1}^\omega\right)\right]^{1-d_j^\omega} \\ &= \sum_{w_i \in \zeta} \sum_{j=2}^{l^w} \left\{\log \left[1 - \sigma\left(X_\omega^T \theta_{j-1}^\omega\right)\right]^{d_j^\omega} \left[\sigma\left(X_\omega^T \theta_{j-1}^\omega\right)\right]^{1-d_j^\omega}\right\} \\ &= \sum_{w_i \in \zeta} \sum_{j=2}^{l^\omega} \left\{d_j^\omega \cdot \log\left[1 - \sigma\left(X_\omega^T \theta_{j-1}^\omega\right)\right] + \left(1 - d_j^\omega\right) \cdot \log\left[\sigma\left(X_\omega^T \theta_{j-1}^\omega\right)\right]\right\}. \end{aligned} \tag{6}$$

Write $L(\omega, j) = d_j^\omega \cdot \log\left[1 - \sigma\left(X_\omega^T \theta_{j-1}^\omega\right)\right] + \left(1 - d_j^\omega\right) \cdot \log\left[\sigma\left(X_\omega^T \theta_{j-1}^\omega\right)\right]$ as $L(\omega, j)$. Then the objective becomes to maximize $L(\omega, j)$, which has two independent variables θ_{j-1}^ω and X_ω. Take partial derivatives for the two variables sequentially and we could get

$$\begin{aligned} \frac{\partial L(\omega, j)}{\partial \theta_{j-1}^\omega} &= \frac{\partial}{\partial \theta_{j-1}^\omega} \left\{d_j^\omega \cdot \log\left[1 - \sigma\left(X_\omega^T \theta_{j-1}^\omega\right)\right] + \left(1 - d_j^\omega\right) \cdot \log\left[\sigma\left(X_\omega^T \theta_{j-1}^\omega\right)\right]\right\} \\ &= d_j^\omega \cdot \sigma\left(\mathbf{X}_\omega^\mathrm{T} \theta_{j-1}^\omega\right) \mathbf{X}_\omega - \left(1 - d_j^\omega\right) \cdot \left[1 - \sigma\left(X_\omega^T \theta_{j-1}^\omega\right)\right] \mathbf{X}_\omega \\ &= \left[1 - d_j^\omega - \sigma\left(\mathbf{X}_\omega^\mathrm{T} \theta_{j-1}^\omega\right)\right] \mathbf{X}_\omega, \end{aligned} \tag{7}$$

and

$$\frac{\partial L(\omega, j)}{\partial \mathbf{X}_\omega} = \left[1 - d_j^\omega - \sigma\left(\mathbf{X}_\omega^\mathrm{T} \theta_{j-1}^\omega\right)\right] \theta_{j-1}^\omega. \tag{8}$$

Update θ_{j-1}^ω and W_{vec} as following equation:

$$\theta_{j-1}^\omega := \theta_{j-1}^\omega + \eta \frac{\partial L(\omega, j)}{\partial \theta_{j-1}^\omega}, \tag{9}$$

$$W_{\text{vec}} := W_{\text{vec}} + \eta \sum_{j=2}^{l^\omega} \frac{\partial L(\omega, j)}{\partial \theta_{j-1}^\omega} \quad W_{\text{vec}} \in \text{Context}(\omega). \tag{10}$$

Stop the procedure of iteration when gradient becomes very small and we finally get the representation for every word in corpus ζ. Therefore, we transform Q&D into word vectors and add them to arrays l_Q and l_D, the dimensions of which are $s \cdot d \cdot mql$ and $s \cdot d \cdot mdl$, respectively.

3.3 High-level feature extraction based on LSTM-DSSM

Comparing with machine learning, deep learning extracts features more effectively and conveniently and without human interaction. As a classical model of information retrieve, DSSM could extract semantic features from texts of Q&D effectively (Huang et al., 2013). In this paper, we apply DSSM to identify purchase intention through constructing LSTM-DSSM based on texts of Q&D. We input the vectorized Q&D into LSTM-DSSM, train the deep learning network, and evaluate it.

Put the t^{th} word representation W_{vec}^{t} into LSTM layer, which is written as:

$$h_t = \overrightarrow{LSTM}\left(W_{vec}^{t}\right). \tag{11}$$

After sending all vectors into the LSTM layer, we get the representation h for each Q and D. Then connect h to a dense layer with $tanh$ activation function and get the semantic representation y, which is written as:

$$y = \tanh\left(W_s \cdot h + b_s\right), \tag{12}$$

where W_s is the semantic projection matrix and b_s is the bias matrix.

Through the above steps, we could get the semantic representation of Q and D, which are denoted as y_Q and y_D. Next, calculate the similarity score between y_Q and y_D, which is written as:

$$R(Q, D) = \text{cosine}\left(y_Q, y_D\right) = \frac{y_Q^T y_D}{\left\|y_Q\right\|\left\|y_D\right\|}. \tag{13}$$

After getting the text representation h via LSTM and calculating the similarity score, if Q&D are similar in the real world, the model would drag the corresponding representation vectors closer under the constraint condition of cosine similarity; if Q&D are lack of resemblance in the real world, the model would push the corresponding representation vectors away under the constraint condition of cosine similarity. We show the text representation under the constraint condition of cosine similarity in Fig. 3.

In order to implement the DSSM method, we transform the semantic similarity between Q and D^+ into posterior probability following (Huang et al., 2013), which is written as:

$$P\left(D^+ \mid Q\right) = \frac{\exp\left(\gamma R\left(Q, D^+\right)\right)}{\sum_{D' \in D} \exp\left(\gamma R\left(Q, D'\right)\right)}, \tag{14}$$

where γ is the smoothing factor of $softmax$. The objective of WL-DSSM is to maximize the semantic similarity of Q and D^+, which is equally to minimize $L(\Lambda)$:

$$L(\Lambda) = -\log \prod_{(Q, D^+)} P\left(D^+ \mid Q\right). \tag{15}$$

4 Experiments

We conduct experiments to assess the performance of WL-DSSM in dealing with identifying purchase intention from user query, namely, find a document of product description which is most related to user query semantically. We performed experiments on a server equipped with 2 NVIDIA GeForce RTX 3090 GPUs, sixteen AMD EPYC 7302 CPUs, and 126 GB RAM, running in a Jupyter notebook. All deep learning models are built using the Keras library (Chollet) with the Tensorflow backend.

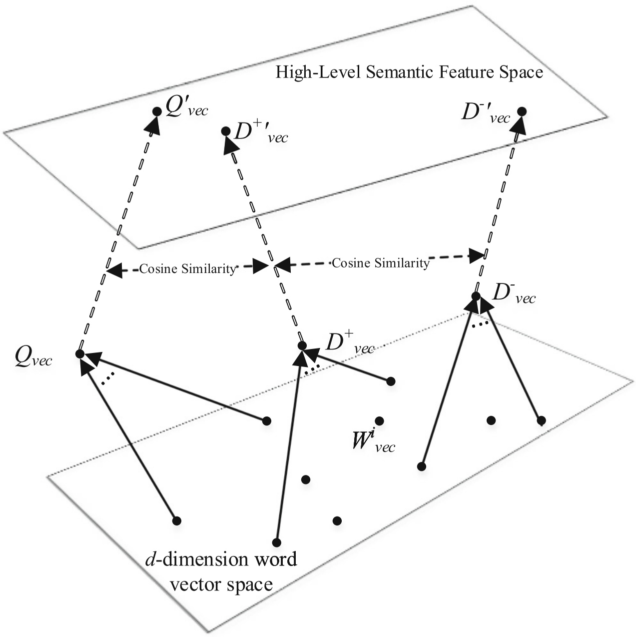

Fig. 3 Text representation under the constraint condition of cosine similarity. Each word W^i in the dictionary ϕ could get word vector W^i_{vec} in d-dimension word vector space. First, get the word representation after applying word segmentation with Q and D. Next, concatenate W^i_{vec} to form Q_{vec}, D^+_{vec} and D^-_{vec}, denoting query vector, right and false document of product description, respectively. Finally, the distance between $Q^{'}_{vec}$ and $D^{+'}_{vec}$ become smaller than $Q^{'}_{vec}$ and $D^{-'}_{vec}$ in the high-level semantic feature space under the constraint condition of cosine similarity

4.1 Dataset

We obtain a Q&D text of an E-Commerce dataset from the Home Depot Product Search Relevance competition[1]. Shoppers rely on Home Depot's product authority to find and buy the latest products and to get timely solutions to their home improvement needs. From installing a new ceiling fan to remodeling an entire kitchen, with the click of a mouse or tap of the screen, customers expect the correct results to their queries quickly. Speed, accuracy and delivering a frictionless customer experience are essential. However, since the names of building material are not common in the real world, the fact is that some shoppers are not familiar to product names so that the platform cannot provide a search result accurately. Therefore, we develop WL-DSSM to discover the customers' needs and to improve their shopping experience.

Table 1 summarizes the description of the dataset. For the value of relevance, three people rated the relevance of search_term and product_title, with 3 indicating perfectly matching and 1 denoting totally not matching, and then averaged the three score to get the final relevance. We show the text length of product_title, search_term and product_description in Fig. 4.

[1] https://www.kaggle.com/c/home-depot-product-search-relevance/data.

Table 1 Dataset description

Name	Description	Data type	Range
Id	A unique key of a pair of search_term, product_uid	Integer	1–240760
Product_uid	A unique key of product	Integer	100001–224428
Product_title	Product name	Text	7–147
Product_description	Document of product description	Text	8–5641
Search_term	User query	Text	1–60
Relevance	The relevance of search_term and product_title	Real number	1–3

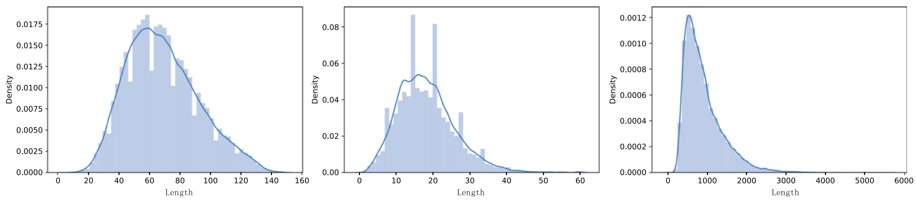

Fig. 4 Text length of product_title (left), search_term (middle) and product_description (right)

4.2 Pre-processing

In this paper, we pre-processed the dataset with the following five steps.

4.2.1 Conventional pre-processing of text dataset

Use function of *SnowballStemmer* to generate stem words and function of *tokenizer* to cut words. These two functions are from *nltk* package. Use *pandas* package to merge all text data before training Word2Vec model and the merged data is the input of CBOW. To simplify the input of WL-DSSM, we concatenate product_title and product_description as the new product_description.

4.2.2 Transformation of semantic similarity

The objective of this paper is to identify purchase intention. Therefore, we must transform the score of relevance to labels, indicating whether D is the correct product or not. Following (Choi et al., 2020), we first round the score of relevance to get discrete labels. Next, take 2.5 as a threshold and we conceive that D is matched with Q (labeled with 1) if the discrete label exceeds 2.5.

4.2.3 Procedures to adapt to DSSM

The inputs of DSSM are only Q and D^+. Q and D^+ are in one-to-one relationship, namely, each Q only has one D^+ and vice versa. As for D^-, DSSM randomly selects the number of J documents from D except D^+. Therefore, we first pick up the data labeled with 1. Next, we use the function of *drop_duplicates* to delete the duplicated data in case that D^- is matched with Q. Finally, we randomly select the number of J documents as D^- for each Q. In this way, our task becomes a multiclass problem, which has the number of $J + 1$ classes.

Table 2 Data splitting

Number	Training set $(J+1)\times 5495$	Verification set $(J+1)\times 1831$	Test set $(J+1)\times 1831$
Correct Q&D pair (9157)	5495	1831	1831
Wrong Q&D pair ($J \times 9157$)	$J \times 5495$	$J \times 1831$	$J \times 1831$

4.2.4 Unification of length with padding

From Table 1 and Fig. 4, we see that the range of the length of *product_title* is very large, so are product_description and search_term. Therefore, it is necessary to set a threshold t to define the maximum length for each field. For those lengths are smaller than t, we add padding to fill the gap. For those lengths exceed t, we delete the exceeded part. If we just use the maximum length of text as the t, on one hand, the dimension of features would be large, resulting in taking too much storage space and time to train the deep learning model. on the other hand, features would become sparse and it would have a bad impact on model convergence rate. As we merged product_title and product_description, we calculate the lengths of text of search_term and new product_description and select 24 and 1200 as the t respectively, covering over 81% texts.

4.2.5 Data splitting

Based on the above pre-processing steps, we split the data as Table 2 shows.

4.3 Parameter setting

All the hyper-parameters in the model are adjusted by the performance of the training set. The number of LSTM units was set to 128. The batch size was set to 64. Optimizer was set to Adadelta. The learning rate of the optimizer was set to 1e-1. Epochs were set to 100. When the validation loss did not inferior to the current lowest loss within 1 round, the model would stop learning early. The L2 regularization of the bias parameter is set to 0.32. The rate of Dropout was set to 0.6.

4.4 Evaluation measures

In many previous studies, the standard F1 is adopted as the evaluation standards (e.g., (Basso et al., 2020)). In this paper, F1 is also selected as the evaluation criteria of the model. Since we are only concerned about whether the algorithms identify the correct product or not, we ignore the evaluation of negative class.

True Positive (TP): a true positive is an outcome where the model correctly predicts the positive class;

False Positive (FP): a false positive is an outcome where the model incorrectly predicts the positive class;

False Negative (FN): a false negative is an outcome where the model incorrectly predicts the negative class.

Table 3 Description of comparative models

Model	Reference	Experiment
DSSM	Huang et al. (2013) used DSSM to map a query to its relevant documents at the semantic level.	The experiment uses word hashing to represent Q&D, applies DSSM to extract high-level feature, and finally uses softmax to classify D.
LSTM+DSSM	Palangi et al. (2014) used LSTM-DSSM to gain better information retrieve performance under the situation of long-term contect information.	After embedding with word hashing, the LSTM is used to extract sentence-level features semantically, and finally softmax is applied to output the probability of classifying.
Word2Vec+C-DSSM	Nikhil and Srivastava (2017) combined the Convolutional Deep Structured Semantic Models (C-DSSM) model with Word2Vec distributed representations of words to classify a document pair as relevant/irrelavant by assigning a score to it.	First, Word2Vec is used for word embedding. Next, CNN and DSSM are applied for text feature extraction sequentially. Finally, softmax function is used for output.

Precision: precision is used to note what proportion of positive identifications was actually correct;

Recall: recall is used to note what proportion of actual positives was identified correctly.

$$\text{Precision } = \frac{TP}{TP + FP}, \tag{16}$$

$$\text{Recall } = \frac{TP}{TP + FN}, \tag{17}$$

$$F1 = \frac{2 \times \text{ Precision } \times \text{ Recall}}{\text{Precision } + \text{ Recall}}. \tag{18}$$

4.5 Comparative models

To assess the performance of our WL-DSSM model against other competitive approaches, we are intended to conduct ablation study using the following existing models for comparative purpose. The description of comparative models is shown in Table 3.

4.6 Result and discussion

As table 4 demonstrate, the proposed model, which is shown in bold, is the best in the correct product identification using the Q&D text. It is widely recognized that multiclass classification tasks are often harder than binary classification tasks. Intuitively, when $J = 1$, there is just one negative sample for each item of the dataset and the goal of the model is to differentiate the positive sample from the two samples. However, when $J > 1$, the model has to choose the right sample from more samples. Thus, as J increases, which denotes the number of the negative sample increases, the F1 scores decrease among all of the models. However, the decreasing amplitude of our model's performance is the smallest among all of the models.

Table 4 Comparison of experimental results of each model

Model	Test F1 ($J = 1$)	Test F1 ($J = 2$)	Test F1 ($J = 3$)	Test F1 ($J = 4$)
DSSM	69.21%	53.70%	43.77%	38.04%
LSTM+DSSM	73.42%	58.50%	50.22%	43.37%
Word2Vec+C-DSSM	92.14%	90.42%	87.46%	87.01%
WL-DSSM	**95.81%**	**94.95%**	**93.11%**	**91.85%**

Table 5 Confusion Matrix ($J = 1$)

Actual class	Predicted class	
	True	False
True	1689	142
False	0	0

Table 6 Confusion Matrix ($J = 2$)

Actual class	Predicted class		
	True	False1	False2
True	1647	75	109
False1	0	0	0
False2	0	0	0

Table 7 Confusion Matrix ($J = 3$)

Actual class	Predicted class			
	True	False1	False2	False3
True	1602	87	76	66
False1	0	0	0	0
False2	0	0	0	0
False3	0	0	0	0

Table 4 also shows the results of the ablation study, where we compare a range of models that only use DSSM, combination using DSSM and LSTM, as well as combination using DSSM, CNN and Word2Vec. Looking at the DSSM and LSTM+DSSM model, we see that there is a minor performance difference between them. When we look at Word2Vec+C-DSSM model, however, it obtains a substantial improvement over DSSM and LSTM+DSSM. Futhermore, WL-DSSM achieves a minor improvement over Word2Vec+C-DSSM, hence proving WL-DSSM as a more competitive approach.

We can see the result in confusion matrix, which is an error table that is used for measuring and visualizing of the performance of any classification algorithms. In error confusion matrix, each column represents the predicted class measures and each row represents actual class measures (Kumar et al., 2018). We calculate the confusion matrix for each J via API from Scikit-learn library.[2] The results are shown in Tables 5, 6, 7 and 8. We can see that only the first row has the values, which is different from a normal confusion matrix. The reason is that there is no negative samples in the dataset and we construct them manually. When

[2] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html.

Table 8 Confusion Matrix ($J = 4$)

Actual class	Predicted class				
	True	False1	False2	False3	False4
True	1555	59	81	56	80
False1	0	0	0	0	0
False2	0	0	0	0	0
False3	0	0	0	0	0
False4	0	0	0	0	0

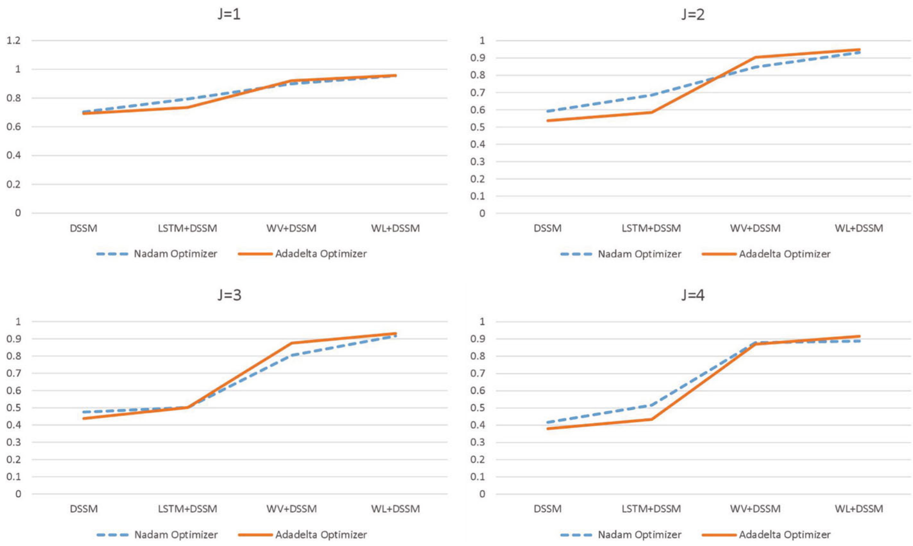

Fig. 5 Impact of optimizer tuning on F1 (Adadelta VS Nadam)

we input an item of the positive data, our model would construct J items of the negative data automatically. These $J + 1$ items are together and the negative samples cannot appear independently. Thus, there is no value for the rows of False in actual class. These tables show that the value of TP decreases as J increases.

Fig. 5 illustrates the variation of F1 score with optimizer. Generally, Nadam optimizer is more effective in achieving higher F1 score than Adadelta for DSSM and LSTM + DSSM, while it is opposite for WV + DSSM and WL + DSSM. Specifically, the performances of the two optimizers are about the same when $J = 1$ for WL + DSSM and $J = 3$ for LSTM + DSSM. Additionally, Nadam optimizer has higher convergency speed than Adadelta.

DSSM can draw Q and D^+ closer while pull Q and D^- away for information retrieval. This paper uses DSSM to conduct purchase intention identification as a comparative model. Since the method of word embedding is word hashing model, DSSM cannot extract the semantic feature of words and ignore the sequences of words. LSTM + DSSM model adds LSTM neural network to DSSM, which can effectively learn sequence information and extract sentence-level feature, so the final model's identification performance is better than DSSM. Word2Vec + C-DSSM replaces word hashing model with Word2Vec algorithm, aiming to get word embedding semantically rather than independent word embedding. Additionally, Word2Vec + C-DSSM adds convolutional layer to DSSM, which carries out the convolution extraction of text features to learn text semantics. The identification performance is significantly enhanced when DSSM introduces Word2Vec and convolutional layer.

However, convolutional layer also has no ability to record the sequence information among word vectors. In order to extract the sentence-level feature and store sequence information, this paper adopts LSTM neural network that connects nodes between hidden layers and adds three control units of the input gate, output gate, and forget gate, which could remember the past information and solve the problem of long sequence dependences. Therefore, the performance of classification is obviously improved, and semantic learning is enhanced via Word2Vec and LSTM.

5 Conclusions

Shoppers rely on E-Commerce platforms to find and buy the latest products. Speed, accuracy and delivering a frictionless customer experience are essential for E-Commerce platforms. However, customers often leave a query with ambiguous intention, resulting in unsatisfied searching results and deteriorating customer experience. Especially in vertical E-Commerce, queries with ambiguous intention are very common in that some product names are unpopular and users even have no idea of what is the name the product they need. Although several deep learning models to identify purchase intention have been developed, none of these give a glance to DSSM, which possess great performance in information retrieval. Our research proposes a model, combining Word2Vec and LSTM as well as DSSM, that can accurately predict the correct products to improve customers' shopping experience in E-commerce platform. We summarize the theoretical and practical contributions as well as the future research opportunities as follows.

5.1 Theoretical contribution

This paper proposes a purchase intention identification model based on the existing text semantics analysis. This research uses advanced text processing deep learning and artificial intelligence technology to mine and analyze the Q&D text data on the E-Commerce platforms based on peer research. Aiming at addressing the problem of identifying purchase intention from user's query, we construct WL-DSSM model for purchase intention identification by analyzing the Q&D text data based on Word2Vec, LSTM and DSSM. This model is used to identify the purchase intention on the E-Commerce platform. The proposed model is better than other existing models in terms of F1 score. Combined with the advantages of each technology, including Word2Vec, LSTM and DSSM, WL-DSSM improves the performance of purchase intention identification. Additionally, data imbalance is very common and has a bad impact on training deep learning models. The strategy of generating negative samples makes the data in each class balanced. This strategy may helpful for training other deep learning models.

5.2 Practical contribution

Depend on the research method proposed in this paper, customers could find and buy the products quickly and accurately. Our model can classify a document of product description into relevant/irrelevant. The relevant product means that it possesses the biggest semantic similarity score with user query and it may be the desirable product in the highest degree. The irrelevant product means that the semantic similarity score is lower than the biggest semantic similarity score and it may be not the correct product shopper is finding. For the E-Commerce platforms, they could utilize this method to output semantic similarity scores of Q&D and the platforms can sort the scores in reverse order. According to the sort results, the platform could return a list of products for the user query. Additionally, our model could remove or minimize human input in search relevance evaluation, which is a slow and subjective process. According to our model, the platforms could improve shopping experience and thus attract and retain customers to strengthen competitive advantage. For the customers, our method allow them to type natural language or what they want to do (e.g. lay the foundations) in the query box in case that customers do not know the name of the product they want.

5.3 Future scope of research

Despite these significant insights, there are still some limitations in this paper. First, the proposed model has only been tested on a dataset from the Home Depot platforms. However, different results might have been obtained if the model had been tested on other E-Commerce platforms. The findings might not be directly applicable on other datasets because the languages may be different. To overcome these limitations, further research should certainly explore the corresponding pre-processing methods for different languages and the model should be tested on multiple datasets collected from additional online platforms. Second, we develop our model using Word2Vec, which is trained by the current corpus. If a new word emerged, it could not get a semantic representation from Word2Vec model. In the future, we would explore a dynamic word representation method and combine it with the existing module of our model.

Acknowledgements This study was supported by the National Natural Science Foundation of China under grant number 72174086 and 71801126, and the Fundamental Research Funds for the Central Universities under grant number NW2020001.

References

An, C., Huang, J., Chang, S., & Huang, Z. (2016). Question similarity modeling with bidirectional long short-term memory neural network. In *2016 IEEE first international conference on data science in cyberspace (DSC)* (pp. 318–322). IEEE.

Balakrishnan, J., & Dwivedi, Y.K. (2021). Conversational commerce: entering the next stage of ai-powered digital assistants. *Annals of Operations Research*, pp. 1–35.

Basso, S., Ceselli, A., & Tettamanzi, A. (2020). Random sampling and machine learning to understand good decompositions. *Annals of Operations Research, 284*(2), 501–526.

Bengio, Y., Frasconi, P., & Simard, P. (1993). The problem of learning long-term dependencies in recurrent networks. In *IEEE international conference on neural networks* (pp. 1183–1188). IEEE.

Choi, J.I., Kallumadi, S., Mitra, B., Agichtein, E., & Javed, F. (2020). Semantic product search for matching structured product catalogs in e-commerce. arXiv preprint arXiv:2008.08180.

Chollet, F.: Keras. https://github.com/fchollet/keras.

Chopra, S., Hadsell, R., & LeCun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, (vol. 1, pp. 539–546). IEEE.

Das, A., Yenala, H., Chinnakotla, M., & Shrivastava, M. (2016). Together we stand: Siamese networks for similar question retrieval. In *proceedings of the 54th annual meeting of the association for computational linguistics* (Volume 1: Long Papers, pp. 378–387).

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Eachempati, P., Srivastava, P. R., Kumar, A., Tan, K. H., & Gupta, S. (2021). Validating the impact of accounting disclosures on stock market: A deep neural network approach. *Technological Forecasting and Social Change, 170,* 120903.

Fu, B., & Liu, T. (2016). Implicit user consumption intent recognition in social media. *Journal of Software, 27,* 2843–2854.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation, 9*(8), 1735–1780.

Huang, P.-S., He, X., Gao, J., Deng, L., Acero, A., & Heck, L. (2013). Learning deep structured semantic models for web search using clickthrough data. In *proceedings of the 22nd ACM international conference on information & knowledge management*,(pp. 2333–2338).

Jia, Y., Han, D., Lin, H., Wang, G., & Xia, l. (2020). Consumption intent recognition algorithms for weibo users. *Acta Scientiarum Naturalium Universitatis Pekinensis,56*, (pp. 68–74).

Kumar, A., Singh, J.P., Dwivedi, Y.K., & Rana, N.P. (2020). A deep multi-modal neural network for informative twitter content classification during emergencies. *Annals of Operations Research*, (pp. 1–32).

Kumar, A., Gopal, R. D., Shankar, R., & Tan, K. H. (2022). Fraudulent review detection model focusing on emotional expressions and explicit aspects: investigating the potential of feature engineering. *Decision Support Systems, 155,* 113728.

Kumar, A., Shankar, R., & Aljohani, N. R. (2020). A big data driven framework for demand-driven forecasting with effects of marketing-mix variables. *Industrial marketing management, 90,* 493–507.

Kumar, A., Shankar, R., & Thakur, L. S. (2018). A big data driven sustainable manufacturing framework for condition-based maintenance prediction. *Journal of computational science, 27,* 428–439.

Kwek, C. L., Tan, H. P., & Lau, T.-C. (1970). Investigating the shopping orientations on online purchase intention in the e-commerce environment: a malaysian study. *The Journal of Internet Banking and Commerce, 15*(2), 1–21.

Lee, D.-G., Lee, K.-H., & Lee, S.-Y. (2015). Implicit shopping intention recognition with eye tracking data and response time. In *proceedings of the 3rd international conference on human-agent interaction*, (pp. 295–298).

Li, C., Du, Y., & Wang, S. (2017). Mining implicit intention using attention-based rnn encoder-decoder model. In *International Conference on Intelligent Computing*, (pp. 413–424). Springer.

Liao, Y., Peng, Y., Shi, S., Shi, V., & Yu, X. (2020). Early box office prediction in china's film market based on a sLiaotacking fusion model. *Annals of Operations Research*, (pp. 1–18).

Luo, X., Gong, Y., & Chen, X. (2018). Central intention identification for natural language search query in e-commerce. In *eCOM@ SIGIR*.

Mai, L., & Le, B. (2020). Joint sentence and aspect-level sentiment analysis of product comments.*Annals of Operations research*, (pp. 1–21).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Mikolov, T., Karafiát, M., Burget, L., Cernockỳ, J., & Khudanpur, S. (2010). Recurrent neural network based language model. In *Interspeech*, (vol. 2, pp. 1045–1048). Makuhari.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *advances in neural information processing systems*, (pp. 3111–3119).

Nassif, H., Mohtarami, M., & Glass, J. (2016). Learning semantic relatedness in community question answering using neural models. In *proceedings of the 1st workshop on representation learning for NLP*, (pp. 137–147).

Nikhil, N., & Srivastava, M.M. (2017). Content based document recommender using deep learning. In 2017 *International Conference on Inventive Computing and Informatics (ICICI)*, (pp. 486–489). IEEE.

Palangi, H., Deng, L., Shen, Y., Gao, J., He, X., Chen, J., Song, X., & Ward, R. (2014). Semantic modelling with long-short-term memory for information retrieval. arXiv preprint arXiv:1412.6629.

Palangi, H., Deng, L., Shen, Y., Gao, J., He, X., Chen, J., et al. (2016). Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech, and Language Processing, 24*(4), 694–707.

Peng, X. (2018). Research on recognition method in customer purchase intention based on intelligent customer service learning. *Master's thesis, Shandong University of Finance and Economics*.

Qayyum, A., Razzak, I., Tanveer, M., & Kumar, A. (2021). Depth-wise dense neural network for automatic covid19 infection detection and diagnosis. *Annals of operations research*, (pp. 1–21).

Qian, Y., Ding, X., Liu, T., & Yiheng, C. (2017). Identification method of user's travel consumption intention in chatting robot. *Sci Sin Inform, 47,* 997–1007.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

Rodríguez, P., Bautista, M. A., Gonzalez, J., & Escalera, S. (2018). Beyond one-hot encoding: Lower dimensional target embedding. *Image and Vision Computing, 75,* 21–31.
Rong, X. (2014). word2vec parameter learning explained. arXiv preprint arXiv:1411.2738.
Seger, C. (2018). An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing.
Sengupta, P., Biswas, B., Kumar, A., Shankar, R., & Gupta, S. (2021). Examining the predictors of successful airbnb bookings with hurdle models: Evidence from europe, australia, usa and asia-pacific cities. *Journal of Business Research, 137,* 538–554.
Shen, Y., He, X., Gao, J., Deng, L., & Mesnil, G. (2014). A latent semantic model with convolutional-pooling structure for information retrieval. In *proceedings of the 23rd ACM international conference on conference on information and knowledge management*, (pp. 101–110).
Sordoni, A., Bengio, Y., Vahabi, H., Lioma, C., Grue Simonsen, J., & Nie, J.-Y. (2015). A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *proceedings of the 24th ACM international on conference on information and knowledge management*, (pp. 553–562).
Sun, X., Xu, W., Jiang, H., & Wang, Q. (2021). A deep multitask learning approach for air quality prediction. *Annals of Operations Research, 303*(1), 51–79.
Tchuente, D., & Nyawa, S. (2021). Real estate price estimation in french cities using geocoding and machine learning. *Annals of Operations Research*, (pp. 1–38).
Xia, H., Liu, J., & Zhang, Z.J. (2020). Identifying fintech risk through machine learning: analyzing the q&a text of an online loan investment platform. *Annals of Operations Research*, (pp. 1–21).
Xiao, L., Wang, G., & Zuo, Y. (2018). Research on patent text classification based on word2vec and lstm. In *2018 11th international symposium on computational intelligence and design (ISCID)*, (vol. 1, pp. 71–74. IEEE).
Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., & Le, Q.V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems* **32**.
Zhou, F., Jin, L., & Dong, J. (2017). Review of convolutional neural network. *Chinese Journal of Computers, 40,* 1229–1251.