Khan et al. Beni-Suef University Journal of Basic and Applied Sciences (2020) 9:10
https://doi.org/10.1186/s43088-020-0035-7

Beni-Suef University Journal of Basic and Applied Sciences

RESEARCH Open Access

Customer gender prediction system on hierarchical E-commerce data

Mohammad Masud Khan[1†], Mohammad Golam Sohrab[2*†] and Mohammad Abu Yousuf[1]

Abstract

Background: E-commerce services provide online shopping sites and mobile applications for small and medium sellers. To provide more efficient buying and selling experiences, a machine learning system can be applied to predict the optimal organization and display of products that maximize the chance of bringing useful information to user that facilitate the online purchases. Therefore, it is important to understand the relevant products for a gender to facilitate the online purchases. In this work, we present a statistical machine learning (ML)-based gender prediction system to predict the gender "male" or "female" from transactional E-commerce data. We introduce different sets of learning algorithms including unique IDs decomposition, context window-based history generation, and extract identical hierarchy from training set to address the gender prediction classification system from online transnational data.

Results: The experiment result shows that different feature augmentation approaches as well as different term or feature weighting approaches can significantly enhance the performance of statistical machine learning-based gender prediction system.

Conclusions: This work presents a ML-based implementational approach to address E-commerce-based gender prediction system. Different session augmentation approaches with support vector machines (SVMs) classifier can significantly improve the performance of gender prediction system.

Keywords: Text classification, Indexing, Term weighting, Machine learning, Statistical learning, Feature selection, Classifier

1 Background

This work presents research on implementing a machine learning (ML)-based automatic text classification (ATC) [1–4] system for gender prediction on transnational E-commerce data. E-commerce services provide online shopping sites and mobile applications for small and medium sellers. To provide more efficient buying and selling experiences, a machine learning system can be applied to predict the optimal organization and display of products that maximize the chance of bringing useful information to user that facilitate the online purchases. In this work, we address this problem to predict the gender "male" or "female" from browsing history using statistical machine learning technique. A set of different approaches are considered to enhance the gender prediction system. We introduce different feature augmentation approaches including unique IDs decomposition, context window-based history generation, and extract identical hierarchy from training set to address the gender prediction classification system from online transnational data. The experiment result shows that different feature augmentation approaches as well as different term or feature weighting approaches can significantly enhance the performance of statistical ML-based gender prediction system.

In E-commerce, business-to-business-to-customer (B2B2C) runs several services that provide online shopping sites and mobile applications for small and medium sellers. Transaction data, such as product browsing and purchasing activities, from buyer, and product portfolio, from seller, can be aggregated, to provide more efficient buying and selling experiences. For example, statistical machine learning (SML) techniques can be applied to

* Correspondence: sohrab.mohammad@aist.go.jp
†Mohammad Masud Khan and Mohammad Golam Sohrab contributed equally to this work.
[2]National Institute of Advanced Industrial Science and Technology, 2-4-7 Aomi, Koto-ku, Tokyo 135-0064, Japan
Full list of author information is available at the end of the article

predict the optimal organization and display of products that maximize the chance of bringing useful information to user, facilitate the online purchases.

The primary motivation of exploiting the session augmentation-based approaches can be attributed to two main properties. First, to create a more effective features that can enhance the system performance from product viewing transaction sessions by augmenting the sessions' $S = \{s_1, s_2, ..., s_j\}$ features. Second, is to create different simple term or feature weighing approaches that can generate a more effective classifier and can further boost the system performance. The term or feature weight is the degree of importance of term or feature t_i in each transaction session s_j. The term weighting approach plays a very significant role to enhance automatic text classification (TC). Therefore, an effective term or feature weighting approach can generate more information-rich terms and assign appropriate weighting values to the terms.

Most of the works in SML-based approaches are based on either document- or class-indexing which is incorporated with document space [2] or category space [3, 4]. In this dataset, the transactional sessions or log sessions are free from either in document or in class space. Therefore, to deal with such data for statistical classification system, we use binary approach, cosine binary approach to normalize the features of each session, and finally normalize term frequency-based approach to address the system.

In this work, each session of sessions from transaction data is modeled as feature vectors, extracted from the session product IDs. The classification task for gender prediction is a binary classification problem, where a session is labeled as "male" or "female" category. The trainable classifier is expected to learn the patterns, by identifying relevant feature values which are most correlated with the classes "male" or "female." When a new session or log file is given to the system, the learned patterns are used to classify each session into either a "male" or "female" session and give it a certain score value between "0" and "1." Our goal is to show that an effective gender prediction system can be created relying solely on different session augmentation and simple weighting approaches to enhance the classification task. The session augmentation with unique IDs decomposition is capable to produce good classification score. Session augmentation with context window approach along with unique IDs decomposition can also boost the system performance. Finally, session augmentation with identical hierarchy incorporated with all together can further boost the system performance. Therefore, these approaches are effective to improve the SML-based gender prediction system.

Recently, many experiments have been conducted using different term weighting approaches [1, 3–7] to address the classification task as a statistical method. Document-indexing-based and four fundamental information-element-based [3, 4] weighting approaches are considered the most popular term weighting method in ATC.

Recently, many experiments have been conducted using a document- and class-indexing-based term weighting approach to address the classification task as a statistical method [2, 7–11]. *TF.IDF* is the most popular term weighting method in successfully performing the ATC task and document-indexing [2]. Salton Buckley [11] discussed many term weighting approaches in the information retrieval (IR) [12–16] field and found that normalized *TF.IDF* is the best document weighting function. Therefore, *TF.IDF* is considered as one of the most standard weighting approaches in statistical machine learning-based approaches, especially in text classification. In contrast, recently Sohrab [3, 4] proposed class-indexing-based ($TF.IDF.ICS_\delta F$) indexing, a subset of documents from the global document space $D = \{d_1, d_2. . . d_n\}$ is allocated to a certain class c_k where ($k = 1, 2, . . . m$)) according to their topics in order to create a boundary line vector space in the training procedure. Therefore, the class space is defined as $C = \{(d_{11}, d_{12}, . . . d_{1n}) C_1, (d_{21}.d_{22}, . . . d_{2n}) C_2, . . ., (d_{m1}, d_{m2}, . . . d_{mn}) Cm\}$ where a set of documents with same topics is assigned to a certain class c_k. With class-indexing-based term weighing approach they have outperformed over all traditional weighting approaches.

In the last few years, researchers have attempted to improve the performance of TC by exploiting statistical classification approaches and machine learning techniques, including probabilistic Bayesian models [17], support vector machines (SVMs) [18, 19], decision trees (Lewis and Ringuette) [17], Rocchio classifiers [20], and multivariate regression models [21]. Among them, SVMs-based classifier achieved great success in classification problem. Therefore, we adopt the SVMs as a classifier to learn and predict the gender prediction system.

2 Methods

The proposed ML-based gender prediction system follows the steps, including (i) session augmentation approach to generate candidate features, (ii) session to vector generation, and (iii) classifiers.

2.1 Session augmentation approach

Session augmentation approach can be decomposed into three different representations including (i) session augmentation with unique IDs decomposition, (ii) session augmentation with context window, and (iii) session augmentation with identical hierarchy. In subsequent sections, we will clarify how these augmentations are produced.

2.1.1 Session augmentation with unique IDs decomposition

In the statistical-based classification method [13, 15, 22–25], it is important to generate good features from text and then apply term weighting approach to generate text to vector which is an input of a classifier to address classification problem. An example of session or single product viewing log from training dataset is composed of four columns and can be read as, u10001, 2014-11-14 00:02:14, 2014-11-14 00:02:20, A00001/B00001/C00001/D00001/ where u10001 is session ID, 2014-11-14 00:02:14 and 2014-11-14 00:02:20 correspond to a session start- and end-time respectively, and a list of product IDs separated by back slash are as A00001/B00001/C00001/D00001.

In case of multiple products view of a single session, the product list is separated with semicolon as, u10001, 2014-11-14 00:02:14, 2014-11-14 00:02:20, A00001/B00001/C00001/D00001/; A00002/B00002/C00002/D00002/.

A distribution of product IDs in the dataset is decomposed into two different combinations uni-gram and bi-gram compositions.

2.1.1.1 Uni-gram-based feature composition For a given product IDs of a single session "A00001/B00001/C00001/D00001/," first generate four different features based on uni-gram, i.e., "A00001," "B00001," "C00001," and "D00001." Since the system is binary classification task to predict male or female label. To adding more features, augment the sessions by product IDs with merging label "A00001-label," "B00001-label," "C00001-label," and "D00001-label" where "label" is indicating a certain session's label which can be "female" or "male" category.

2.1.1.2 Bi-gram-based feature composition The product IDs follow the hierarchy from top category "A" to leaf category "D" by following the intermediate categories "C" and "D." Therefore, in the bi-gram feature composition, follow the top-down manner to create bi-gram features. Features from the above single session are "A00001-B00001," "B00001-C00001," and "C00001-D00001" based on the label augmented features are "A00001-B00001-label," "B00001-C00001-label," and "C00001-D00001-label." Based on Uni- and Bi-gram-based approaches, we can generate more features for the gender prediction system.

2.1.2 Session augmentation with context window

It is important to analyze the behavior that how similar the current session with surrounding sessions. To address the contextual information of this task, we create a history based on window size. The context window-based approach augments the current session. A session space $S = \{s_1, s_2, s_3, ..., s_m\}$ is a set of sessions from training dataset. In the context window approach, we set a window size from a certain session to generate history based on surrounding sessions. For instance, if we set window size = 3, then it creates history from current position to its two previous sessions and to two next sessions. Figure 1 shows an example of setting context window among the sessions. In this figure, window size = 3 is set on session 3 to generate the history incorporating from two previous sessions and two more next sessions. In Fig. 1, term $t_i = t_1, t_2, ..., t_n$ represents corresponding session IDs of a single session. We build the history of a certain session if it does satisfy the following conditions,

Algorithm 1: History Generation
Input: Sessions
Output: Augmented Session
1: **procedure** HISTORY (CURRENT-SESSION, PREVIOUS-SESSION, NEXT-SESSION, WS=3)
2: **while** Window-size is not WS **do**
3: **if** curSession.prevSession.endTime != curSession.startTime **then**
4: curSession.features ← prevSession.features
5: **end if**
6: **if** curSession.endTime != curSession.nextSession.startTime **then**:
curSession.features ← nextSession.features
8: **end if**
9: **end while**
10: return augmented-context-session
11: **end procedure**

Algorithm 1 shows the session augmentation based on context window to create history of a certain session. In this Algorithm, augment the current session's features with previous and next session's features based on window size.

2.1.3 Session augmentation with identical hierarchy

In this section, we will discuss creation of hierarchy and further analysis based on the hierarchy which leads to create identical hierarchy from all the product log view of training data. First, we generate the hierarchy from training data where Fig. 2 shows the architecture of the hierarchy. In F,ig. 2 the IDs starting with letter "A" are top categories or nodes of the hierarchy. The IDs starting with letters "B" and "C" are the intermediate categories or nodes and finally the IDs starting with letter "D" are leaf categories or nodes. In the training data, it has 22,440 hierarchies. Top nodes are consisting of 11 distinct categories where intermediate nodes starting with "B" and "C" are consisting of 91 and 441 categories respectively. And the leaf nodes have 36,122 different categories that represent the target products.

2.1.3.1 Mapping identical hierarchy Once we generated the hierarchy, we then enumerate the identical hierarchy from intermediate nodes or categories. Here, the term identical hierarchy denotes if a hierarchy construct with parent and child category and the hierarchy appears only in a certain category. From the definition of identical hierarchy, it seems that the identical hierarchies are only possible to be created from categories starting with "B" and "C" and not from "A" and "D" since they are free

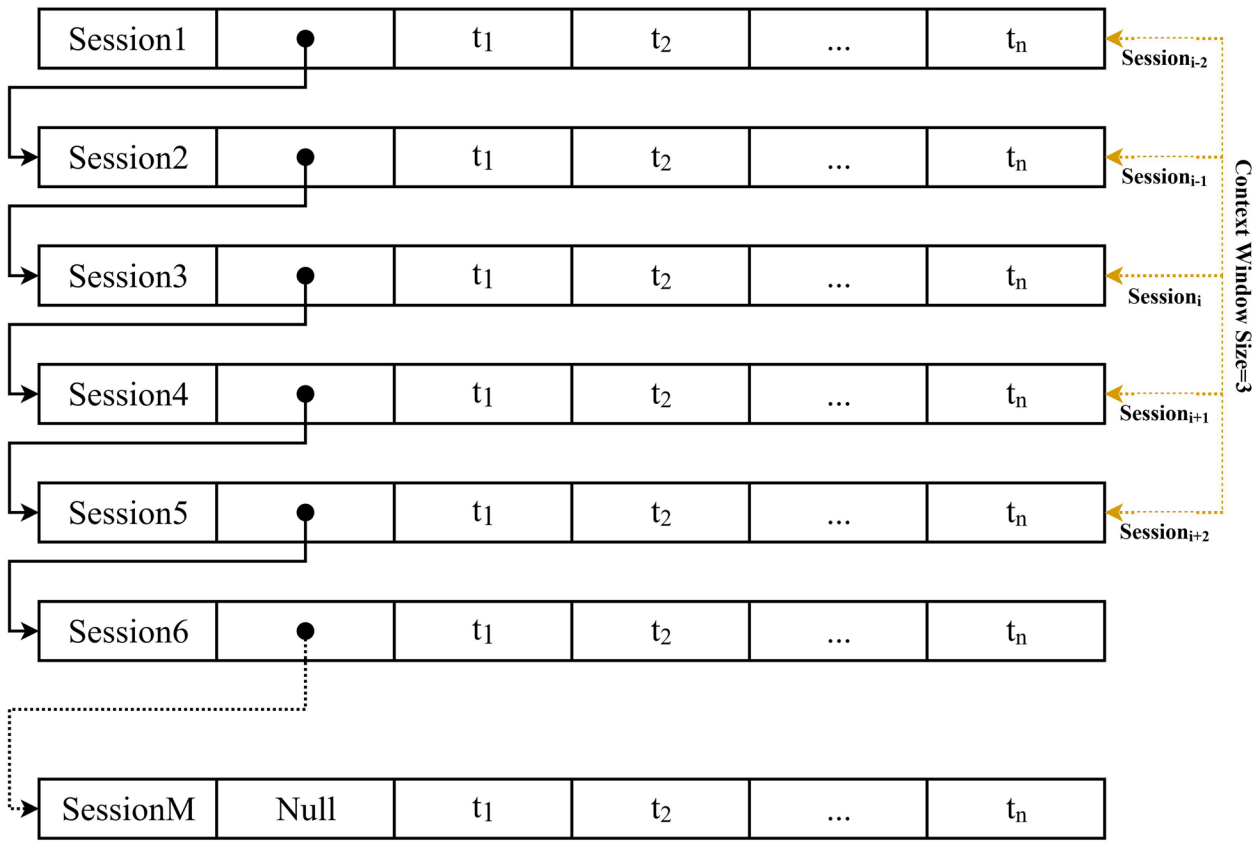

Fig. 1 Session augmentation with context window size = 3

from parent and child categories respectively. For instance, a given product IDs of a single session "A00003/B00008/C00026/D00070/" and can be represented as in Fig. 3, where "A00003" is the parent category of "B00008," "B00008" is the parent category of "C000026," and "C00026" is the parent of "D00070." Here, first we determine the identical categories if and only if a certain product IDs that appears in certain gender category like "male" or "female." For a certain top, intermediate, and leaf-level categories, the label weight is assigned with a numeric value between 1 and 0 for overlapping and non-overlapping categories. If categories in intermediate-level are non-overlapped with gender categories, we then determine as an identical category and extract all possible parent- and child-list from that identical categories which are denoted as identical hierarchy of a certain gender's label. Figure 4 shows an example of session augmentation with identical hierarchy. Figure 4a represents a hierarchy of "A00003/B00008/C00026/D00070/" where "B00008" is an identical category based on training

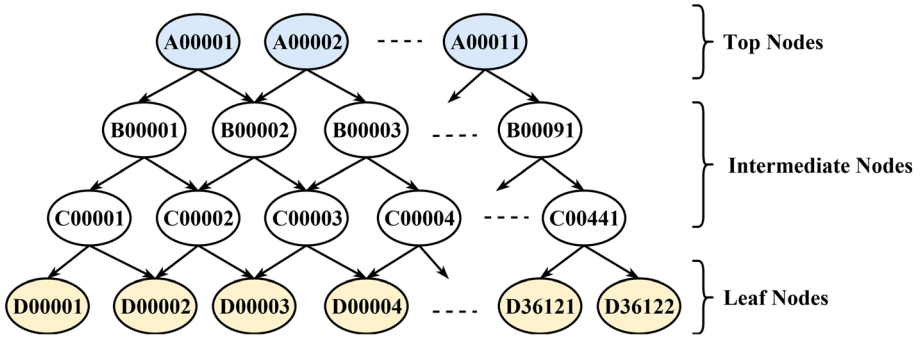

Fig. 2 Hierarchy from product log IDs

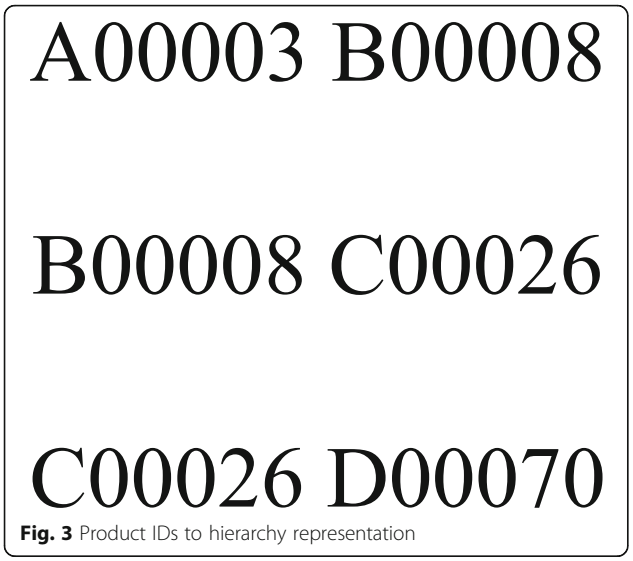

Fig. 3 Product IDs to hierarchy representation

data. Figure 4b shows the parent- and child-list from identical category "B00008" which are extracted from training samples. Finally, Fig. 4c shows the child-list from "B00008" which are embedded to the graph. In Fig. 4a, we can represent the hierarchy as in stated in Fig. 3. In contrast, the session is augmented after the child categories are embedded in Fig. 4c and can be represented as in Fig. 5.

2.2 Session to vector generation

In the machine learning workbench, term to vector generation [3, 4] plays a very significant role to boost the system performance. After the session augmentation process, we then assign a weight of each term (t_i) in a certain session (s_j) using binary weighting approach where term weight is assigned a numeric value between 1 and 0 for overlapping and non-overlapping terms in a certain label (c_k) and cab ne denoted as,

$$w(t_i) = \begin{cases} 1, if \text{ term appears in a certain category}, t_i \varepsilon \ c_k \\ 0, \quad \text{otherwise} \end{cases} \tag{1}$$

We then normalize the session using cosine normalization and is denoted as,

$$W^{\text{norm}}(t_i) = \frac{W(t_i)}{\sqrt{\sum_{t_i \ \varepsilon \ s_j} [W(t_i)]^2}}, \tag{2}$$

Besides, we also introduce a simple weighting approach based on term frequency (TF). We calculate the number of terms appear in the session and normalize as,

$$W^{TF}(t_i) = \frac{TF(t_i)}{TF(t_i) + 1}, \tag{3}$$

2.3 Classifier

Machine learning method constructs a classification model to predict the category of new test documents by learning the statistic of training data. In the machine learning workbench, support vector machines (SVMs) are achieved great success in classification problem and are considered one of the most robust and accurate methods among all well-known classification algorithms. In order to evaluate the effectiveness of the proposed ML-based gender prediction system, we use the liblinear[1] package, a library for large linear classification. It supports L2-regularized classifiers, L2-loss linear SVM, L1-loss linear SVM, and logistic regression (LR), L2-loss linear SVM and logistic regression (LR) for SVM classifier etc. The

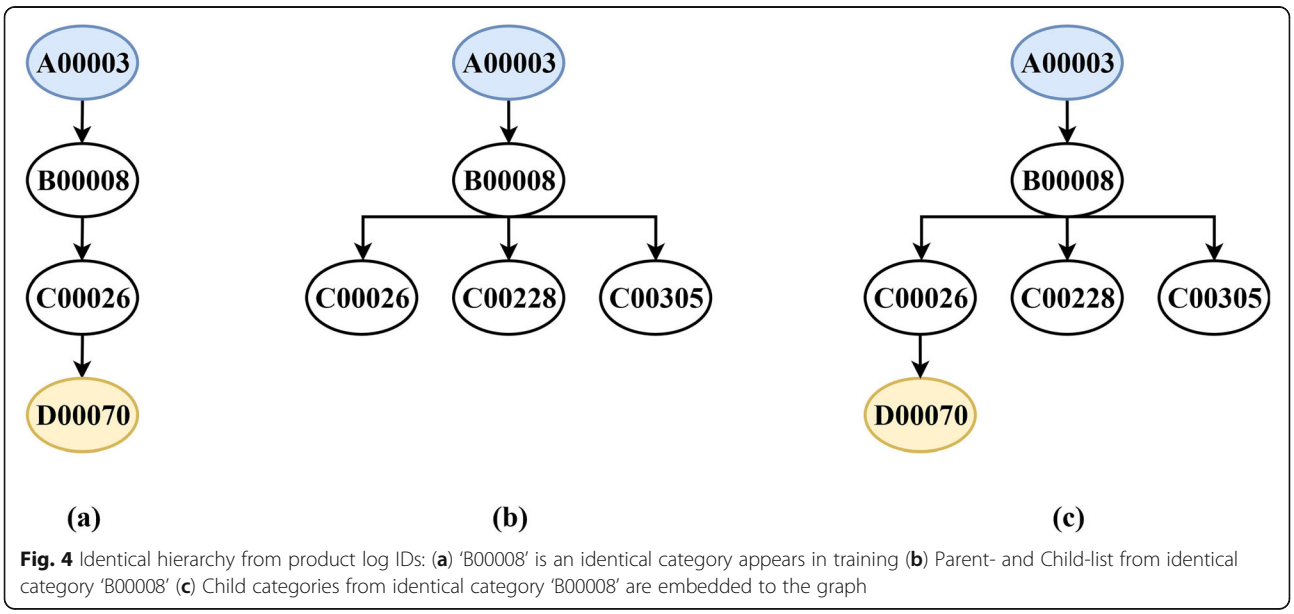

Fig. 4 Identical hierarchy from product log IDs: (**a**) 'B00008' is an identical category appears in training (**b**) Parent- and Child-list from identical category 'B00008' (**c**) Child categories from identical category 'B00008' are embedded to the graph

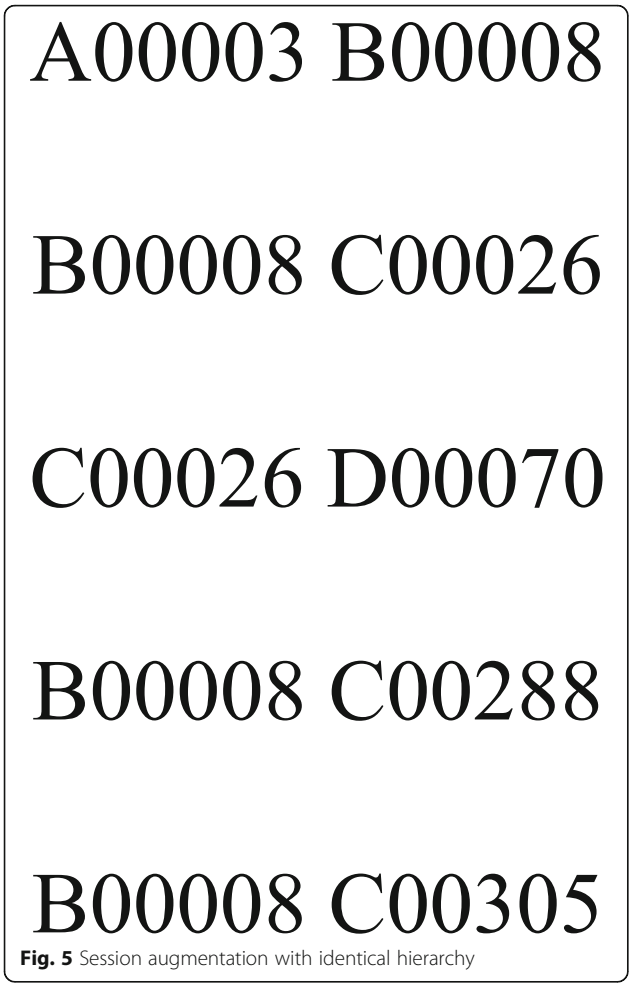

Fig. 5 Session augmentation with identical hierarchy

parameter $-c$ was set to 1.0, which is considered as a default setting in this toolbox.

3 Results

In this section, we provide empirical evidence for the effectiveness of proposed gender prediction system using different session augmentation-based approaches that can perform to compare the results of our proposed machine learning-based classification system for gender prediction based on E-commerce data.

3.1 Experimental dataset

The data in this experiment for customer gender prediction is used which is provided by financing and promoting technology (FPT)[2] group and the dataset[3] is divided into separate training and test sets—*trainingData.csv* and *testData.csv*, respectively. Training contains 15,000 records which correspond to product viewing logs. In 15,000 sessions, 11,703 and 3297 sessions are in "female" and "male" category respectively which is quite unbalanced dataset. A single log is composed of four columns, separated by commas where the first column is a session ID. The second and third columns correspond to a session start time and session end time, respectively. The last column contains a list of product IDs. Consecutive product IDs are separated by semicolons. Besides, a *trainingLabels.csv* file is available which contains label of corresponding sessions. Each product ID can be decomposed into four different IDs which are separated by slashes. The IDs starting with letter "A" are the most general categories and those starting with "D"' correspond to individual products. The IDs which start with "B" and "C" are associated with subcategories and sub-subcategories, respectively.

[2]Available at https://www.fpt.com.vn/en/
[3]Available at https://knowledgepit.ml/pakdd15-data-mining-competition/

3.2 Cross validation

To split the data-set, we adopt n-fold cross-validation problem where n = 3, 5, 10. For instance, three-fold cross-validation problem, we randomly split the data into three different folds and each turn during training one-fold is used as test and others are training. For model validation, cross-validation technique is very effective for assessing how the results of a statistical analysis will generalize to an independent dataset.

3.3 Performance measurement

The standard methods used to judge the performance of a gender prediction are precision, recall, and the F1 measure [8, 26]. These measures are defined based on a contingency table of predictions for a target category c_k. The precision $P(C_k)$, recall $R(C_k)$, and the F1 measure $F_1(C_k)$ are defined as in Eqs. 4–6 respectively:

$$P(C_k) = \frac{TP(C_k)}{TP(C_k) + FP(C_k)} \tag{4}$$

$$R(C_k) = \frac{TP(C_k)}{TP(C_k) + FN(C_k)} \tag{5}$$

$$F_1(C_k) = \frac{2.P(C_k).R(C_k)}{P(C_k) + R(C_k)} = \frac{2.TP(C_k)}{2TP(C_k) + FP(C_k) + FN(C_k)}, \tag{6}$$

$TP(C_k)$ is the set of test sessions correctly classified to the category C_k, $FP(C_k)$ is the set of test sessions incorrectly classified to the category, $FN(C_k)$ is the set of test sessions wrongly rejected, and $TN(C_k)$ is the set of test sessions correctly rejected. To compute the average performance, we used macro-average, micro-average, and overall accuracy. The macro-average of precision (P^M), recall (R^M), and the F_1 measure (F_1^M) of the class space are computed as in Eqs. 7–9 respectively:

$$P^M = \frac{1}{m}\sum_{k=1}^{m} P(C_k) \tag{7}$$

$$R^M = \frac{1}{m}\sum_{k=1}^{m} R(C_k) \tag{8}$$

$$F_1^M = \frac{1}{m}\sum_{k=1}^{m} F_1(C_k) \tag{9}$$

Therefore, the micro-average of precision (P^μ), recall (R^μ), and the F_1 measure (F_1^μ) of the class space are computed as in Eqs. 10–12 respectively:

$$P^\mu = \frac{\sum_{k=1}^{m} TP(C_k)}{\sum_{k=1}^{m}(TP(C_k) + FP(C_k))} \tag{10}$$

$$R^\mu = \frac{\sum_{k=1}^{m} TP(C_k)}{\sum_{k=1}^{m}(TP(C_k) + FN(C_k))} \tag{11}$$

$$F_1^\mu = \frac{2.P^\mu.R^\mu}{P^\mu + R^\mu} \tag{12}$$

3.4 System performances

In this section, we judge the system performances on different weighting approaches as well as session augmentation approaches, including unique IDs decomposition, context window generation, and identical hierarchy with *n*-fold cross-validation.

3.4.1 The effect of unique IDs decomposition

Table 1 shows the effect of adding uni-gram and bi-gram-based unique IDs decomposition from each session where the term weight is computed based on binary approach.

3.4.2 The effect of context window to create history

Table 2 shows performance when context window (CW) size is set to CW = 3. The performances based on context shows an improvement in terms of precision, recall, and f-score over the performances based on only adding unique IDs in Table 1.

3.4.3 The effect of different term weighting approaches

Here, we show the performances of different weighting approaches with different cross-validation size.

Table 1 Performance measure with unique IDs decomposition using cross validation = 3

Category (C_k)	P (C_k)	R (C_k)	F_1 (C_k)
Female	0.9648	0.9714	0.9681
Male	0.8959	0.8741	0.8849
macro-avg (F_1^M)	0.9303	0.9228	0.9265
micro-avg(F_1^μ)	0.9501	0.9501	0.9501

3.4.3.1 Performance based on binary weighting approach Tables 3, 4, and 5 show the performances based on binary weighting approach which is stated in Eq. 1, where different cross-validation are taken into account to judge the performance of the dataset.

3.4.3.2 Performance based on normalized binary weighting approach Tables 6, 7, and 8 show the performances based on normalized binary weighting approach which is stated in Eq. 2, where different cross-validations are taken into account to judge the performance of the dataset.

3.4.3.3 Performance based on normalized term frequency-based approach Tables 9, 10, and 11 show the performances based on normalized term frequency-based (norm-TF) weighting approach which is stated in Eq. 3, where different cross-validation are taken into account to judge the performance of the dataset.

3.4.4 The effect of identical hierarchy

Table 12 shows performance when identical hierarchies are added to augment the session.

3.5 Discussions

All the results in Section 5.4 show that different combination of session augmentation approaches as well as different term weighting approaches can significantly improve the system performance. Using unique IDs decomposition in Table 1, we achieved 96.81% and 88.89% in terms of F-score in female and male category respectively and 92.65% and 95% in terms macro and micro F-score. Table 2 shows an improvement when we apply context window-based history generation approach. Table 2 achieved 97.35% and 90.46% in terms of F-score in female and male category respectively and 93.91% and 95.85% in terms macro and micro F-score. Further improvement is also seen when we apply different term weighting approaches. Table 12 shows a significant improvement using identical hierarchy. Since the female and male category session are quite unbalanced but it shows identical hierarchy can significantly improve the categorical performance. Figures 6, 7, and 8 show the performance of each fold in three-fold, five-fold, and ten-fold cross validation respectively and the performances are compared based on F-score. In Fig. 6, 15,000 training samples are randomly divided into three-fold where in each turn 10,000 is used for training and 5000 for test. In contrast for in terms of five-fold in Fig. 7, where in each turn 12,000 and 3000 samples are used for training and text respectively. Finally, in Fig. 8 for ten-fold cross validation, samples are randomly divided into ten-fold in and for each iteration 13,500 is used for training and the remaining 1500 is used as test data. Figures 6,

Table 2 Performance measure with context window using cross validation = 3

Category (C_k)	P (C_k)	R (C_k)	F_1 (C_k)
Female	0.9709	0.9760	0.9735
Male	0.9132	0.8963	0.9046
macro-avg(F_1^M)	0.9421	0.9361	0.93.91
micro-avg(F_1^μ)	0.9585	0.9585	0.95.84

Table 3 Results with binary approach with cross validation = 3

Category (Ck)	P (Ck)	R (Ck)	$F1$ (Ck)
Female	0.9709	0.9760	0.9735
Male	0.9132	0.8963	0.9046
macro-avg (F_1^M)	0.9420	0.9361	0.9390
micro-avg (F_1^μ)	0.9585	0.9585	0.9585

Table 4 Results with binary approach with cross validation = 5

Category (C_k)	P (C_k)	R (C_k)	F_1 (C_k)
Female	0.9714	0.9751	0.9733
Male	0.9105	0.8981	0.9043
macro-avg (F_1^M)	0.9410	0.9366	0.9388
micro-avg (F_1^μ)	0.9582	0.9582	0.9582

Table 5 Results with binary approach with cross validation = 10

Category (C_k)	P (C_k)	R (C_k)	F_1 (C_k)
Female	0.9709	0.9760	0.9735
Male	0.9132	0.8963	0.9046
macro-avg (F_1^M)	0.9420	0.9361	0.9390
micro-avg (F_1^μ)	0.9585	0.9585	0.9585

Table 6 Results with norm-binary approach with cross validation = 3

Category (C_k)	P (C_k)	R (C_k)	F_1 (C_k)
Female	0.9626	0.98464	0.9734
Male	0.9397	0.8644	0.9005
macro-avg (F_1^M)	0.9512	0.9244	0.9369
micro-avg (F_1^μ)	0.9580	0.9580	0.9580

Table 7 Results with norm-TF with cross validation = 5

Category (C_k)	P (C_k)	R (C_k)	F_1 (C_k)
Female	0.9637	0.9847	0.9741
Male	0.9412	0.8684	0.9033
macro-avg (F_1^M)	0.9524	0.9265	0.9387
micro-avg (F_1^μ)	0.9591	0.9591	0.9591

Table 8 Results with norm-TF with cross validation = 10

Category (C_k)	P (C_k)	R (C_k)	F_1 (C_k)
Female	0.9626	0.9844	0.9733
Male	0.9396	0.8641	0.9003
macro-avg (F_1^M)	0.9511	0.9242	0.9368
micro-avg (F_1^μ)	0.9579	0.9579	0.9579

Table 9 Results with norm-TF with cross validation = 3

Category (C_k)	P (C_k)	R (C_k)	$F_1(C_k)$
Female	0.9604	0.9643	0.9624
Male	0.8714	0.8590	0.8651
macro-avg (F_1^M)	0.9159	0.9116	0.9137
micro-avg (F_1^μ)	0.9411	0.9411	0.9411

Table 10 Results with norm-TF with cross validation = 5

Category (C_k)	P (C_k)	R (C_k)	$F_1(C_k)$
Female	0.9580	0.9656	0.9618
Male	0.8745	0.8496	0.8618
macro-avg (F_1^M)	0.9162	0.9076	0.9118
micro-avg (F_1^μ)	0.9401	0.9401	0.9401

Table 11 Results with norm-TF with cross validation = 10

Category (C_k)	P (C_k)	R (C_k)	$F_1(C_k)$
Female	0.9561	0.9695	0.9616
Male	0.8961	0.8555	0.8617
macro-avg (F_1^M)	0.9261	0.9125	0.9117
micro-avg (F_1^μ)	0.9427	0.9427	0.9399

Table 12 Performance measure with identical hierarchy using cross validation = 3

Category (C_k)	P (C_k)	R (C_k)	$F_1(C_k)$
Female	0.9694	0.9815	0.9754
Male	0.9314	0.8899	0.9102
macro-avg (F_1^M)	0.9503	0.9357	0.9427
micro-avg (F_1^μ)	0.9613	0.9613	0.9613

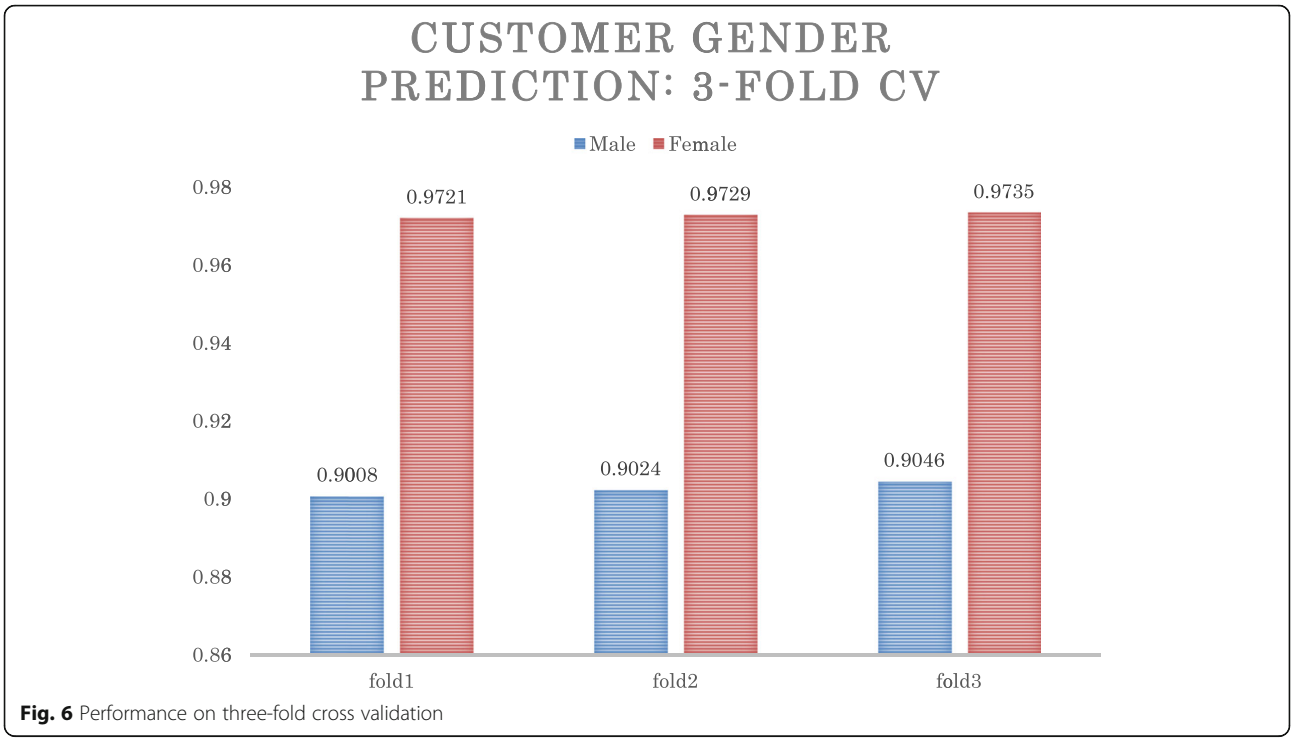

Fig. 6 Performance on three-fold cross validation

7, and 8 show that the results are very consistent even in different cross validation settings. It also shows that for each fold in three-fold to five-fold and then to ten-fold, the performances are a bit improved since five-fold and ten-fold have more training samples than three-fold.

4 Conclusions

In this implementation, we investigated the effectiveness of proposed session augmentation approaches, including unique IDs decomposition, context window, and identical hierarchy approaches with other different term weighing approaches using SVM classifier to address the ML-based gender prediction system. First, we implement a unique IDs decomposition-based session augmentation approach to create good features and conducted our experiments with this implementation. We then implemented a context window-based session augmentation approach to create history based on surrounding sessions. We then

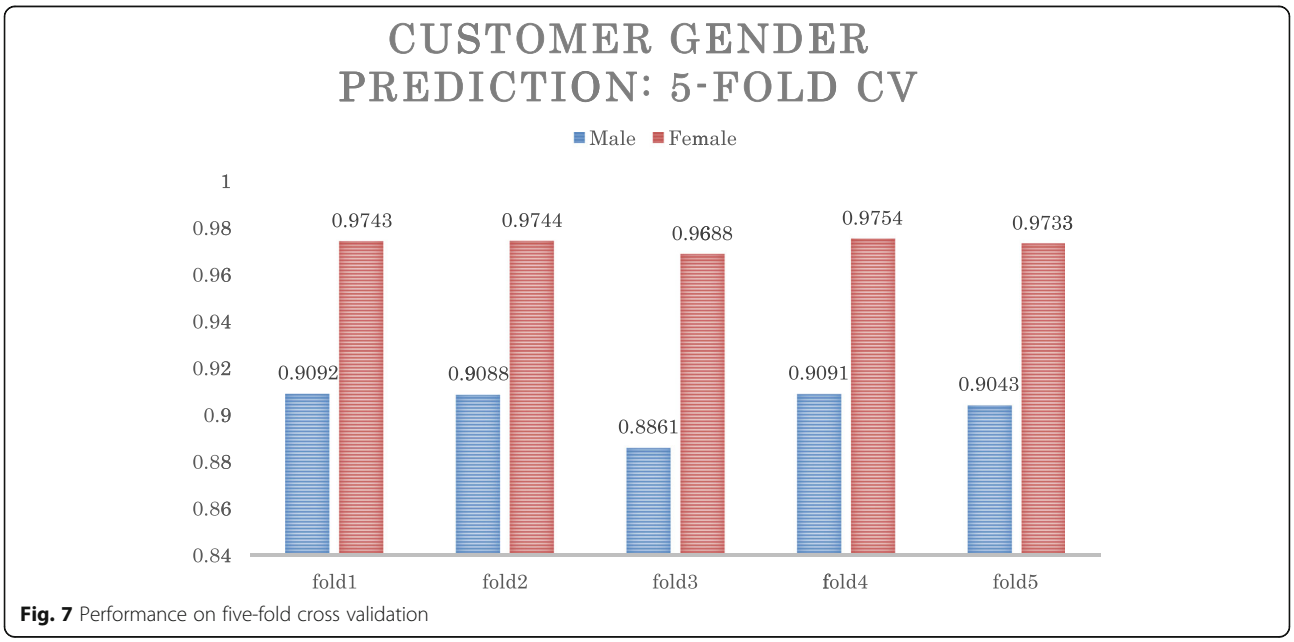

Fig. 7 Performance on five-fold cross validation

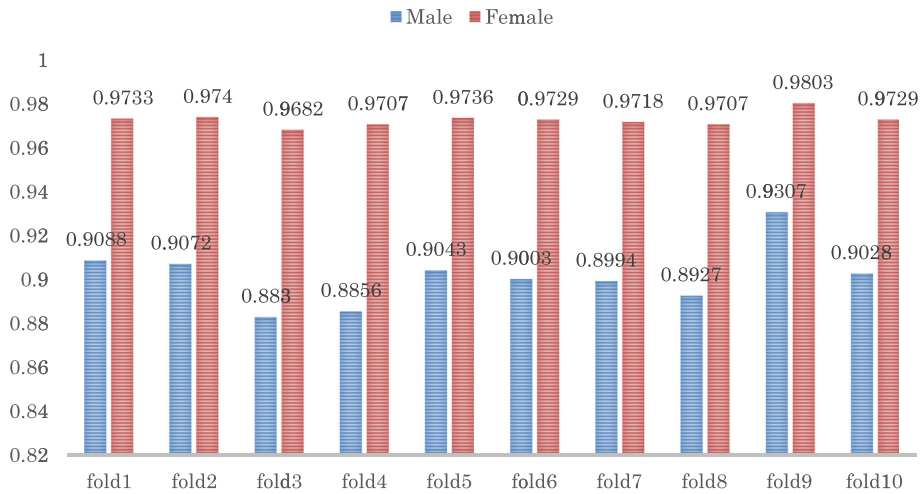

Fig. 8 Performance on ten-fold cross validation

incorporate this approach with unique IDs decomposition and improve the system performance. Finally, we introduced to extract identical hierarchy and that is further improve the system performance. Besides, the simple binary weighting approach also shows the effectiveness in classification task. Therefore, the combination of all session augmentation approach with SVM classifier can significantly improve the performance of gender prediction system.

Abbreviations

ATC: Automatic text classification; B2B2C: Business-to-business-to-customer; CW: Context window; FPT: Financing and promoting technology; IR: Information retrieval; LR: Logistic regression; ML: Machine learning; SML: Statistical machine learning; SVM: Support vector machine; TC: Text classification; TF.IDF.ICS$_\delta$F: Term frequency inverse document frequency incorporated with inverse class space density frequency; TFIDF: Term frequency inverse document frequency; VSM: Vector space model; WS: Window size

Acknowledgments

We thank the anonymous reviewers for their valuable comments.

Authors' contributions

MGS designed, implemented a set of different approaches and algorithms, carried out experiments and analysis to address the machine learning-based gender prediction systems, and drafted the manuscript. MMK carried out experiments to reproduce and analyze the results using cross-validation. MAY participated in research coordination. All the authors read and approved the final manuscript.

Funding

This research has been carried out with funding from AIRC/AIST and results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO). The corresponding author is funded by the NEDO project to design, implement, analysis, and in writing the manuscript.

Availability of data and materials

Dataset used for supporting the conclusions of this article are available from the public data repository at the website of https://knowledgepit.ml/pakdd15-data-mining-competition/. Dataset is also available upon request to corresponding author.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interest.

Author details

[1]Institute of Information Technology, Jahangirnagar University, Dhaka, Bangladesh. [2]National Institute of Advanced Industrial Science and Technology, 2-4-7 Aomi, Koto-ku, Tokyo 135-0064, Japan.

Received: 4 July 2019 Accepted: 9 January 2020
Published online: 17 March 2020

References

1. Debole F, Sebastiani F (2003) Supervised term weighting for automated text categorization. In Proceedings of the 2003 ACM symposium on applied computing, pp 784–788
2. Salton G, McGill MJ (1983) Introduction to modern information retrieval
3. Sohrab MG, Ren F (2012) Class-indexing: the effectiveness of class space density in high and low-dimensional vector space for text classification. In: 2nd International Conference of IEEE CCIS, pp 2034–2042
4. Sohrab MG, Ren F (2013) Class-indexing-based term weighting for automatic text classification. Inform Sci 236:109–125
5. Flora S, Agus T (2011) Experiments in term weighting for novelty mining. Expert Syst Appl 38(11):14094–14101
6. Kansheng S, Jie H, Hai-tao L, Nai-tong Z, Wen-tao S (2011) Efficient text classification method based on improved term reduction and term weighting. J China Univ Posts Telecomm 18(1):131–135
7. Ko Y, Seo J (2009) Text classification from unlabeled documents with bootstrapping and feature projection techniques. Inform Processing Manag 45(1):70–83
8. Fuhr N, Buckley C (1991) A probabilistic learning approach for document indexing. In: Information Sciences, vol 9, pp 223–248
9. Liu Y, Loh H, Sun A (2009) Imbalanced text classification: a term weighting approach. Expert Systems with Applications 36:690–701

10. Salton G (1975) A theory of indexing
11. Salton G, Wong A, Yang CS (1975) A vector space model for automatic indexing, by. Assoc Comp Mach 18:613–620
12. Joachims T (2001) A statistical learning model of text classification for support vector machines. In: SIGIR-2001: Proceedings of the 24th annual international ACM SIGIR conference on Research and Development in Information Retrieval, pp 128–136
13. Jones KS (1972) A statistical interpretation of term specificity and its application in retrieval. J Document 28:11–21
14. Korfhage RR (1997) Information storage and retrieval. Wiley
15. Luo Q, Chen E, Xiong H (2011) A semantic term weighting scheme for text classification. Exp Syst Appl 38(10):12708–12716
16. Singhal A (2001) Modern information retrieval: a brief overview, by. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering 24: 35–43
17. Lewis DD, Ringuette M (1994) A Comparison of two learning algorithms for text categorization. In: Proceedings of the third annual symposium on document analysis and information retrieval, pp 81–93
18. Godbole S, Sarawagi S, Chakrabarti S (2002) Scaling multi-class support vector machine using inter-class confusion. In: In Proceedings of the 8th ACM international conference on knowledge discovery and data mining, pp 513–518
19. Joachims T (1998) Text categorization with support vector machines: learning with many relevant features. In: Proceedings of 10th European conference on machine learning. Springer Verlag, Heidelberg, pp 137–142
20. Lewis DD et al (1996) Training algorithms for linear text classifiers. In: In Proceedings of 19th ACM International Conference on Research and Development in Information Retrieval
21. Schutze H, Hull D, Pedersen JO (1995) A comparison of classifiers and document representations for the routing problem. In: In Proceedings of the 18th ACM International Conference on Research and Development in Information Retrieval, pp 613–620
22. Davil L (2008) Advanced data mining techniques, Olsen, and Dursun Delen
23. James G, Witten D, Hastie T, Tibshirani R (2013) Introduction to statistical learning, with Applications in R
24. Han J, Kamber M, Pei J. Data Mining: Concepts and Techniques, Third Edition (The Morgan Kaufmann Series in Data Management Systems). 2012.
25. Wu X, Kumar V et al (2008) Top 10 algorithms in data mining. Knowledge Inform Syst 14:1–37
26. Yang Y (1999) An evaluation of statistical approaches to text categorization. J Inform Retr 1(1/2):67–88

Publisher's Note