Persian sentiment analysis of an online store independent of pre-processing using convolutional neural network with fastText embeddings

Sajjad Shumaly [1] , **Mohsen Yazdinejad** [2] , **Yanhui Guo** [Corresp. 3]

[1] Industrial Engineering, Sharif University of Technology, Tehran, Iran

[2] Computer Engineering, Isfahan University, Isfahan, Iran

[3] Computer Science, University of Illinois at Springfield, Springfield, United States

Corresponding Author: Yanhui Guo
Email address: yguo56@uis.edu

Sentiment analysis plays a key role in companies, especially stores, and increasing the accuracy in determining customers' opinions about products assists to maintain their competitive conditions. We intend to analyze the users' opinions on the website of the most immense online store in Iran; Digikala. However, the Persian language is unstructured which makes the pre-processing stage very difficult and it is the main problem of sentiment analysis in Persian. What exacerbates this problem is the lack of available libraries for Persian pre-processing, while most libraries focus on English. To tackle this, approximately 3 million reviews have been gathered in Persian from the Digikala website using web-mining techniques and have been used the fastText method to create a word embedding. It has been assumed it would dramatically cut down the need for the text pre-processing through the skip-gram method considering the position of the words in the sentence and the words' relations to each other. Another word embedding has been created using the TF-IDF in parallel with fastText to compare their performance. In addition, the results of the CNN, BiLSTM, Logistic Regression, and Naïve Bayes models have been compared. As a significant result, we obtained 0.996 AUC, and 0.956 F-score using fastText and CNN. In this article, not only it has been demonstrated to what extent it is possible to be independent of pre-processing but also the accuracy obtained is also better than other researches done in Persian. Avoiding complex text preprocessing is also important for other languages since most text preprocessing algorithms have been developed for English and cannot be used for other languages. The created word embedding due to its high accuracy and independence of pre-processing has other applications in Persian besides sentiment analysis.

Persian Sentiment Analysis of an Online Store independent of pre-processing using Convolutional Neural Network with fastText Embeddings

Sajjad Shumaly[1], Mohsen Yazdinejad[2], Yanhui Guo[3]

[1] Industrial Engineering, Sharif University of Technology, Tehran, Iran
[2] Computer Engineering, Isfahan University, Isfahan, Iran
[3] Computer Science, University of Illinois at Springfield, Springfield, United States

Corresponding Author:
Yanhui Guo
The University of Illinois, Springfield, United States
Email address: yguo56@uis.edu

Abstract

Sentiment analysis plays a key role in companies, especially stores, and increasing the accuracy in determining customers' opinions about products assists to maintain their competitive conditions. We intend to analyze the users' opinions on the website of the most immense online store in Iran; Digikala. However, the Persian language is unstructured which makes the pre-processing stage very difficult and it is the main problem of sentiment analysis in Persian. What exacerbates this problem is the lack of available libraries for Persian pre-processing, while most libraries focus on English. To tackle this, approximately 3 million reviews have been gathered in Persian from the Digikala website using web-mining techniques and have been used the fastText method to create a word embedding. It has been assumed it would dramatically cut down the need for the text pre-processing through the skip-gram method considering the position of the words in the sentence and the words' relations to each other. Another word embedding has been created using the TF-IDF in parallel with fastText to compare their performance. In addition, the results of the CNN, BiLSTM, Logistic Regression, and Naïve Bayes models have been compared. As a significant result, we obtained 0.996 AUC, and 0.956 F-score using fastText and CNN. In this article, not only it has been demonstrated to what extent it is possible to be independent of pre-processing but also the accuracy obtained is also better than other researches done in Persian. Avoiding complex text preprocessing is also important for other languages since most text preprocessing algorithms have been developed for English and cannot be used for other languages. The created word embedding due to its high accuracy and independence of pre-processing has other applications in Persian besides sentiment analysis.

1. Introduction

With the advancement of technology and the spread of the Internet, proper conditions have been provided for the online store's activities. Due to some advantages such as high variety, delivery speed, and time savings, the customers of this type of store are constantly increasing (Liang and Wang 2019). When buying from online stores, due to the gap between the buyer and the product, there may be some problems such as poor quality of products, inadequate after-sales service, or inconsistency between product descriptions and performance (Ji, Zhang, and Wang 2019). One of the viable solutions to overcome the problems is to use the opinion of users who have already purchased the product.

In the past, if people needed to know the other's opinion, they would ask questions of family, friends, or relatives. Similarly, companies and stores used surveys to find out the opinions of people or customers. But today, if people require to buy or companies and stores need to know the opinions of customers to provide better services and products, they can easily refer to people's comments and discussions on online store websites or forums. Therefore, online reviews are important sources of information about the quality of goods that play a key role in customer awareness of products (X. Li, Wu, and Mai 2019). Online reviews enable the customer to have a comprehensive view of the products and their alternatives before making a purchase, thus, it has a significant impact on the extent of product sales (Hu, Liu, and Zhang 2008). As a matter of fact, the immediate response of stores to their customers' complaints is essential in maintaining their competitive position. But analyzing these reviews manually is quite time-consuming and costly. Also, automatic comment analysis has some obstacles, problems such as using sentences with incorrect grammar, using slang terms, and not following the correct punctuation are an integral part of making text analysis difficult (Irfan et al. 2015). When it comes to resolving these problems, sentiment analysis techniques play an essential role. These techniques automatically estimate customer sentiment into positive, negative, and even neutral classes. Therefore, sentiment analysis for online stores is highly valued because it can extract users' sense of goods and help to make decisions to increase customer satisfaction and product sales. Sentiment analysis can be considered as a type of content analysis that specifically seeks to determine the emotional tone of the text (Oscar et al. 2017). This is done based on the emotional evidence between words and phrases (Tausczik and Pennebaker 2010).

In this article, we are seeking to analyze the feelings of customer reviews on the website of the largest and well-known online store in Iran (Digikala). At first, lingual problems were taken into account as a significant challenge. There are several problems in Persian text pre-processing such as using slang, using letters of other languages especially Arabic, lack of a clear boundary between phrases. To tackle the problems, we employed fastText and skip-gram because we wanted to examine whether the utilize of the methods capable of reducing the need for data pre-processing and make language processing easier. In the following, we will inspect this assumption and compare the obtained results with other algorithms and other reports. Another severe limitation was that the deep learning models required an immense dataset, but most of the available datasets in Persian are small to such an extent that they cannot be employed in deep models. Thus, a rich

79 and immense dataset had to be extracted from the Digikala website which was conducted by web-
80 mining methods. It should be noted that this article seeks to achieve the following goals:

81 • Investigating the reduction of the need for text pre-processing by implementing methods
82 such as fastText and skip-gram, either in Persian language processing or others.

83 • Gathering comprehensive customer reviews dataset based on various types of digital goods
84 to create a general word embedding for a various range of works related to digital goods.

85 • Sentiment analysis of Digikala website's reviews with high accuracy even compared to
86 other researches.

87

2. Related Works

89 Sentiment analysis methods are divided into three general categories Lexicon based, traditional
90 Machine Learning, and Deep Learning models (Yadav and Vishwakarma 2020). The first category
91 is the sentiment analysis using a sentiment lexicon and it is an unsupervised method. In this case,
92 emotional similarities of words and phrases are used and its accuracy is highly dependent on pre-
93 learned weights (Taboada et al. 2011). This method collects a set of pre-compiled sentiment words,
94 terms, phrases, and idioms with a specific thematic category such as opinion finder lexicon
95 (Wilson et al. 2005) and ontologies (Kontopoulos et al. 2013).

96 The second category is based on machine learning methods which are divided into supervised and
97 unsupervised categories. The accuracy of these methods is strongly influenced by the extracted
98 features from the text. Supervised techniques such as Naïve Bayes, SVM, Maximum Entropy, and
99 Logistic Regression are the most common techniques in this field (Ye, Zhang, and Law
100 2009)(Montejo-Ráez et al. 2014). However, unsupervised methods are suitable for situations
101 where labeling for the dataset is impossible (Paltoglou and Thelwall 2012).

102 Deep learning has grown and been used in many areas in the last decade, for example in the field
103 of object recognition (Ghoreyshi, AkhavanPour, and Bossaghzadeh 2020)(Ali et al. 2020), speech
104 recognition (Deng, Hinton, and Kingsbury 2013)(H. Li, Baucom, and Georgiou 2020), anomaly
105 detection (Zhao et al. 2018), feature extraction (Lin, Nie, and Ma 2017)(Rajaraman et al. 2018),
106 auto-encoding (Pu et al. 2016). Also, in cases where deep learning along with machine learning
107 has been used for text analysis and sentiment analysis, good results have been obtained (Tang,
108 Qin, and Liu 2015)(Severyn and Moschitti 2015). The main difference between sentiment analysis
109 by deep learning and other methods is how to extract the features. To be specific, one of the
110 advantages of deep learning models is that there is no need for user intervention in feature
111 extraction, which of course requires a large amount of data to perform the feature extraction
112 operation. Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), Long Short-
113 Term Memory (LSTM), and Gated Recurrent Unit (GRU) are the most common models of deep
114 learning in sentiment analysis (Zhang, Wang, and Liu 2018).

115 The most basic and widely used CNN model for sentiment analysis at the sentence level is the one
116 presented by Kim (Kim 2014). Then, Zhang and Wallace proposed a special single-layer CNN
117 architecture by examining improvements made by changing the model configuration (Zhang and
118 Wallace 2015). Many developments have been made to improve the performance of CNN-based

119 sentiment analysis models. In this regard, an example of CNN combined with fuzzy logic called
120 the Fuzzy Convolutional Neural Network (FCNN) (Nguyen, Kavuri, and Lee 2018) is noteworthy.
121 The use of CNN in natural language processing is now a common topic and much research is being
122 done in this area (Wehrmann et al. 2017)(Gan et al. 2020)(Arora and Kansal 2019).
123 Deep neural networks are difficult to train because they often suffer from the problem of vanishing
124 gradients. LSTM architecture was introduced (Hochreiter and Schmidhuber 1997) to overcome
125 this shortcoming to learn long-term dependencies. After the original work, the LSTM has
126 experienced several improvements such as adding forget gate (Gers 1999). A neural network
127 architecture could not be so great adopted into practice without strong theoretical support,
128 therefore, a widespread review considering the several LSTM variants and their performances
129 relative to the so-called vanilla LSTM model was conducted by Greff et al. (Greff et al. 2017). The
130 vanilla LSTM model is interpreted as the primary LSTM block with the addition of the forget-gate
131 and peephole connections. Also, to overcome some limitations in conventional RNN models,
132 bidirectional RNN (BRNN) models were proposed. Using this model's structure, both future and
133 past situations of sequential inputs in a time frame are evaluated without delay (Schuster and
134 Paliwal 1997). By combining the ideas of BRNN and LSTM it is possible to achieve Bidirectional
135 LSTM (BiLSTM) which has better performance than LSTM in classification processes. With the
136 development of LSTM in recent years, it has been used in projects such as Google Translate and
137 Amazon Alexa (Wu et al. 2016)(Vogels 2016) in natural language processing.
138

139 3. Materials and methods

140 All the taken steps, methods, codes, and results that are presented below, along with a part of the
141 extracted dataset are fully accessible on the repository (Yazdinejad and Shumaly 2020).
142

143 3.1. Dataset

144 Having access to a large dataset with richness and content integrity is indispensable to train a deep
145 model. Most available datasets to train a deep model and sentiment analysis are in English. To
146 collect a rich dataset, web-mining methods were used and the reviews on the Digikala website
147 were extracted which were in Persian. Posted reviews by buyers express their level of satisfaction
148 with their purchase and product features. After submitting their reviews, buyers could choose
149 between the "I suggest" and "I do not suggest" options. These two options were extracted and used
150 in the model as labels for the problem of sentiment analysis. Our goal was to analyze the opinions
151 of users of the Digikala website, so we extracted the data of the section related to digital goods
152 using web-mining libraries such as the Beautiful Soup (Richardson 2020). Beautiful Soup is a
153 Python package to parse XML and HTML documents and it is useful for web scraping (Hajba
154 2018).

155

156 3.2. Pre-processing

157 One of the first steps in natural language processing problems has always been data pre-processing.
158 At this stage, the texts need to be cleaned and prepared to begin the analysis. In Persian, this stage
159 is even more difficult and important because it has its complexities. This field has attracted many
160 researchers in the last decade, therefore, libraries and algorithms in the field of pre-processing in
161 Persian have been developed (Mohtaj et al. 2018)(Nourian 2013) which have become more
162 complete and better over time. However, these algorithms cannot work as well as similar
163 algorithms in English and need further development. We are seeking a way to achieve an accurate
164 result by avoiding the complications of data pre-processing steps in Persian. Regular expressions
165 are used for data pre-processing in all of the following steps using the "re" library (Rachum 2020)
166 in python. Pre-developed libraries for the Persian language have not been used to perform data
167 pre-processing steps and we assume that the use of fastText and skip-gram in creating word
168 embedding reduces the need for complex pre-processing.
169

3.2.1. Normalization

171 In Persian, some letters are not unique and may have different alternatives to other languages such
172 as Arabic. For example, the letter "ى" is Persian, but the letter "ي" is Arabic, and these two letters
173 will likely be used interchangeably. This causes the created words to be considered as two different
174 words. In this way, they may be considered separately in the calculations and a vector can be drawn
175 for each with its characteristics. To solve this issue, it is necessary to use the standard form for all
176 available texts.
177

3.2.2. Tokenization

179 Tokenization is a stage in which attempts are made to divide sentences into meaningful words and
180 phrases that can be considered as a suitable input for the next steps. The main challenge of the
181 Persian language at this stage is that sometimes there are no clear boundaries between phrases as
182 a result of three different modes of spacing in Persian. In other words, phrases in Persian can be
183 without space, with half-space, or with space, which is often mistakenly used instead of each other.
184 For instance, the words "نرم افزار" and "نرمافزار", which both mean software, are written with both
185 space and half-space forms. If the wrong form is used, the phrase border will be mistakenly
186 recognized as two separate words "نرم" and "افزار". Vice versa, phrases that consist of several words
187 can be considered as one word due to a mistake in using space. For example, the word "از مسیر
188 دیگر", which means "from another path", maybe written as "ازمسیردیگر" without any spaces. These
189 kinds of mistakes blur the line between phrases and words and make it difficult to pre-process.
190

3.2.3. Stemming

192 The stemming process seeks to remove part of the word in such a way that the root of the word is
193 determined (Willett 2006). The root of the word does not necessarily mean the dictionary root of
194 the word and is acceptable in cases where it can improve the performance of the model. For

195 example, we can refer to the phrase "رنگهایشان". In this phrase, "رنگ" means color, and "ها" is used
196 to represent plural and "یشان" for determination of ownership. A significant number of stemming
197 algorithms use the following rule (Mohtaj et al. 2018):
198   (possessive suffix)(plural suffix)(other suffixes)(stem)(prefixes)
199 Stemming is a rule-based process that is usually done by removing suffixes and prefixes.
200 Consequently, it cannot be used in other languages and each language requires its algorithms.
201 Stemming algorithms are being developed in Persian but due to the complexity of the Persian
202 language, their performance needs to be improved.
203

3.3. Pseudo labeling

205 In classification problems, it is a common problem that a large number of samples do not have
206 labels and therefore cannot be used in model training. Techniques such as pseudo-labeling can be
207 used to overcome this problem and determine the labels of some samples (Lee 2013). The first
208 step in pseudo-labeling is to develop a model based on labeled samples in the dataset that is in
209 charge of determining the label of unlabeled samples. Only labels that have been estimated with
210 high confidence are accepted. In the next step, another model is developed using training data
211 along with the new labeled data which can be used to predict test data with higher accuracy. In this
212 way, 104.8 thousand Negative Feedback reviews and 30.5 thousand Positive Feedback reviews
213 were labeled and could be used in the dataset for subsequent analysis. As will be shown in the
214 results section, this method had a significant impact on improving accuracy.
215

3.4. Data balancing

217 Unequal distribution of data in different classes in a classification problem leads to data imbalance.
218 The class with the most data is called the majority class, and the class with the least data is called
219 the minority class. In these cases, the models tend to ignore the minority class and predict in favor
220 of the majority class. Many machine learning models, such as Support Vector Machine, Naïve
221 Bayes, Decision Tree, and Artificial Neural Network cannot have good results in this situation
222 (Díez-Pastor et al. 2015)(Vorraboot et al. 2015). In general, data balancing solutions can be divided
223 into two categories; over-sampling and under-sampling. The goal of both solutions is to
224 approximate the number of data distributed in the minority and majority classes. In over-sampling,
225 this is done by increasing the amount of data in the minority class, and in under-sampling by
226 reducing the amount of data in the majority class. In the present problem, we used the random
227 oversampling method to balance the dataset.
228

3.5. Feature Extraction
3.5.1.fastText

231   Neural network-based methods have become very popular in the field of natural language
232   processing due to their accuracy. However, most of these methods are slow to analyze large
233   datasets and they need to receive word embedding to analyze texts. For this reason, a method called
234   fastText has been proposed (Joulin et al. 2016). fastText is an efficient, fast, and open-source
235   model that Facebook has recently released. In fastText, a set of tricks has been used to improve
236   the processing speed and performance of the model, one of which is skip-gram. Data sparsity has
237   always been one of the biggest problems in natural language analysis. In other words, the main
238   problem of modern language processing is that language is a system of rare events, so varied and
239   complex, that we can never model all possibilities (Preethi Krishna and Sharada 2020). Therefore,
240   skip-gram allows some words to be skipped and non-adjacent words to be examined together.
241   Mikolov et al (Mikolov et al. 2013) found the skip-gram model to be superior to the bag-of-word
242   model in a semantic-syntactic analogy task. Skip-gram is popular, easy to implement, and it is
243   proven and reliable (Gurunath Shivakumar and Georgiou 2019). Accordingly, in this article, word
244   embeddings have been provided using fastText and skip-gram to investigate the reduction of
245   language processing dependence on data-preprocessing.
246

247   ## 3.6.   Sentiment analysis model
248   ### 3.6.1. Convolution neural network

249   Using CNN has shown high accurate results based on studies in English texts (Nedjah, Santos, and
250   de Macedo Mourelle 2019). This model can receive and analyze word embedding as input instead
251   of images, which are also effective in this area (Kim 2014). Each row of the input matrix represents
252   a word. Figure 1 shows the architecture of a CNN model used for the NLP classification problem
253   (Zhang and Wallace 2015). This figure shows how a CNN model treats a 6-word sentence. The
254   matrix formed for this sentence is analyzed by 6 different convolution filters and converted to
255   maps of attributes with dimensions of 1x4, 1x5, and 1x6. Finally, the pooling operation is
256   performed on the maps and their outputs are merged to create a unique vector that can be used as
257   input for the SoftMax layer and determine the class. The CNN model used in this article is based
258   on the mentioned model and its architecture is shown in table 1.
259

260   ### 3.6.2. Bidirectional Long Short-Term Memory (BiLSTM)

261   Another deep model used to solve the problem is BiLSTM. The LSTM model can decide which
262   information is useful and should be preserved and which information can be deleted based on the
263   dataset it has trained with. The LSTM has been widely used in NLP such as long document
264   categorization and sentiment analysis (Rao et al. 2018). Figure 2 is a demonstration of an LSTM
265   cell used in this article, which has an input layer, an output layer and a forget layer (Gers 1999).
266   Based on the figure, the LSTM cell mathematically expressed as follows:

267
$$f_t = \sigma(W_{fh}h_{t-1} + W_{fx}x_t + b_f) \tag{1}$$

268
$$i_t = \sigma(W_{ih}h_{t-1} + W_{ix}x_t + b_i) \tag{2}$$

269 $$\tilde{c}_t = tanh(W_{\tilde{c}h}h_{t-1} + W_{\tilde{c}x}x_t + b_{\tilde{c}}) \tag{3}$$

270 $$c_t = f_t.c_{t-1} + i_t.\tilde{c}_t \tag{4}$$

271 $$o_t = \sigma(W_{oh}h_{t-1} + W_{ox}x_t + b_o) \tag{5}$$

272 $$h_t = o_t.tanh(c_t) \tag{6}$$

273

274 where x_t denotes the input; h_{t-1} ,and h_t denote the output of the last LSTM unit and current
275 output; c_{t-1}, and c_t denote memory from the last LSTM unit and cell state; f_t denotes forget gate
276 value; W_i, $W_{\tilde{c}}$, and W_o are the weights; b is the bias; the operator '·' denotes the pointwise
277 multiplication of two vectors. In LSTM, the input gate can decide what new information can be
278 stored in the cell state, also the output gate can decide what information can be output based on
279 the cell state. By combining the ideas of BRNN and LSTM it is possible to achieve Bidirectional
280 LSTM (BiLSTM) which has better performance than LSTM in classification processes especially
281 in speech processing tasks (Graves and Schmidhuber 2005). Therefore, this article uses the
282 BiLSTM structure, and figure 3 is shown a basic structure of the BiLSTM network (Yildirim
283 2018). The BiLSTM model used in this article architecture is shown in table 2.
284

285 3.7. Evaluation

286 Due to imbalanced data, indicators such as accuracy is not appropriate for this study. Because the
287 developed model in the face of this type of data tends to ignore the minority class and can still be
288 accurate. For this purpose, AUC and F-score indexes will be used, which are good choices for
289 problems dealing with imbalanced data (Sokolova, Japkowicz, and Szpakowicz 2006). AUC
290 indicates the area below the diagram in the ROC curve, and the ROC curve is a method for judging
291 the performance of a two-class classifier (Luo et al. 2020). In the ROC curve, the vertical axis is
292 the TPR (represents the true positive rate), Also, the horizontal axis is FPR (represents the false
293 positive rate).

294 $$FPR = \frac{fp}{tn + fp} \tag{7}$$

295 $$TPR = \frac{fp}{tp + fn} \tag{8}$$

296   - TP: Positive samples are classified as positive
297   - FN: Positive samples are classified as negative
298   - TN: Negative samples are classified as negative
299   - FP: Negative samples are classified as positive
300

301 The F-score is the harmonic mean of precision and recall (Velupillai et al. 2009) and represents a
302 weighted average of precision and recall (Gacesa, Barlow, and Long 2016). This index has wide
303 applications in natural language processing (Derczynski 2016), and like the AUC, it can be used
304 in problems involved with imbalanced data.

305 $$Precision = \frac{tp}{tp + fp} \tag{9}$$

306
$$Recall = \frac{tp}{tp + fn} \qquad\qquad (10)$$

307
$$F - score = 2 \times \frac{Precision \ \times \ Recall}{Precision \ + \ Recall} \qquad\qquad (11)$$

308

309   All the steps mentioned in the methodology section can be summarized in figure 4.

310

4. Results and discussion

4.1. Dataset

313 The digital goods' reviews of the Digikala website were extracted, which are a total of 2,932,747
314 reviews. Figure 5 shows the frequency of comments per category. Examining the comments of
315 different product categories can increase the comprehensiveness of the model. To be specific, the
316 words, phrases, and sentences are different in reviews of the products in the different categories,
317 and considering various types of product categories will improve the generalization of the model
318 in different situations. Table 3 shows the general structure of the collected dataset. In this table,
319 the "Comment ID" column stores the unique values for each comment, the "Original Comment"
320 column is the original of the comments written in Persian, the "Translated Comment" column is a
321 translation of the "Original Comment" column into English. The "Translated Comment" column
322 is used only to increase readability in the table and does not exist in the dataset. In the "Negative
323 Feedback" column, if the value is 1, means that the user is not advised to buy the product, and in
324 the "Positive Feedback" column, if the value is 1, it means the user is advised to buy the product,
325 and the "Cat. Name" column represents the product category for which the comment was written.

326 The positive point of this website is that the buyers after submitting their comments can choose an
327 option that states whether they generally recommend the product to others or not. Therefore, a
328 significant number of extracted reviews are labeled. In other words, 308,122 of the reviews in the
329 dataset do not recommend purchased items to others and the "Negative Feedback" column of these
330 reviews in the dataset shows the number one. Likewise, 1,749,055 of the reviews in the dataset
331 recommend the purchased items to others, and the "Positive Feedback" column of these comments
332 in the dataset shows the number one. A significant part of the reviews is without labels and the
333 reviews with labels are also imbalanced and these problems must be addressed in some ways.

334

4.2. Pre-processing

336 During the initial review of the comments, the first correction that had to be made was the removal
337 of escape sequences. An escape sequence is a series of two or more characters starting with a
338 backslash and when used in a string, are not displayed as they are written. In most reviews, there
339 were some escape sequences such as "\n" and "\t" that needed to be removed. Also, sometimes
340 users wrote some URLs to link to personal content that had to be removed. At this stage, all Persian
341 numbers were converted to English, and letters that had different alternatives were standardized to

342 normalize the text. Then all the phrases were tokenized by defining the word boundary and
343 converting the half-space to space. In the stemming stage, prefixes and suffixes used were
344 removed.

345 After the pre-processing steps, the number of words in the Positive Feedback class was 6.1 million
346 and the number of words in the Negative Feedback class was 34.1 million. Using the word cloud
347 diagram, the most repetitive words in each of the classes can be depicted. Figure 6 and figure 7
348 show the repetitive words in the Positive Feedback and Negative Feedback classes, respectively.
349 Words like "I gave back", "bad" and "I do not recommend" can be seen in the Negative Feedback
350 figure, and words like "I'm satisfied", "Appropriate" and "Speed" can be seen in the Positive
351 Feedback figure.

352

353 4.3. Sentiment analysis

354 Data balancing is a crucial step that can increase accuracy. The random over-sampling method was
355 used to balance the data. In other words, some data with the label of "Negative Feedback" were
356 randomly selected and repeated. As a matter of fact, one of the common mistakes in this section is
357 to apply the balancing method to the entire data which leads to errors in estimating the indicators.
358 In these cases, the indicators are in a better position than the model capability and the results are
359 reported incorrectly well. To avoid this, the balancing method was used only for the training data.
360 After using the pseudo-labeling method, the number of positive feedbacks was about 1.8 million
361 and the number of negative feedbacks was about 400 thousand. In this way, the negative feedbacks
362 were repeated randomly about four times to balance the dataset.

363 The stratified K-fold cross-validation method is used to perform the evaluation. It is a method for
364 model evaluation that determines how independent the results of statistical analysis on a dataset
365 are from training data. In K-fold cross-validation, the dataset is subdivided into a K subset and
366 each time one subset is used for validation and the other K-1 is used for training. This procedure
367 is repeated K times and all data is used exactly once for validation. The average result of this K
368 computing is selected as a final estimate. Stratified sampling is the process of dividing members
369 of a dataset into similar subsets before sampling and this type of data sampling was selected due
370 to imbalanced data. Using the stratified K-fold cross-validation method, we expect the values of
371 the indicators to be more real. In all stages of measuring the accuracy of the model, K was
372 considered equal to 5.

373 As stated in the methodology, TF-IDF and fastText methods were used to extract the features. The
374 BiLSTM and CNN models used the fastText output, and the Naïve Bayes and Logistics Regression
375 models used the TF-IDF output, and their accuracy was finally compared with each other in table
376 4. According to this table, the results of BiLSTM and CNN models are more accurate than others
377 and CNN has given the best results.

378 As expected, due to the use of fastText and skip-gram methods, the need for data pre-processing
379 has been reduced. In other words, stemming and normalization methods have not affected the final
380 result. To examine this more closely, we chose the CNN model as the best model and we once

381 performed the sentiment analysis process using the pre-processing steps and again without these
382 steps. The AUC and F-score were 0.9943 and 0.9291 before pre-processing, and 0.9944 and 0.9288
383 after pre-processing. The results can be seen in table 5. In the table, the meaning of the "before
384 preprocessing" is just before the stemming and normalization steps. In other words, the methods
385 used to create word embedding can depict the same words in the same range of spaces without the
386 need to standardize letters and also without the need to identify the original root of words.

387 To implement pseudo-labeling, we developed a model that can estimate labels for unlabeled
388 reviews using fastText and CNN models. After estimating all the labels, those with more than 90%
389 probability for the Negative Feedback class and less than 1×10^{-7} for the Positive Feedback class
390 were selected. Therefore, 104.8 thousand Negative Feedback reviews and 30.5 thousand Positive
391 Feedback reviews were labeled and could be used in the dataset for subsequent analysis. In using
392 the pseudo-labeling technique, most of our focus was on Negative Feedback as a minority class,
393 which also leads to balance the dataset as much as possible. In this way, a significant amount of
394 unlabeled data that had been excluded from the sentiment analysis process was re-entered into the
395 process and helped to increase the accuracy and generalizability of the model.

396 Contrariwise of pre-processing, the use of the pseudo-labeling method significantly improved the
397 results. After using pseudo-labeling, the values of AUC and F-score improved to 0.996 and 0.956.
398 The values of the three mentioned states can be seen based on different folds in table 5. Figure 8
399 also shows the ROC curve for all three states.

400 The suggested model has had better results than the previous models which have used pre-
401 processing methods in Persian sentiment analysis. For instance, some researchers introduced pre-
402 processing algorithms and succeed to enhance the results of machine learning algorithms (Saraee
403 and Bagheri 2013). In the research, the F-score of the proposed pre-processing algorithms
404 employing Naïve Bayes as a classifier algorithm is 0.878. In another research, the various
405 alternatives for pre-processing and classifier algorithms were examined and the best result was
406 assisted with an SVM classifier by 0.915 F-score value (Asgarian, Kahani, and Sharifi 2018). Also,
407 some researches were attempted to utilize state-of-the-art deep models in such a way to reduce
408 dependency on pre-processing and avoiding complex steps (Roshanfekr, Khadivi, and Rahmati
409 2017). The F-score of the BiLSTM and CNN algorithms in the research is 0.532 and 0.534. All
410 mentioned article's focus was on the digital goods reviews in Persian two-class sentiment analysis
411 as same as this article. A comparison of the results in this paper with other researches and other
412 common algorithms indicates that not only the dependence on data pre-processing has been
413 eliminated but also the accuracy has increased significantly.

414 The result reveals that it is quite possible to create independent models from the pre-processing
415 process using the method of fastText and skip-gram. Moreover, BiLSTM and CNN methods can
416 have significant results. However, all of the mentioned methods need to have an immense dataset.
417 To prove this, It is noteworthy that the use of the pseudo-labeling method because of increasing
418 training data has a great impact on the result. Independency from pre-processing steps is not related
419 to the Persian language. The results are reachable for other languages using sufficient labeled
420 samples and the mentioned methods in this article.

421

5. Conclusion

423 The dataset included approximately 3 million reviews was extracted from the digital goods section
424 of the Digikala website as the largest online store in Iran. Basic pre-processing methods were used
425 to modify the words and tokenize them. Due to the lack of labels for a large part of the dataset, the
426 pseudo-labeling method was employed which improved the results. Data balancing was also
427 performed using random over-sampling. Persian data pre-processing was found difficult, so the
428 fastText method was conducted to reduce the need for data pre-processing and word embedding
429 development. The embeddings were employed as the input to the BiLSTM, and CNN models.
430 Using the suggested model, not only the obtained results have been very desirable and are much
431 more accurate in Persian compared to other reports but also there are no complications related to
432 data pre-processing. The effect of stemming and normalization on the output was evaluated and
433 revealed that the proposed method is not dependent on data pre-processing. Eventually, Besides
434 the comparison of machine learning and deep learning methods in sentiment analysis, the TF-IDF
435 and fastText methods were compared to create word embedding. The best result was associated
436 with fastText and CNN. The main achievement of this model is the reduction of the need for data
437 pre-processing. Data pre-processing in English is convenient and accurate due to the expanded text
438 pre-processing libraries. However, in other languages, data pre-processing is very complicated
439 because of the lack of proper libraries. Over the suggested model was proved that this need is
440 largely solvable (AUC= 0.996) and the pre-processing steps can be reduced to preliminary
441 tokenization processes. Avoiding complex text preprocessing is also important for other languages
442 Since most text preprocessing algorithms have been developed for English and cannot be used for
443 other languages. In other words, the taken steps are possible to be implemented for other languages
444 to achieve the same results independently of pre-processing steps. Moreover, the created word
445 embedding due to its high accuracy can be used in other text analysis problems especially related
446 to online digital goods.

447

6. References

449

450 Ali, Abder-Rahman, Jingpeng Li, Guang Yang, and Sally Jane O'Shea. 2020. "A Machine Learning
451   Approach to Automatic Detection of Irregularity in Skin Lesion Border Using Dermoscopic Images."
452   *PeerJ Computer Science* 6 (June): e268. https://doi.org/10.7717/peerj-cs.268.

453 Amiri, Fatemeh, Simon Scerri, and Mohammadhassan Khodashahi. 2015. "Lexicon-Based Sentiment
454   Analysis for Persian Text." In *Proceedings of the International Conference Recent Advances in
455   Natural Language Processing*, 9–16. https://doi.org/10.13140/RG.2.1.2537.8327.

456 Boiy, Erik, and Marie-Francine Moens. 2009. "A Machine Learning Approach to Sentiment Analysis in
457   Multilingual Web Texts." *Information Retrieval* 12 (5): 526–58. https://doi.org/10.1007/s10791-
458   008-9070-z.

459 Dashtipour, Kia, Mandar Gogate, Jingpeng Li, Fengling Jiang, Bin Kong, and Amir Hussain. 2020. "A
460   Hybrid Persian Sentiment Analysis Framework: Integrating Dependency Grammar Based Rules and

461   Deep Neural Networks." *Neurocomputing* 380 (March): 1–10.
462   https://doi.org/10.1016/j.neucom.2019.10.009.

463 Deng, Li, Geoffrey Hinton, and Brian Kingsbury. 2013. "New Types of Deep Neural Network Learning for
464   Speech Recognition and Related Applications: An Overview." In *2013 IEEE International Conference
465   on Acoustics, Speech and Signal Processing*, 8599–8603. IEEE.
466   https://doi.org/10.1109/ICASSP.2013.6639344.

467 Derczynski, Leon. 2016. "Complementarity, F-Score, and NLP Evaluation." In *Proceedings of the Tenth
468   International Conference on Language Resources and Evaluation (LREC'16)*, 261–66.
469   https://www.aclweb.org/anthology/L16-1040.

470 Díez-Pastor, José F, Juan J Rodríguez, César García-Osorio, and Ludmila I Kuncheva. 2015. "Random
471   Balance: Ensembles of Variable Priors Classifiers for Imbalanced Data." *Knowledge-Based Systems*
472   85 (September): 96–111. https://doi.org/10.1016/j.knosys.2015.04.022.

473 Forootan, Faezeh, and Mohammad Rabiei. 2019. "Sentiment Analysis User Comments On E-Commerce
474   Online Sale Websites." *International Journal of Web Research* 2 (2): 1–8.
475   https://doi.org/10.22133/IJWR.2020.210555.1048.

476 Gacesa, Ranko, David J Barlow, and Paul F Long. 2016. "Machine Learning Can Differentiate Venom
477   Toxins from Other Proteins Having Non-Toxic Physiological Functions." *PeerJ Computer Science* 2
478   (October): e90. https://doi.org/10.7717/peerj-cs.90.

479 Ghoreyshi, Amir Mohammad, Alireza AkhavanPour, and Alireza Bossaghzadeh. 2020. "Simultaneous
480   Vehicle Detection and Classification Model Based on Deep YOLO Networks." In *2020 International
481   Conference on Machine Vision and Image Processing (MVIP)*, 1–6. IEEE.
482   https://doi.org/10.1109/MVIP49855.2020.9116922.

483 Gurunath Shivakumar, Prashanth, and Panayiotis Georgiou. 2019. "Confusion2Vec: Towards Enriching
484   Vector Space Word Representations with Representational Ambiguities." *PeerJ Computer Science* 5
485   (June): e195. https://doi.org/10.7717/peerj-cs.195.

486 Hajba, Gábor László. 2018. *Website Scraping with Python*. Berkeley, CA: Apress.
487   https://doi.org/10.1007/978-1-4842-3925-4.

488 Hu, Nan, Ling Liu, and Jie Jennifer Zhang. 2008. "Do Online Reviews Affect Product Sales? The Role of
489   Reviewer Characteristics and Temporal Effects." *Information Technology and Management* 9 (3):
490   201–14. https://doi.org/10.1007/s10799-008-0041-2.

491 Irfan, Rizwana, Christine K King, Daniel Grages, Sam Ewen, Samee U Khan, Sajjad A Madani, Joanna
492   Kolodziej, Lizhe Wang, Dan Chen, Ammar Rayes, Nikolaos Tziritas, Cheng-Zhong Xu, Albert Y.
493   Zomaya, Ahmed Saeed Alzahrani and Hongxiang Li. 2015. "A Survey on Text Mining in Social
494   Networks." *The Knowledge Engineering Review* 30 (2): 157–70.
495   https://doi.org/10.1017/S0269888914000277.

496 Ji, Pu, Hong-Yu Zhang, and Jian-Qiang Wang. 2019. "A Fuzzy Decision Support Model With Sentiment
497   Analysis for Items Comparison in E-Commerce: The Case Study of Http://PConline.Com." *IEEE
498   Transactions on Systems, Man, and Cybernetics: Systems* 49 (10): 1993–2004.
499   https://doi.org/10.1109/TSMC.2018.2875163.

500 Joulin, Armand, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. "Bag of Tricks for Efficient
501   Text Classification." *ArXiv Preprint ArXiv:1607.01759*, July. http://arxiv.org/abs/1607.01759.

502   Kim, Yoon. 2014. "Convolutional Neural Networks for Sentence Classification." *ArXiv Preprint*
503   *ArXiv:1408.5882*, August. http://arxiv.org/abs/1408.5882.

504   Lee, Dong-Hyun. 2013. "Pseudo-Label: The Simple and Efficient Semi-Supervised Learning Method for
505   Deep Neural Networks." *Workshop on Challenges in Representation Learning, ICML* 3 (2).

506   Li, Haoqi, Brian Baucom, and Panayiotis Georgiou. 2020. "Linking Emotions to Behaviors through Deep
507   Transfer Learning." *PeerJ Computer Science* 6 (January): e246. https://doi.org/10.7717/peerj-
508   cs.246.

509   Li, Xiaolin, Chaojiang Wu, and Feng Mai. 2019. "The Effect of Online Reviews on Product Sales: A Joint
510   Sentiment-Topic Analysis." *Information & Management* 56 (2): 172–84.
511   https://doi.org/10.1016/j.im.2018.04.007.

512   Liang, Ruxia, and Jian-qiang Wang. 2019. "A Linguistic Intuitionistic Cloud Decision Support Model with
513   Sentiment Analysis for Product Selection in E-Commerce." *International Journal of Fuzzy Systems*
514   21 (3): 963–77. https://doi.org/10.1007/s40815-019-00606-0.

515   Lin, Yi-zhou, Zhen-hua Nie, and Hong-wei Ma. 2017. "Structural Damage Detection with Automatic
516   Feature-Extraction through Deep Learning." *Computer-Aided Civil and Infrastructure Engineering*
517   32 (12): 1025–46. https://doi.org/10.1111/mice.12313.

518   Luo, Jun, Senchun Chai, Baihai Zhang, Yuanqing Xia, Jianlei Gao, and Guoqiang Zeng. 2020. "A Novel
519   Intrusion Detection Method Based on Threshold Modification Using Receiver Operating
520   Characteristic Curve." *Concurrency and Computation: Practice and Experience* 32 (14): e5690.
521   https://doi.org/10.1002/cpe.5690.

522   Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. "Efficient Estimation of Word
523   Representations in Vector Space." *ArXiv Preprint ArXiv:1301.3781*, January.
524   http://arxiv.org/abs/1301.3781.

525   Mohtaj, Salar, Behnam Roshanfekr, Atefeh Zafarian, and Habibollah Asghari. 2018. "Parsivar: A Language
526   Processing Toolkit for Persian." In *Proceedings of the Eleventh International Conference on
527   Language Resources and Evaluation (LREC 2018)*. https://www.aclweb.org/anthology/L18-1179.

528   Nedjah, Nadia, Igor Santos, and Luiza de Macedo Mourelle. 2019. "Sentiment Analysis Using
529   Convolutional Neural Network via Word Embeddings." *Evolutionary Intelligence*, April, 1–25.
530   https://doi.org/10.1007/s12065-019-00227-4.

531   Nourian, Alireza. 2013. "Hazm: Python Library for Digesting Persian Text." 2013.
532   https://github.com/sobhe/hazm.

533   Oscar, Nels, Pamela A Fox, Racheal Croucher, Riana Wernick, Jessica Keune, and Karen Hooker. 2017.
534   "Machine Learning, Sentiment Analysis, and Tweets: An Examination of Alzheimer's Disease Stigma
535   on Twitter." *The Journals of Gerontology: Series B* 72 (5): 742–51.
536   https://doi.org/10.1093/geronb/gbx014.

537   Preethi Krishna, P., and A. Sharada. 2020. "Word Embeddings - Skip Gram Model." In *ICICCT 2019 –
538   System Reliability, Quality Control, Safety, Maintenance and Management*, 133–39. Singapore:
539   Springer Singapore. https://doi.org/10.1007/978-981-13-8461-5_15.

540   Pu, Yunchen, Zhe Gan, Ricardo Henao, Xin Yuan, Chunyuan Li, Andrew Stevens, and Lawrence Carin.
541   2016. "Variational Autoencoder for Deep Learning of Images, Labels and Captions." *Advances in*

542   *Neural Information Processing Systems*, September, 2352–60. http://arxiv.org/abs/1609.08976.

543 Rajaraman, Sivaramakrishnan, Sameer K Antani, Mahdieh Poostchi, Kamolrat Silamut, Md A Hossain,
544   Richard J Maude, Stefan Jaeger, and George R Thoma. 2018. "Pre-Trained Convolutional Neural
545   Networks as Feature Extractors toward Improved Malaria Parasite Detection in Thin Blood Smear
546   Images." *PeerJ* 6 (April): e4568. https://doi.org/10.7717/peerj.4568.

547 Arora, Monika, and Vineet Kansal. 2019. "Character Level Embedding with Deep Convolutional Neural
548   Network for Text Normalization of Unstructured Data for Twitter Sentiment Analysis." *Social
549   Network Analysis and Mining* 9 (1): 12. https://doi.org/10.1007/s13278-019-0557-y.

550 Asgarian, Ehsan, Mohsen Kahani, and Shahla Sharifi. 2018. "The Impact of Sentiment Features on the
551   Sentiment Polarity Classification in Persian Reviews." *Cognitive Computation* 10 (1): 117–35.
552   https://doi.org/10.1007/s12559-017-9513-1.

553 Gan, Chenquan, Lu Wang, Zufan Zhang, and Zhangyi Wang. 2020. "Sparse Attention Based Separable
554   Dilated Convolutional Neural Network for Targeted Sentiment Analysis." *Knowledge-Based Systems*
555   188 (January): 104827. https://doi.org/10.1016/j.knosys.2019.06.035.

556 Gers, F.A. 1999. "Learning to Forget: Continual Prediction with LSTM." In *9th International Conference
557   on Artificial Neural Networks: ICANN '99*, 1999:850–55. IEE. https://doi.org/10.1049/cp:19991218.

558 Graves, Alex, and Jürgen Schmidhuber. 2005. "Framewise Phoneme Classification with Bidirectional
559   LSTM and Other Neural Network Architectures." *Neural Networks* 18 (5–6): 602–10.
560   https://doi.org/10.1016/j.neunet.2005.06.042.

561 Greff, Klaus, Rupesh K. Srivastava, Jan Koutnik, Bas R. Steunebrink, and Jurgen Schmidhuber. 2017.
562   "LSTM: A Search Space Odyssey." *IEEE Transactions on Neural Networks and Learning Systems* 28
563   (10): 2222–32. https://doi.org/10.1109/TNNLS.2016.2582924.

564 Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. "Long Short-Term Memory." *Neural Computation* 9
565   (8): 1735–80. https://doi.org/10.1162/neco.1997.9.8.1735.

566 Kim, Yoon. 2014. "Convolutional Neural Networks for Sentence Classification." *ArXiv Preprint
567   ArXiv:1408.5882*, August. http://arxiv.org/abs/1408.5882.

568 Kontopoulos, Efstratios, Christos Berberidis, Theologos Dergiades, and Nick Bassiliades. 2013.
569   "Ontology-Based Sentiment Analysis of Twitter Posts." *Expert Systems with Applications* 40 (10):
570   4065–74. https://doi.org/10.1016/j.eswa.2013.01.001.

571 Montejo-Ráez, Arturo, Eugenio Martínez-Cámara, M. Teresa Martín-Valdivia, and L. Alfonso Ureña-
572   López. 2014. "Ranked WordNet Graph for Sentiment Polarity Classification in Twitter." *Computer
573   Speech & Language* 28 (1): 93–107. https://doi.org/10.1016/j.csl.2013.04.001.

574 Nguyen, Tuan-Linh, Swathi Kavuri, and Minho Lee. 2018. "A Fuzzy Convolutional Neural Network for Text
575   Sentiment Analysis." Edited by Seong Oun Hwang. *Journal of Intelligent & Fuzzy Systems* 35 (6):
576   6025–34. https://doi.org/10.3233/JIFS-169843.

577 Paltoglou, Georgios, and Mike Thelwall. 2012. "Twitter, MySpace, Digg." *ACM Transactions on Intelligent
578   Systems and Technology* 3 (4): 1–19. https://doi.org/10.1145/2337542.2337551.

579 Rachum, Ram. 2020. "Regular Expression Operations." 2020.
580   https://github.com/python/cpython/blob/3.9/Lib/re.py.

581 Rao, Guozheng, Weihang Huang, Zhiyong Feng, and Qiong Cong. 2018. "LSTM with Sentence
582   Representations for Document-Level Sentiment Classification." *Neurocomputing* 308 (September):
583   49–57. https://doi.org/10.1016/j.neucom.2018.04.045.

584 Richardson, Leonard. 2020. "Beautiful Soup 4.9.3." Crummy. 2020.
585   https://www.crummy.com/software/BeautifulSoup/.

586 Schuster, M., and K.K. Paliwal. 1997. "Bidirectional Recurrent Neural Networks." *IEEE Transactions on*
587   *Signal Processing* 45 (11): 2673–81. https://doi.org/10.1109/78.650093.

588 Vogels, W. 2016. "Bringing the Magic of Amazon AI and Alexa to Apps on AWS." 2016.
589   https://www.allthingsdistributed.com/2016/11/amazon-ai-and-alexa-for-all-aws-apps.html.

590 Wehrmann, Joonatas, Willian Becker, Henry E. L. Cagnini, and Rodrigo C. Barros. 2017. "A Character-
591   Based Convolutional Neural Network for Language-Agnostic Twitter Sentiment Analysis." In *2017*
592   *International Joint Conference on Neural Networks (IJCNN)*, 2384–91. IEEE.
593   https://doi.org/10.1109/IJCNN.2017.7966145.

594 Wilson, Theresa, Paul Hoffmann, Swapna Somasundaran, Jason Kessler, Janyce Wiebe, Yejin Choi, Claire
595   Cardie, Ellen Riloff, and Siddharth Patwardhan. 2005. "OpinionFinder." In *Proceedings of*
596   *HLT/EMNLP on Interactive Demonstrations -*, 34–35. Morristown, NJ, USA: Association for
597   Computational Linguistics. https://doi.org/10.3115/1225733.1225751.

598 Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey,
599   Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson,
600   Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith
601   Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex
602   Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, Jeffrey Dean. 2016. "Google's Neural
603   Machine Translation System: Bridging the Gap between Human and Machine Translation,"
604   September. http://arxiv.org/abs/1609.08144.

605 Yadav, Ashima, and Dinesh Kumar Vishwakarma. 2020. "Sentiment Analysis Using Deep Learning
606   Architectures: A Review." *Artificial Intelligence Review* 53 (6): 4335–85.
607   https://doi.org/10.1007/s10462-019-09794-5.

608 Ye, Qiang, Ziqiong Zhang, and Rob Law. 2009. "Sentiment Classification of Online Reviews to Travel
609   Destinations by Supervised Machine Learning Approaches." *Expert Systems with Applications* 36
610   (3): 6527–35. https://doi.org/10.1016/j.eswa.2008.07.035.

611 Yildirim, Özal. 2018. "A Novel Wavelet Sequence Based on Deep Bidirectional LSTM Network Model for
612   ECG Signal Classification." *Computers in Biology and Medicine* 96 (May): 189–202.
613   https://doi.org/10.1016/j.compbiomed.2018.03.016.

614 Zhang, Ye, and Byron Wallace. 2015. "A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional
615   Neural Networks for Sentence Classification," October. http://arxiv.org/abs/1510.03820.

616 Roshanfekr, Behnam, Shahram Khadivi, and Mohammad Rahmati. 2017. "Sentiment Analysis Using Deep
617   Learning on Persian Texts." In *2017 Iranian Conference on Electrical Engineering (ICEE)*, 1503–8.
618   IEEE. https://doi.org/10.1109/IranianCEE.2017.7985281.

619 Saraee, Mohamad, and Ayoub Bagheri. 2013. "Feature Selection Methods in Persian Sentiment
620   Analysis." In *International Conference on Application of Natural Language to Information Systems*,
621   303–8. Springer. https://doi.org/10.1007/978-3-642-38824-8_29.

622 Severyn, Aliaksei, and Alessandro Moschitti. 2015. "Twitter Sentiment Analysis with Deep Convolutional
623   Neural Networks." In *Proceedings of the 38th International ACM SIGIR Conference on Research and
624   Development in Information Retrieval - SIGIR '15*, 959–62. New York, New York, USA: ACM Press.
625   https://doi.org/10.1145/2766462.2767830.

626 Sokolova, Marina, Nathalie Japkowicz, and Stan Szpakowicz. 2006. "Beyond Accuracy, F-Score and ROC:
627   A Family of Discriminant Measures for Performance Evaluation." In *Australasian Joint Conference
628   on Artificial Intelligence*, 1015–21. Springer. https://doi.org/10.1007/11941439_114.

629 Taboada, Maite, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. "Lexicon-Based
630   Methods for Sentiment Analysis." *Computational Linguistics* 37 (2): 267–307.
631   https://doi.org/10.1162/COLI_a_00049.

632 Tang, Duyu, Bing Qin, and Ting Liu. 2015. "Deep Learning for Sentiment Analysis: Successful Approaches
633   and Future Challenges." *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 5
634   (6): 292–303. https://doi.org/10.1002/widm.1171.

635 Tausczik, Yla R, and James W Pennebaker. 2010. "The Psychological Meaning of Words: LIWC and
636   Computerized Text Analysis Methods." *Journal of Language and Social Psychology* 29 (1): 24–54.
637   https://doi.org/10.1177/0261927X09351676.

638 Velupillai, Sumithra, Hercules Dalianis, Martin Hassel, and Gunnar H Nilsson. 2009. "Developing a
639   Standard for De-Identifying Electronic Patient Records Written in Swedish: Precision, Recall and F-
640   Measure in a Manual and Computerized Annotation Trial." *International Journal of Medical
641   Informatics* 78 (12): e19–26. https://doi.org/10.1016/j.ijmedinf.2009.04.005.

642 Vorraboot, Piyanoot, Suwanna Rasmequan, Krisana Chinnasarn, and Chidchanok Lursinsap. 2015.
643   "Improving Classification Rate Constrained to Imbalanced Data between Overlapped and Non-
644   Overlapped Regions by Hybrid Algorithms." *Neurocomputing* 152 (March): 429–43.
645   https://doi.org/10.1016/j.neucom.2014.10.007.

646 Willett, Peter. 2006. "The Porter Stemming Algorithm: Then and Now." *Program* 40 (3): 219–23.
647   https://doi.org/10.1108/00330330610681295.

648 Yazdinejad, Mohsen, and Sajjad Shumaly. 2020. "Persian Sentiment Analysis of an Online Store Using
649   Convolutional Neural Network with FastText Embeddings." 2020.
650   https://github.com/mosiomohsen/persian-sentiment-analysis-using-fastText-word-embedding-
651   and-pseudo-labeling.

652 Zhang, Lei, Shuai Wang, and Bing Liu. 2018. "Deep Learning for Sentiment Analysis: A Survey." *Wiley
653   Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8 (4): e1253.
654   https://doi.org/10.1002/widm.1253.

655 Zhao, Hongshan, Huihai Liu, Wenjing Hu, and Xihui Yan. 2018. "Anomaly Detection and Fault Analysis of
656   Wind Turbine Components Based on Deep Learning Network." *Renewable Energy* 127 (November):
657   825–34. https://doi.org/10.1016/j.renene.2018.05.024.

658

Figure 1

A convolutional network architecture to sentiment classification

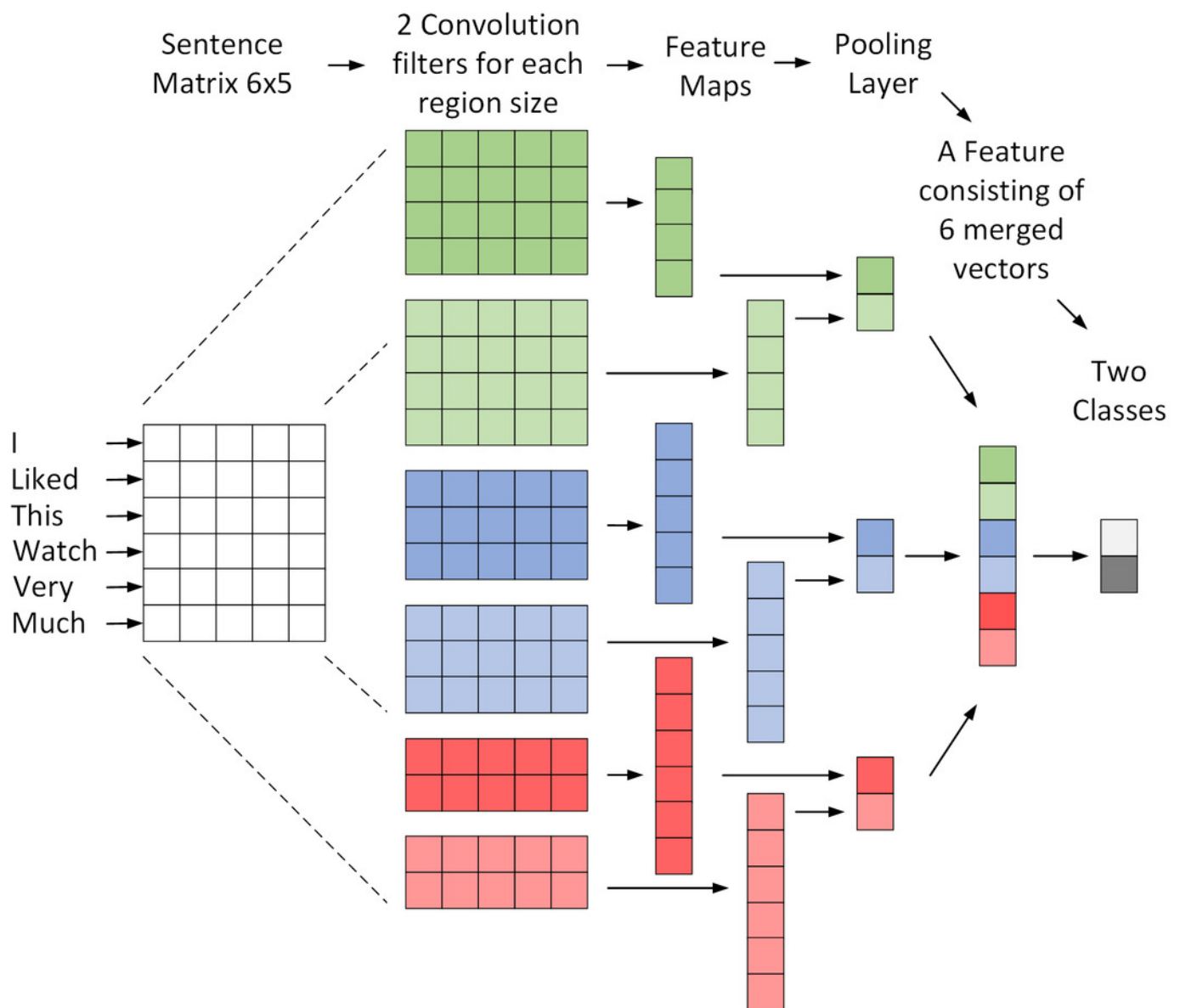



Figure 2

A demonstration of an LSTM cell

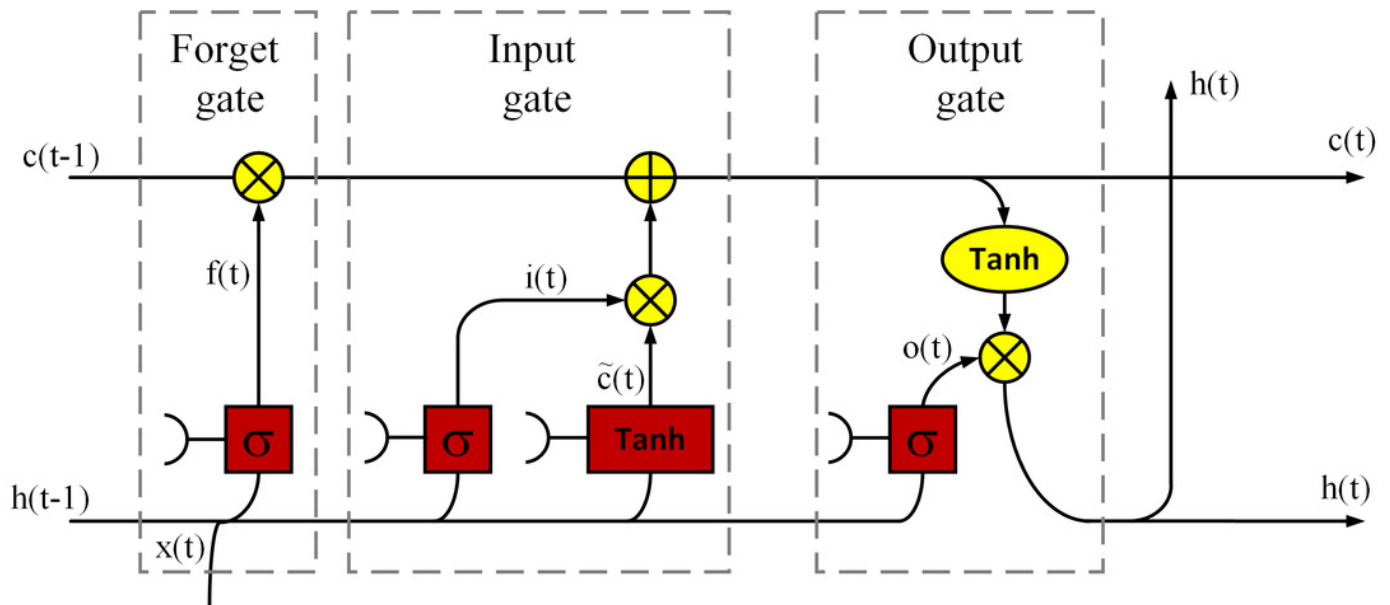



Figure 3

A basic structure of the BiLSTM network

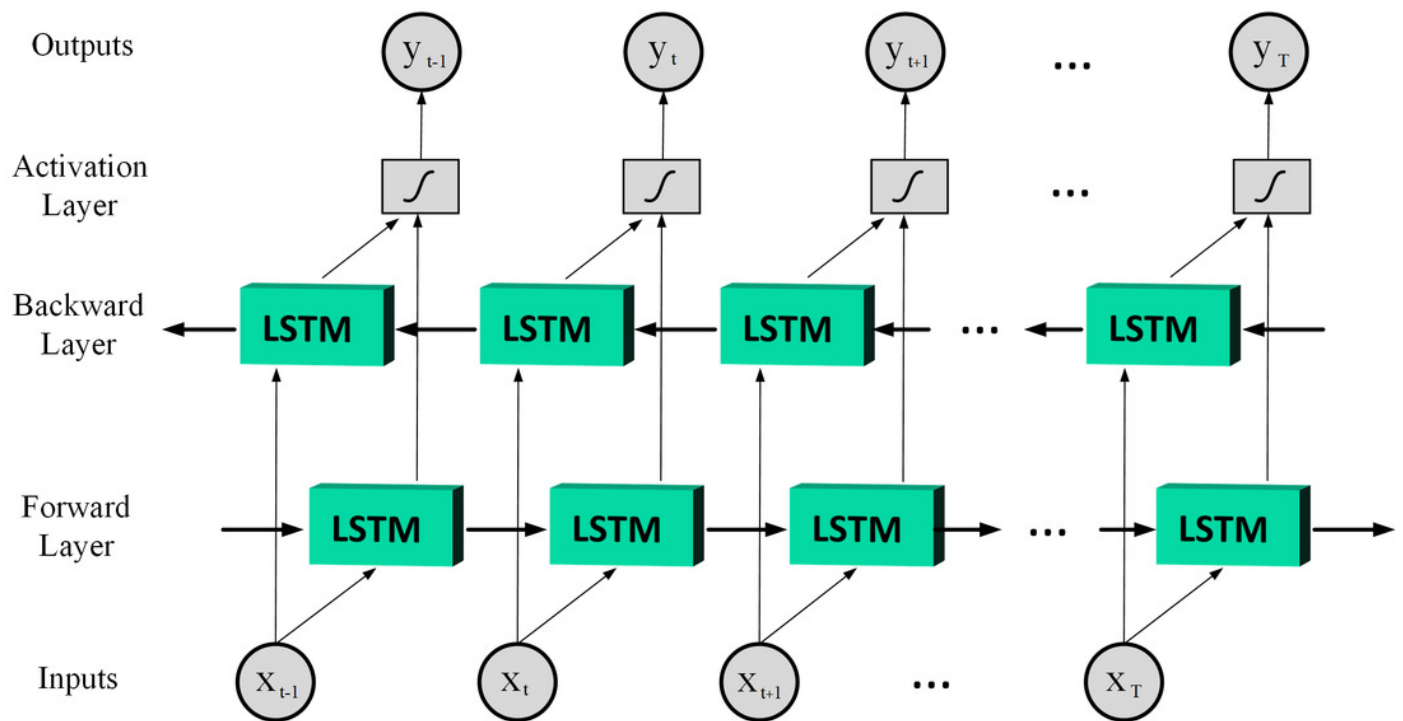



Figure 4

A flowchart representing the taken steps

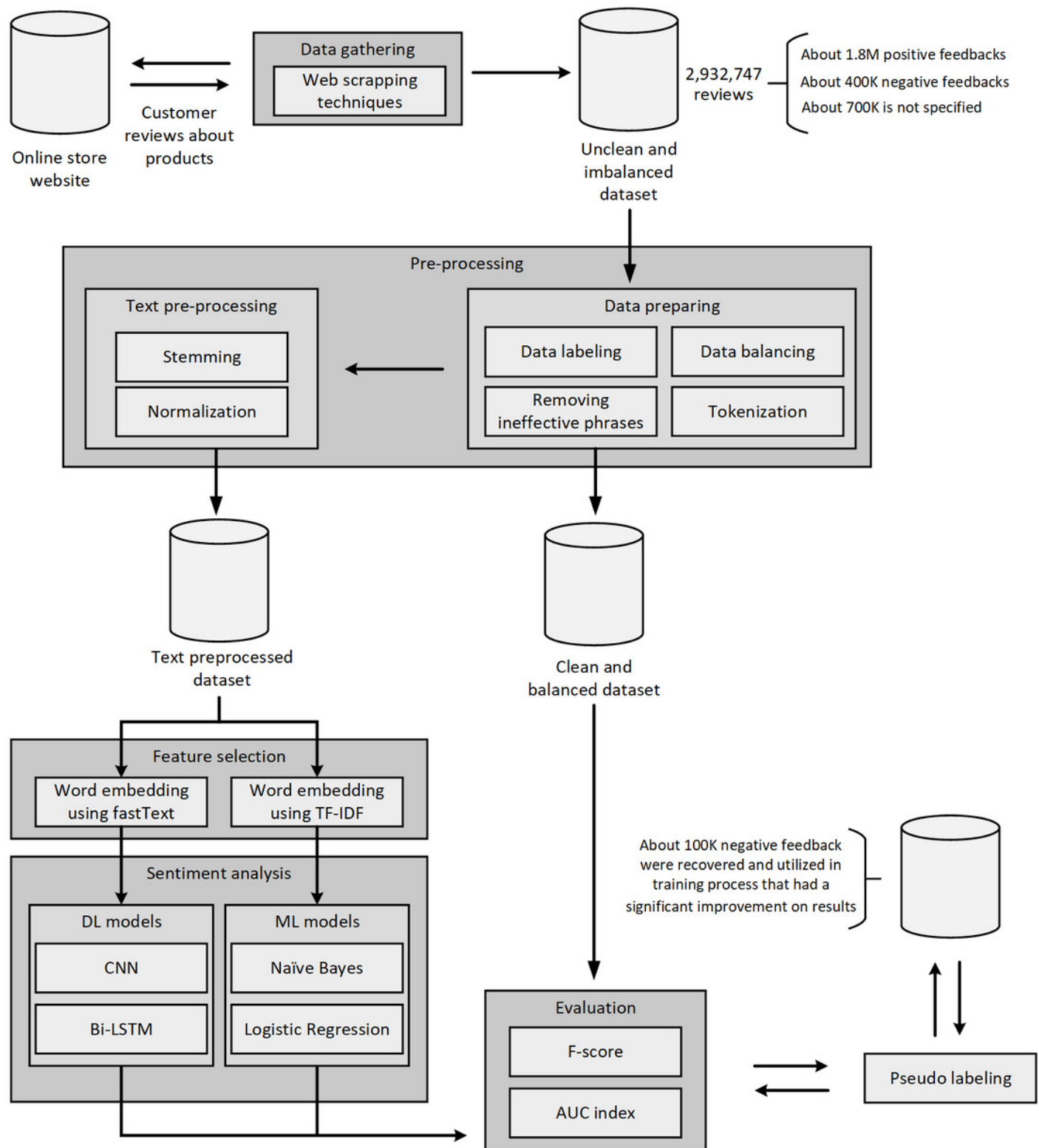



Figure 5

Frequency of reviews per category

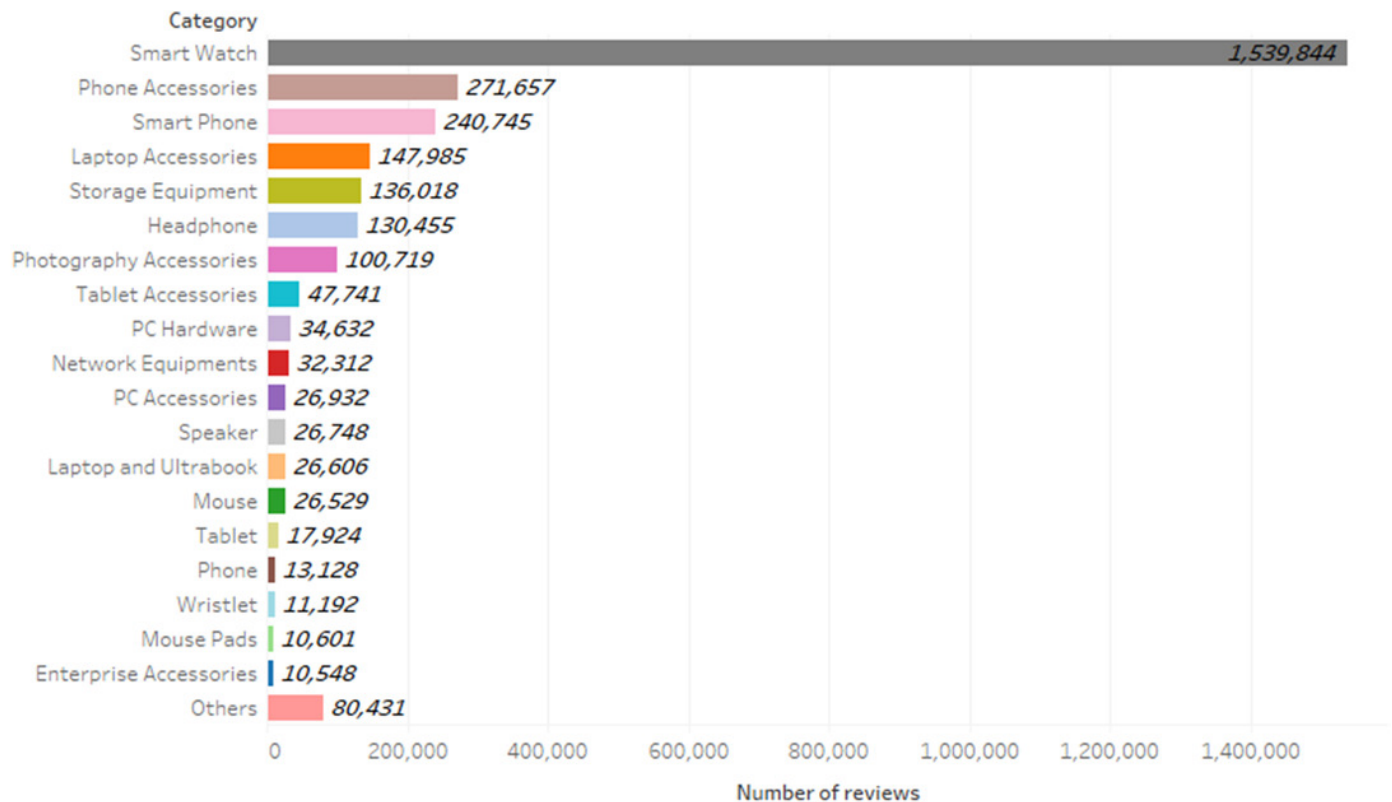



Figure 6

Positive feedback class word cloud

English	Persian
I was satisfied	راضی بودم
Good	خوب
Quality	کیفیت
I suggest	پیشنهاد میکنم
Worth buying	ارزش خرید
Speed	سرعت
Excelent	عالی
Very good	بسیار خوب
Price	قیمت
Good quality	کیفیت خوب
Material	جنس
I am satisfied	من راضیم

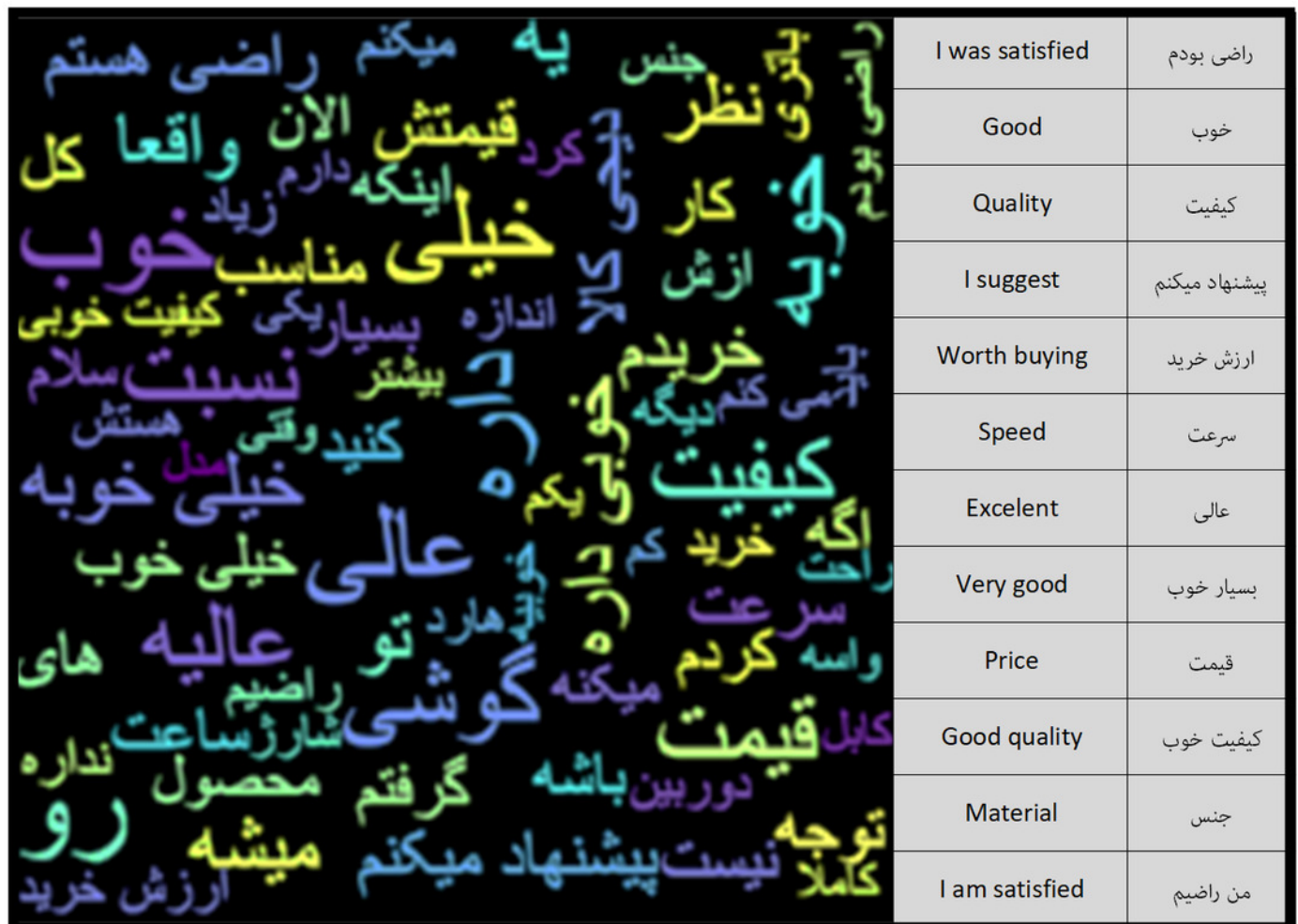



Figure 7

Negative feedback class word cloud

English	Persian
It did not work	کار نکرد
Broken	خراب
Awful	افتضاح
I referred	مرجوع کردم
Does not have	نداره
Unfortunately	متاسفانه
Is not	نیست
Not worth buying	ارزش خرید ندارد
Not good	خوب نیست
Do not buy	نخرید
I do not recommend	پیشنهاد نمیکنم
It did not turn on	روشن نشد

Figure 8

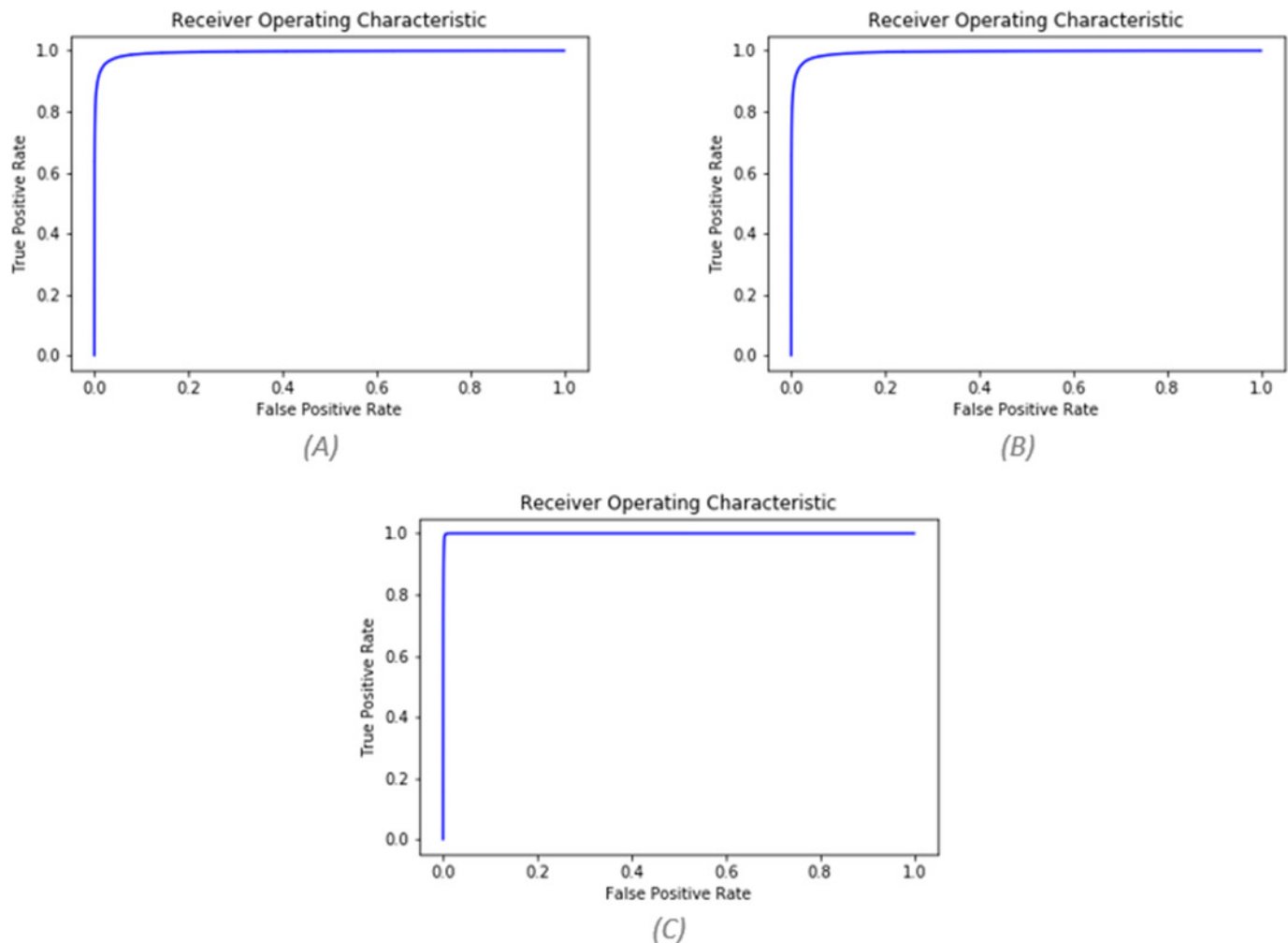
A: AUC before pre-processing (AUC=0.9943), B: AUC after pre-processing (AUC=0.9944), C: AUC after pseudo labeling (AUC=0.9996)

The CNN model structure

Table 1 The CNN model structure

Layer (type)	Output Shape	Number of Parameters
embedding_2 (Embedding)	(None, 400, 100)	11147700
dropout_8 (Dropout)	(None, 400, 100)	0
conv1d (Conv1D)	(None, 400, 128)	38528
global_max_pooling1d (Global)	(None, 128)	0
dense_6 (Dense)	(None, 64)	8256
dropout_9 (Dropout)	(None, 64)	0
dense_7 (Dense)	(None, 16)	1040
dropout_10 (Dropout)	(None, 16)	0
dense_8 (Dense)	(None, 1)	17
Total parameters:		11,195,541

The BiLSTM model structure

Table 2 The BiLSTM model structure

Layer (type)	Output Shape	Number of Parameters
embedding_1 (Embedding)	(None, 400, 100)	11147700
dropout_4 (Dropout)	(None, 400, 100)	0
bidirectional_1 (Bidirection)	(None, 64)	34048
dropout_5 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 64)	4160
dropout_6 (Dropout)	(None, 64)	0
dense_4 (Dense)	(None, 16)	1040
dropout_7 (Dropout)	(None, 16)	0
dense_5 (Dense)	(None, 1)	17
Total parameters:		11,186,965

A sample of the collected dataset

1   *Table 3 A sample of the collected dataset*

Comment ID	Original Comment	Translated Comment	Negative Feedback	Positive Feedback	Cat. Name
0	من کاملا با این محصول آشنا بودم و از خریدش مطمین بودم	I was completely familiar with this product and I was sure of buying it	0	1	Smart Watch
31	به نسبت قیمتش عاااالیه	Grrrreat for the price	0	1	Smart Watch
84278	چیز بدی نیست کار راه میندازه	Not a bad thing	0	0	Phone Accessories
3083	با توجه به معرفی پرچمدار جدید اچ تی سی u11 خرید این گوشی عاقلانه نیست .	Given the introduction of the new flagship HTC u11, buying this phone is not a wise choice	1	0	Smart Phone
1503094	رنگ مد نظر ارسال نشد	The requested color was not sent	1	0	Smart Watch

Performance of different models based on AUC and F-measure

1   *Table 4 Performance of different models based on AUC and F-measure*

States	Index	Fold1	Fold2	Fold 3	Fold 4	Fold 5	Mean	Error (SEM)
BiLSTM	AUC:	0.9934	0.9937	0.993	0.993	0.9934	0.9933	$13.4*10^{-5}$
	F-score:	0.9224	0.9244	0.9238	0.9216	0.9232	0.9230	$49.6*10^{-5}$
CNN	AUC:	0.9945	0.9945	0.9943	0.9946	0.9945	0.9944	$4.9*10^{-5}$
	F-score:	0.9293	0.9251	0.9306	0.9299	0.93	0.9289	$99.2*10^{-5}$
Naïve Bayes	AUC:	0.9877	0.9881	0.9878	0.988	0.9881	0.9879	$8.12*10^{-5}$
	F-score:	0.8856	0.8856	0.886	0.8863	0.8863	0.8859	$15.7*10^{-5}$
Logistic Regression	AUC:	0.9888	0.9891	0.9888	0.989	0.9881	0.9887	$17.5*10^{-5}$
	F-score:	0.8894	0.8901	0.8898	0.8895	0.8863	0.8890	$69.1*10^{-5}$

Performance of the CNN model in different situations based on AUC and F-measure

1   *Table 5 Performance of the CNN model in different situations based on AUC and F-measure*

States	Index	Fold1	Fold2	Fold 3	Fold 4	Fold 5	Mean	Error (SEM)
Before Prep.	AUC:	0.9943	0.9943	0.9944	0.9944	0.9945	0.994	$3.7*10^{-5}$
	F-score:	0.928	0.9298	0.9303	0.9271	0.9304	0.929	$66.4*10^{-5}$
After Prep.	AUC:	0.9945	0.9945	0.9943	0.9946	0.9945	0.994	$4.8*10^{-5}$
	F-score:	0.9293	0.9251	0.9306	0.9299	0.93	0.928	$99.1*10^{-5}$
After Pseudo labeling	AUC:	0.9944	0.9943	0.9946	0.9995	0.9996	0.996	$12.5*10^{-5}$
	F-score:	0.9431	0.9434	0.9443	0.9767	0.9758	0.956	$80*10^{-5}$