E-ISSN: 2676-6167 | P-ISSN: 2538-5100

Journal of Applied Research on Industrial Engineering

www.journal-aprie.com

J. Appl. Res. Ind. Eng. Vol. x, No. x (xx) x–x.

Paper Type: Original Article

# INDCAPS: The IndRNN Capsule Approach for Persian Multi-Domain Sentiment Analysis

Ramin, Mousa; Mohammad Ali, Dadgostarnia; Amir, Olfati Malamiri; Elham, Behnam; Shahram, Miri Kelaniki

**Citation:**

LastName, (Abreviation of FirstName). , & LastName, (Abreviation of FirstName. (Date). Paper Title. *Journal of applied research on industrial engineering, Volume* (Issue), PP.

Received: Reviewed: Revised: Accept:

## Abstract

Sentiment Analysis (SA) is the computational analysis of ideas, feelings and opinions using natural language processing techniques, computational methods and text analysis to extract polarity (positive, negative or neutral) from unstructured documents or textual comments. Multi-domain SA is based on a labelled dataset, which reduces the dependence on large amounts of domain-specific data and addresses data scarcity issues by leveraging existing data from other domains. This paper presents a novel deep learning-based approach for Persian multi-domain SA analysis. The proposed Bi-IndRNNCapsule technique combines bidirectional IndRNN and CapsuleNet, which use Bi-GRU to extract features for CapsuleNet. In IndRNN, recurrent layer neurons operate independently, with simple RNN computing the hidden state h via element-wise vector multiplication u * state, indicating that each neuron has a solitary recurrent weight linking it to the most recent hidden state. We evaluated the proposed approach on the Digikala dataset and found it to provide acceptable accuracy compared to existing techniques.

**Keywords**: Multi-domain sentiment analysis, Deep learning, Persian sentiment analysis, Natural language processing, IndRNN

## 1 | Introduction

The recent emergence of online communities, social networking platforms and blogging sites has not only streamlined expression, but also significantly changed how emotions or opinions can be communicated between users more efficiently and without restrictions [2]. These systems use text-based evaluations to determine how users feel about products and services. While the sheer volume of data makes data extraction and analysis time-consuming and expensive, it is necessary [3] as interactions between people are expressed through opinions and sentiments. Opinions and sentiments influence almost all human actions, motivating researchers to extract information from the vast amount of available data [4]. Opinion mining (OM) or sentiment analysis (SA) is the use of computational techniques to understand people's opinions, feelings and evaluations of services and events [4].



Corresponding Author:

Email: Raminmousa@znu.ac.ir

https://doi.org/10.22105/jarie.2023.345352.1477

SA is a subset of artificial intelligence. An important aspect of AI is the creation of machines that can replicate human characteristics, such as learning and reasoning. These devices also have capabilities such as making decisions based on nuanced factors while exhibiting natural behavior in their interactions with humans [5]. This technology has been integrated into various areas of society, including sensory data analysis [6], big data analysis [7], security monitoring [8], management and planning [9], education [10] and medical care [11]. Artificial intelligence has many applications in NLP, including sentiment analysis [12], keyword extraction [13], and text classification [14].

SA is a task that can be performed using Natural Language Processing (NLP) to classify documents and identify sentiments related to certain topics [15]. Mathematically, an opinion can be presented as $\langle ei,aij,ooijkl,hk,tl \rangle$, where $ei$ is the name of an entity, aij is an aspect of ei, ooijkl is the polarity (positive, negative, or neutral) about aspect $aij$ of $ei$, $hk$ is the opinion holder who expressed the opinion, and $tl$ is the time when the opinion about aspect $aij$ of entity $ei$ is expressed by $hk$ [16]. SA is known as a domain-dependent problem. Creating a classifier based on train data may have many failures in the test data during testing. To solve this problem, a large amount of labelled data is required for each domain. Collecting this data is both expensive and difficult, and in some domains, this information is rarely available [17]. The purpose of multi-domain SA is to find out the sentiment of a text using data from specific domains [15].

This paper attempts to address the challenge of Persian multi-domain sentiment analysis. The proposed Bi-IndRNN capsule approach consists of the following broad steps:

1. Word embedding generation from the reviews using pre-trained FastText embedding.

2. Generation of a label mask.

3. Train the Bi-IndRNNCapsule

4. Compute the output of polarity detection and domain identification.

The proposed technique is unique because it combines two neural networks for multi-domain sentiment analysis, and there is limited research in this area in Persian. In addition, the proposed approach attempts to include domain detection at the same time as polarity detection in the proposed approach. A temporal feature can be learned with IndRNN, and a specific feature can be learned with a capsule using this approach. Through this research, the main results can be outlined as follows:

1. Implementation of Bi-IndRNN and capsule approaches for linguistic domain overlap.
2. Employ the deep learning approach for multi-domain Persian sentiment analysis.
3. Use the Bi-IndRNN for feature extraction and the capsule for feature learning.

This paper is structured as follows:

Section 2 presents an evaluation of machine learning and deep learning models for sentiment analysis, and Section 3 introduces the Bi-IndRNN-capsule method for multi-domain sentiment analysis. Section 4 provides a comprehensive analysis of the experimental results, while Section 5 scrutinizes the error analysis of the proposed approach, and identifies potential future work. Finally, Section 6 concludes the paper.

## 2 | RELATED WORKS

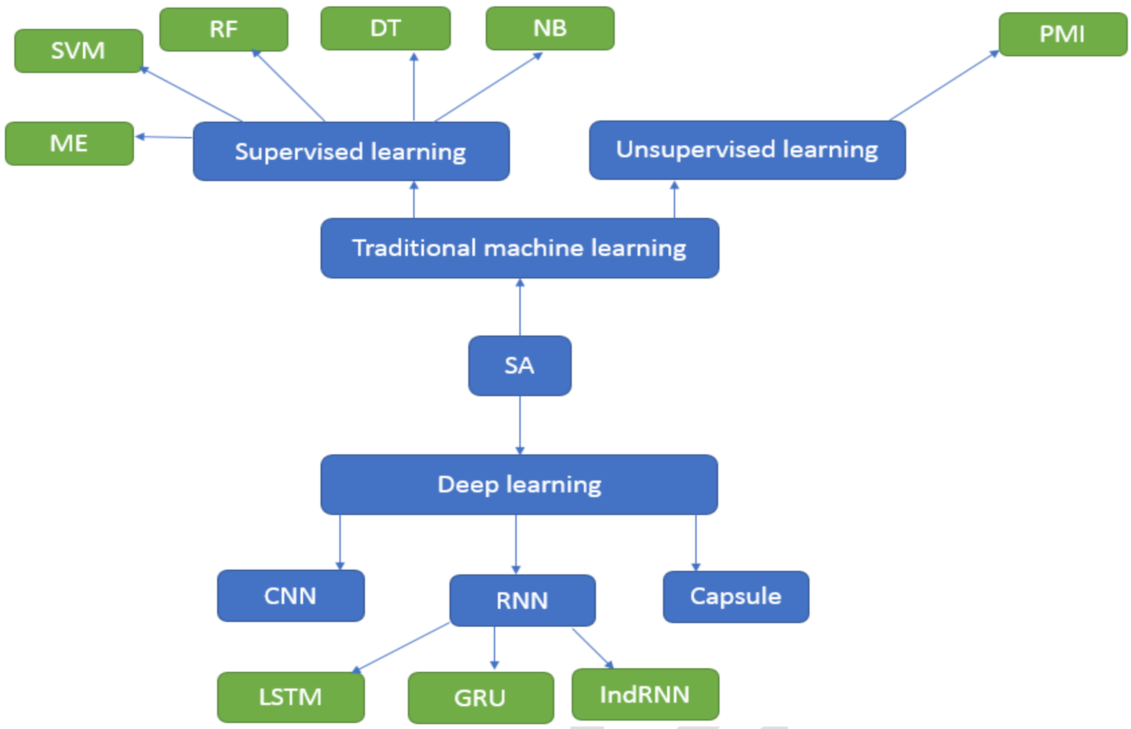


Fig 1. Sentiment analysis approaches [1].

Extensive research on SA has been conducted and documented in the literature [18]. In this section, we will examine and outline two individual approaches to SA: the traditional machine learning approach and the deep learning approach. Figure 1 illustrates the overall structure of all SA approaches.

### 2.1 | Traditional Machine Learning Approaches

The impressive amount of work that SA has put into exploring the many aspects of the literary world is evident [19-21].In this section, we will examine an alternative strategy for SA and evaluate some of the approaches commonly used.

### 2.2 Traditional Machine Learning Approach for SA

As research into sentiment analysis continues to grow, machine-learning techniques for such analysis have become increasingly popular. These methods have two main components: supervised and unsupervised techniques.

#### 2.2.1. Supervised approaches for SA

Supervised learning for sentiment analysis requires labelled data, which is difficult to collect. A supervisor must provide input data and expected results for these algorithms [21]. Several supervised strategies such as Support Vector Machine (SVM), Naïve Bayes (NB) and Maximum Entropy (ME) have achieved great success in this research area [22]. The author [23] evaluated the performance of three classifiers - SVM, NB and ME. The success of these classifiers depends on the extracted features. To achieve their goal, they used unigrams, bigrams, unigrams+bigrams, and a combination of unigrams and bigrams with part-of-speech (POS) tags. The NB classifier produced higher scores with a small feature space, while the SVM classifier proved to be more successful when the feature space was larger. Similarly, [24] used SVM, NB and a character-based n-gram model to identify sentiment in online reviews of travel destinations.

A variety of classifiers have been used to analyse the labelled data. For example, Z. Zhang et al. [25] used SVM and NB approaches to classify film reviews. They used unigrams, bigrams and trigrams as features to train their classifiers. B. Pang et al. [23] used SVM, NB and ME to classify film reviews by considering unigrams, bigrams and the position of adjectives with respect to words. The authors in [24] used SVM, NB and character-based n-grams to classify the sentiment of destination reviews by analysing the frequency of unigrams. Furthermore, in [26], some approaches were considered, mainly SVM and rule-based classifiers, along with POS tag and n-gram features to be evaluated. The use of these features has been implemented for the organisation of movie reviews, product reviews, and space comments. SVM has been used for movie reviews and MPQA classification in [27, 28]. In these works, various features such as unigram, bigram, extracted pattern features, adjective word frequency, and percentage of the review group have been used to train the classifier.

### 2.2.2. Unsupervised Learning Approach for SA

Supervised methods require data to be labelled, and the cost of data collection can be significant. Unsupervised learning approaches use unlabelled data to find similar patterns and useful information within the input data. These strategies are employed when collecting annotated data is challenging, while acquiring annotated data is comparatively more straightforward [29]. In contrast to supervised approaches, there has been little research on these methods, and only a few exist. Turny [30] implemented an unsupervised technique to classify reviews into two categories: recommended or not recommended. His proposed algorithm consists of two fundamental steps. The first step is to obtain sentences containing either an adjective or an adverb, and the second step is to compute the Semantic Orientation (SO) of the sentences obtained in the first step. The PMI-IR algorithm is used to achieve this goal. Mutual information is used by the PMI-IR algorithm to calculate the semantic similarity between two phrases. This is the method for calculating the Pointwise Mutual Information (PMI) between two words according to [31]:

$$\mathrm{PMI}(\mathrm{word}_1, \mathrm{word}_2) = \log_2\left[\frac{\mathrm{p}(\mathrm{word}_1 \& \,\mathrm{word}_2)}{\mathrm{p}(\mathrm{word}_1).\,\mathrm{p}(\mathrm{word}_2)}\right] \qquad (1)$$

The probability of the simultaneous occurrence of $word_1$ and $word_2$ is symbolised by $p(word_1 \& \, word_2)$ , whereas the probability of $word_1$ is represented by $p(word_1)$ and the probability of $word_2$ is indicated by $p(word_2)$. The two words 'excellent' and 'poor' were derived using formula (2) as follows:

$$\text{SO (phrase)} = \text{PMI (phrase, excellent)} - \text{PMI (phrase, poor)} \qquad (2)$$

In [32], several authors proposed a novel unsupervised technique for sentiment analysis on Twitter and tested the algorithm across 12 domains to determine its efficiency. They used dependency parsing, a pre-built sentiment dictionary, linguistic structures, and linguistic content to classify tweets. Taras and John [33] proposed an independent approach to sentiment labelling in Chinese using a set of automatically selected seed words based on two assumptions:

1. The candidate words selected for the seed may be associated with negative words. For example, in Chinese, it is common to use ‘not good’ instead of ‘bad’.
2. The polarity of the candidates had to be identified. To do this, they used the term 'good' as a good standard for the positive words and calculated the patterns contained in the seed candidates with the patterns shown in the gold standard. The results showed that their unsupervised approach was on a par with the supervised approach and in some cases even outperformed it.

### 2.3. Deep Learning for SA

In recent years, with the increasing power of GPUs and the availability of large amounts of data, deep learning (DL) has achieved numerous successes. DL has significantly improved the traditional ML and statistical techniques used in the past for various NLP tasks [34]. Deep learning is helpful in text generation, vector representation, word representation estimation, sentence classification, sentiment analysis, sentence modelling, and feature representation [35]. One problem with using a neural network on text is that the input cannot be provided in raw format as the neural network requires numerical vectors as data and provides a matrix or vector as output. Instead of assigning specific dimensions to each feature, they attempt to embed each feature in a d-dimensional dense vector. The main advantage of these vectors is that similar words are close together in the vector space [36]. Further exploration of the various artificial neural network models in the field of deep learning will be the focus of the following subsections.

#### 2.3.1. Convolutional Neural Networks (CNNs)

CNNS is a specific type of DNN that was first introduced in [37] for image classification. CNNS have been used for various purposes, including human pose estimation [38], object identification [39], speech comprehension [40], time series [41], and NLP [42]. Convolutional Neural Networks (CNNS) are feed-forward neural networks that feature convolution layers, sparse connections, parameter sharing, and pooling [43]. These networks consist of three fundamental layers:

1. **Convolutional layer:** The CNN highlights features by using kernels from the input feature map or intermediate feature map.
2. **Pooling layer:** In order to minimise the amount of network parameters, various methods have been employed such as stochastic max-pooling, average pooling, pooling [44], spatial pyramid pooling [45], and def-pooling [46], which are used as layers to reduce network parameters.
3. **Fully connected layers:** They have a similar architecture to traditional neural networks (NN). In the field of sentiment analysis (SA), CNNs have shown significant success. For instance, Kim [42] experimented with the CNN Multichannel and CNN non-static models for the binary classification of movie reviews. The results indicated an accuracy of 81.1% and 81.5%, respectively.

#### 2.3.2. Recurrent Neural Networks

RNNs are a class of neural networks that take into account the sequential or temporal nature of input data, allowing information from past inputs to influence the network's behavior. Time is incorporated into the model through the use of edges, which can generate cycles. The current state of an RNN is determined by both the current input and the previous time step [47]. Given the current input $X_t$ and the previous network state $h_t$, the output of an RNN is calculated using a set of equations [47]:

$$y^t = \mathrm{softmax}\left(w^{hy} h_t + b_y\right) \tag{3}$$

$$h^t = \sigma\left(w^{hx} * x_t + w^{hh} * h_{t-1} + b_n\right) \tag{4}$$

During the backpropagation process in neural networks, issues such as vanishing gradients and exploding gradients can occur over long time steps. In order to resolve this problem, Hochreiter and Schmidhuber [48] have suggested the implementation of Long-Short Term Memory (LSTM) networks. These networks are similar to RNN networks, but instead of hidden layer nodes, they have memory cells. Each memory cell contains a node with a self-connected recurrent edge, which ensures that the gradient can move through many time steps without vanishing and exploding gradients [48, 49].

The Gated Recurrent Units (GRUs) are an alternative to traditional RNNs for solving gradient vanishing and explosion problems. These are specifically designed to support natural language translation [50]. One of the most recent developments in RNNs is the Independent Recurrent Neural Network (IndRNN), which was constructed with the idea of incorporating independence. IndRNN is a critical function that can be described as [51]:

$$h_t = \sigma\left(Wx_t + u \odot h_{t-1} + b\right) \quad (5)$$

The notation $h_t \in R^N$ represents the hidden state at time step $t$, $\sigma$ denotes the sigmoid activation function, while $W \in R^{NM}$ is a learnable matrix. $x_t \in R^M$ is input at time $t$, $u \in \mathrm{R}^{\mathrm{n}}$ is current input, $\odot$ is the Hadamard product, and $b \in R^n$ is the bias of the neurons. The variables $N$ and $M$ denote the number of neurons in the recurrent layer and the input dimensions, respectively. One of the significant differences between IndRNN and traditional RNNs is the return weight $u$ .In traditional RNNs, this weight is represented as a matrix that transforms the input through matrix multiplication. However, in IndRNN, it is represented as a vector that undergoes process multiplication based on its elements. Each neuron in each layer of the IndRNN operates independently of the other neurons. This is why the term independently recurrent is used to describe it [51]. For the $n-th$ neuron in IndRNN, the hidden state $h_{(n,t)}$ can be computed as follows:

$$\mathrm{h}_{(\mathrm{n,t})} = \sigma\left(\mathrm{w_n X_t} + \mathrm{u_n h_{(n,t-1)}} + \mathrm{b_n}\right) \quad (6)$$

Section 4 provides more information about these networks. Due to their natural language essence and architecture, they are well-suited for sentiment analysis. For example, Li et al. [50] used RNN and LSTM networks to classify binary movie review classification and achieved 74% and 78% accuracy, respectively, for fine-grained classification. Tai et al. [52] introduced the Tree-LSTM algorithm based on LSTM networks which was used for two tasks: (i) predicting the semantic relatedness of two sentences (SemEval 2014 Task1) and (ii) classification (Stanford Sentiment Treebank). Using the D-Tree-LSTM and C-Tree-LSTM models, the binary classifier achieved 85.7% and 88% accuracy for the sentiment treebank, respectively, and 48.4% and 51% accuracy for the fine-grained classification, respectively.

#### 2.3.3. CapsuleNet

CapsuleNet [53] was developed as a solution to address the challenges associated with CNNs and Capsules use a vector as their output instead of a scalar value. This vector contains all the features of an entity, and the magnitude of the vector shows the probability of the entity's existence. The low-layer capsules provide their outputs to the higher-level capsules. With the capsule output represented by a vector, the dynamic routing algorithm sends the output to the parent capsule appropriately. Initially, the output may be sent to all the capsules in the layer below the parent's, but over time, the dynamic routing algorithm selects the optimal output by adjusting the coupling coefficient [53]. The main advantage of CapsuleNet over CNNs is that it requires less training data and can learn a good view of each class of data more quickly [53]. Examples of using CapsuleNet for sentiment analysis can be found in [54-56]. The IndRNN capsule structure for multi-domain sentiment analysis is elaborated in Section 4.

Table 1 lists some critical research conducted for Digikala in Persian SA, and all the investigations focus on single-domain SA.

**Table 1: A Comparison between Sentiment Analysis works in the Persian Language for Digikala.**





| Model | Reference | Tasks | Single domain / Multi-domain | Measure |
|---|---|---|---|---|
| Different classifiers and feature engineering approaches, ML | [57] | Polarity detection | Single domain | Precision=91.22% Recall=91.71% F1=91.46% |
| Unsupervised methods, NN | [58] | Polarity detection | Single domain | Precision=73.7% Recall=99.1% F1=58.6% |
| Sentiment-lexicon generation, polarity classification | [59] | Polarity detection | Single domain | Accuracy=86% F1=80% Recall =75% |
| Sentence-level Sentiment Analysis, lexicon-based, | [60] | Polarity detection | Single domain | Accuracy=94% F1=89% Recall =88% Precision=90% |
| Polarity detection, SVM, | [61] | Polarity detection | Single domain | F1=90.15% Precision=93.03% Recall =87.42% |
| Polarity detection, IndRNN Capsule | Current research | Polarity detection and domain identification | Multi-domain | Polarity detection: Accuracy=0.9489, precision=0.9535, Recall=0.9524 Domain identification: Accuracy=0.8020 Precision=0.9223 Recall0.8829 |



## 3 | Method

In this section, we will discuss the steps necessary to preprocess the data and arrange it for the recommended model. We will also provide a mathematical illustration of our proposed approach.

### 3.1. Data Preparation

In the pre-processing stage, each document is broken down into its constituent words. Stop words and punctuation are then removed from this sequence as they do not contribute to the classification process. This removal allows the classifier to gather more important information. The input sequence is then mapped to a sequence of low-dimensional dense vectors determined by the word embedding table $T^{emb}$, which is created using Fasttext word embedding.

After the mapping, the input sequence can be represented as $S = [X^{<1>},\ X^{<2>}, \ldots, X^{<n>}]$. Where $X \in R^k$, $k$ refers to the dimension of the word embedding, and $i$ is the index of that word in the lookup table $T^{emb}$. To use these embedded vectors in the models, they must first be mapped to a constant size equal to the length of all sequences in the entire dataset using zero padding.

Due to the variable lengths of the sequences in the dataset, padding is necessary. An important step in the preprocessing stage is the preparation of class labels. It is crucial for the classification to be able to perform two tasks:

i) Domain identification
ii) Polarity identification.

For each class label, a vector of 30 was assigned, where the bits 0-9 represent the domain to which the labels belong, bits 10-19 indicate a positive polarity and bits 20-29 indicate a negative polarity. Depending on the domain and polarity, only two of the 60 bits are assigned a value of 1 while the rest are assigned a value of 0.

The following example illustrates the pre-processing steps for two different sequences.

- Doc1: آویز طراحی جالب و زیبایی داره که توی گردن خیلی خوشگل به نظر میاد. [Domain 10, positive]
- Doc2: گوشی برخلاف ظاهر زیباش به شدت باطری ضعیفی داره و زود شارژ خالی میکنه [Domain 7, negative]

**Step1: Delete stop words and punctuations**:

- Doc1: آویز طراحی جالب زیبای توی گردن خیلی خوشگل نظر. [Domain 10, positive]
- Doc2: گوشی برخلاف ظاهر زیباش شدت باطری ضعیفی زود شارژ خالی [Domain 7, negative]

**Step 2: Create an embedding matrix by Fasttext:** In addition to neural networks, word embeddings are also an important aspect of natural language processing. These embeddings are generated based on word-context relationships in a matrix, where each column is associated with a specific context. Each element in the matrix represents the number of times the corresponding word and context appear together. Common techniques used to generate these dense vectors include GloVe [62], Word2vec [63], BERT, and FastText [64]. These methods typically require a large dataset to develop the embedded vectors. The proposed approach used pre-trained FastText vectors, as shown in Figure 2.

**Step 3**: **Mask labels:** Figure 3 illustrates this process.

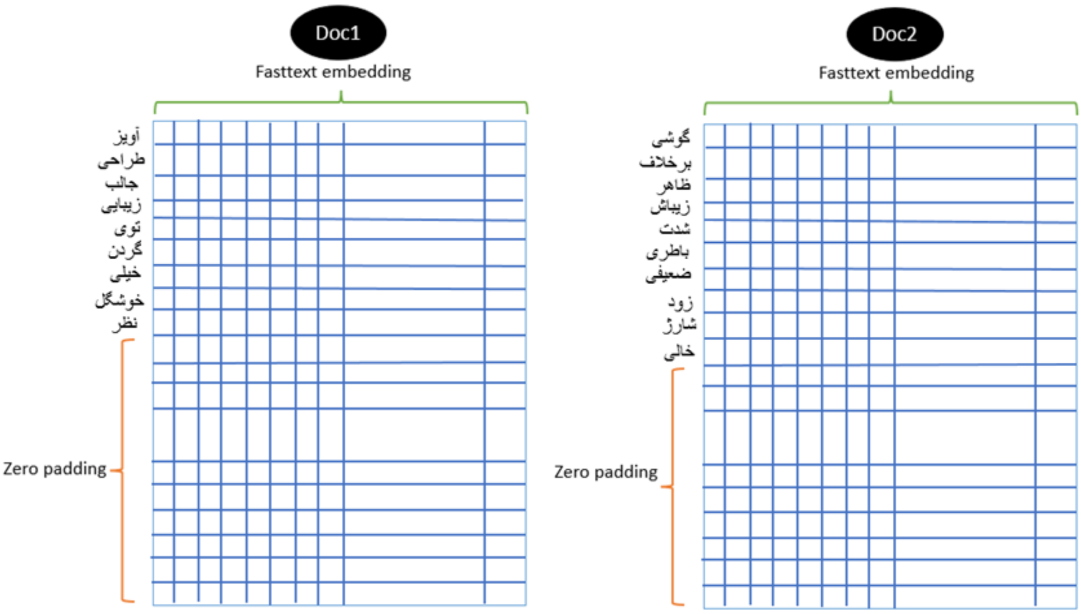


Fig 2. Embedding matrix for Doc1 and Doc2.

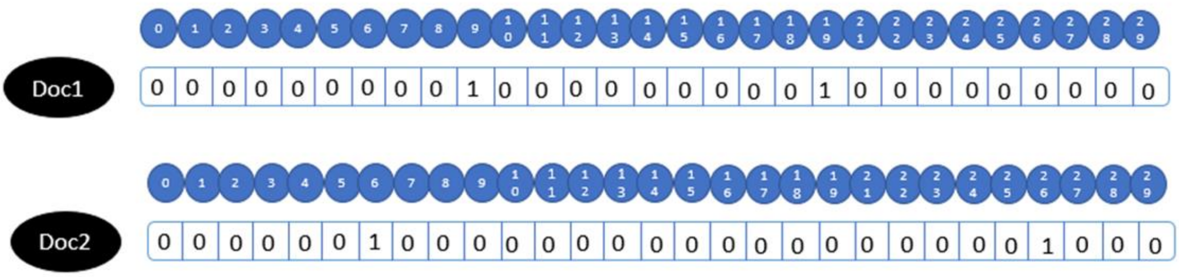


Fig 3.Mask labels for Doc1 and Doc2.

## 3.2 | Network Description

Figure 4 shows the architecture of the Bi-IndRNN Capsule network. The following are the indices, parameters and variables used to construct the proposed model.

Indices:

$i$ Index of word ; $i \in \{1, \ldots, \text{Max length}\}$,

Parameters:

$X_i$ $Word\ i$

$W$ $wehited\ matrix$

$h^t$ hidden vector

$Z^t$ indicator for capsule

$v_j$ Normalizerd capsule

$D_i$ Domain identification

$c_i$ Polarity identification

It is necessary to build the network in four layers:

**1- Word embedding layers:**

The documents within this layer are converted into dense vectors based on the FastText word embedding described in section three. The result of this layer for each document is equivalent to

$$\mathrm{Out}_{\mathrm{embed}} = \mathrm{M} * \mathrm{E} \tag{5}$$

In the present project, E refers to the size of the dense vector, which is equal to 300, while M signifies the maximum document length identified in the whole dataset.

**2- Bi-directional IndRNN layer**: The IndRNN inputs are $Out_{embed}$ that result from the embedding layer. If this matrix is represented by $Out_{embed} = [X_1, X_2, \ldots, X_n]$, then IndRNN input in step $t$ is equal to $X_t \in R^{300}$. Based on these inputs, the sequence of hidden vectors $h^t = [h_1, h_2, \ldots, h_t]$ in the IndRNN is calculated using the following equations:

$$\mathrm{h_t} = \sigma\left(\mathrm{Wx_t} + \mathrm{u} \odot \mathrm{h_{t-1}} + \mathrm{b}\right) \tag{6}$$

In IndRNN for the $n-th$ neuron, the hidden state $h_{(n,t)}$ can be obtained as:

$$h_{(n,t)} = \sigma\left(w_n X_t + u_n h_{(n,t-1)} + b_n\right) \tag{7}$$

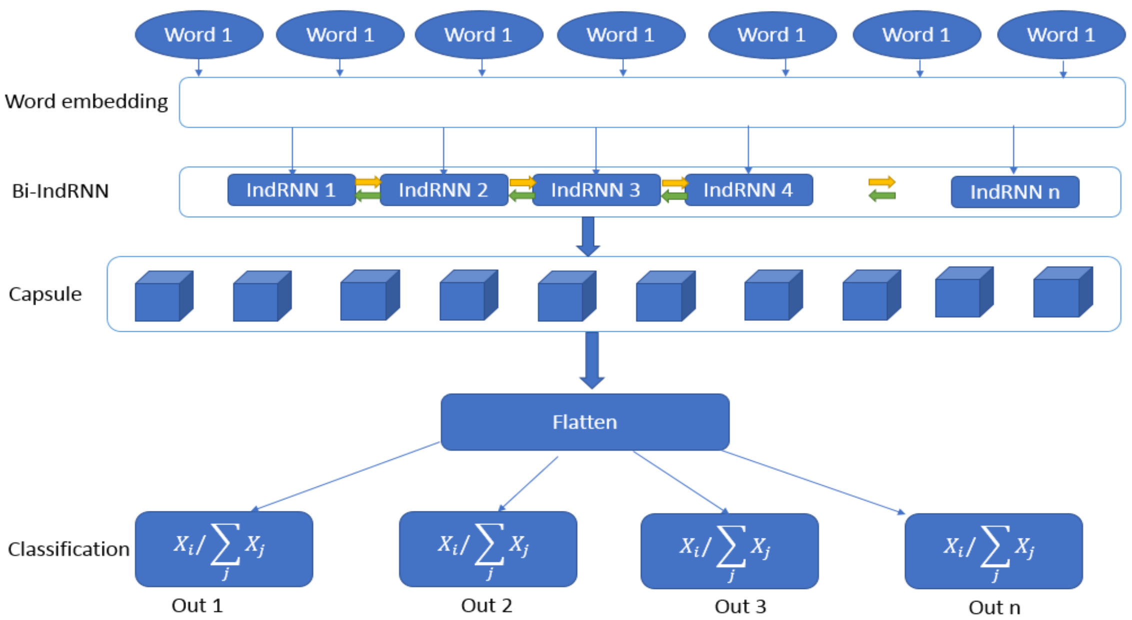


Fig 4. Bi-IndRNN Capsule architecture

The IndRNN model is capable of simulating input data in one direction. In this step, we used the bidirectional mode, which allowed us to use it in two directions.

**3- Capsule layer (Caps):** The capsule layer contains the Bi-IndRNN's encoded features. This layer consists of a collection of capsules. The capsules use the scalar features extracted by the Bi-IndRNN layer and convert them into vector-valued capsules to capture the features of the input sequence. If the Bi-IndRNN output is $h_i$, and $w$ is a weighted matrix, then $\hat{t}_{i|j}$, which represents the predictor vector, is obtained from the following equation:

$$\hat{t}_{i|j} = w_{ij} h_i \quad (8)$$

The set of inputs to a capsule $Z_j$ is a weighting set of all prediction vectors $\hat{t}_{i|j}$, which is computed according to the equation (10).

$$Z_j = \sum_i c_{ij}\, \hat{t}_{i|j} \quad (9)$$

Where $c_{ij}$ is the coupling coefficient, which is iteratively repeatedly by the dynamic routing algorithm [53]. The "squash" function is used as a non-linear function to map the values of the $Z_j$ vectors to [0-1]. This function is applied to $Z_j$ according to the following equation:

$$v_j = \frac{||Z_j||^2}{1 + ||Z_j||^2} \cdot \frac{Z_j}{||Z_j||} \quad (10)$$

The output of a capsule is a vector which, after being selected, is forwarded to another high-level capsule. The proposed architecture uses dynamic routing [53] as the routing protocol.

**4- Classification Layer:**

In contrast to conventional approaches, GRU-Capsule produces different results because the network is designed to learn two different tasks simultaneously - domain identification and polarity identification. The flattened outputs of the capsule layer, represented by F, are then fed into a fully connected layer of 30 neurons.

$$P = W_{dens} * F \quad (11)$$

The output of $P$ should be a representation of the probability of each of the 30 classes. For this purpose, we use the $Softmax$ function, which is calculated for each $fi \in F$ as follows:

$$p_i = \frac{e^{-f_i}}{\sum_{f_j \in f} e^{-f_j}} \quad (12)$$

The identification of the domain is the search for the highest value in the first 10 bits.

$$\text{Domain identification} = \text{Arg}_i \max\{D_i\} \quad (13)$$

Where $D_i = P[0 - 9]$.

To determine the polarity, the highest probability in the state $Negi$ and $Posi$ should be identified for each domain. It is necessary to generate a polarity vector from these values. If $C$ is taken into account, then each component of this vector is obtained from the following statement.

$$c_i = \begin{cases} \mathrm{Pos_i} & \mathrm{poa_i} \geq \mathrm{Neg_i} \\ -\mathrm{Neg_i} & \mathrm{Pos_i} < \mathrm{Neg_i} \end{cases} \tag{14}$$

$i$ is the index that applies to the domain. To obtain the final polarity of the document, the vectors $D$ and $C$ must be multiplied together.

$$\mathrm{Polarity(d)} = \mathrm{D_{(d)}}.\mathrm{C_{(d)}} \tag{15}$$

If $Polarity\ (d)\ \geq 0$, then the polarity is positive otherwise negative.

For a better understanding, let's look at the following values for 3 of the 10 areas of document d:

$D_1(d) = 0.03$

$D_2(d) = 0.75$

$D_3(d) = 0.40$

$C_1(d) = 0.06$

$C_2(d) = 0.08$

$C_3(d) = -0.03$

Then, according to $Arg_i\max\{D_1, D_2, D_3\}$, the domain is $D_2$, and the polarity is positive.

$$Polarity(d) = 0.03 * 0.06 + 0.75 * 0.08 + 0.40 * -0.03 = 0.049 \tag{16}$$

## 4 | Experiments

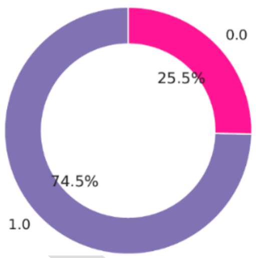


Fig.5. Shoes

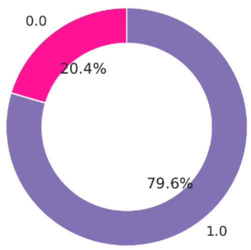


Fig.6. Perfume

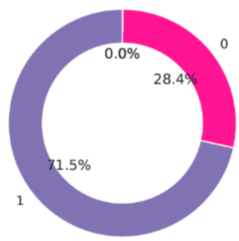


Fig.7.Phone

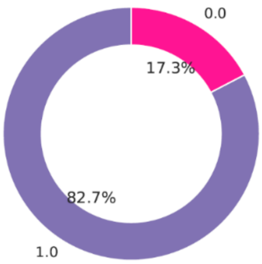


Fig.8.Cold cream

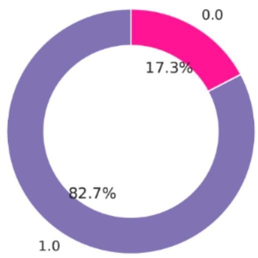


Fig.9.Printer

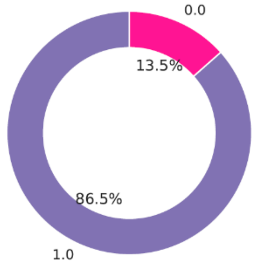


Fig.10. Dress

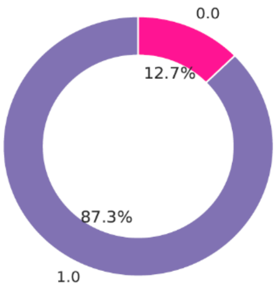


Fig.11. Book

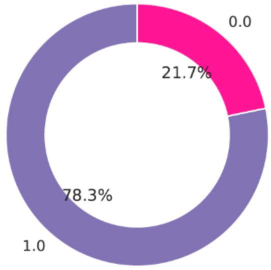


Fig.12.Bed

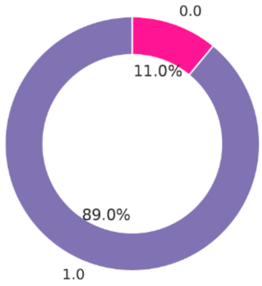


Fig.13.Shaving

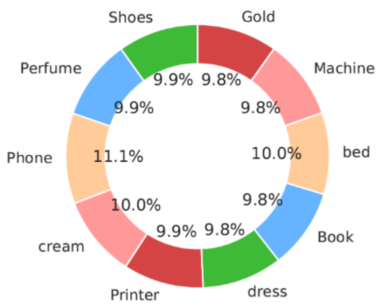


Fig.14. All data domain frequency.

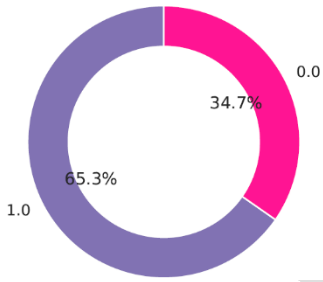


**Fig.15. Jewelry domain labels**

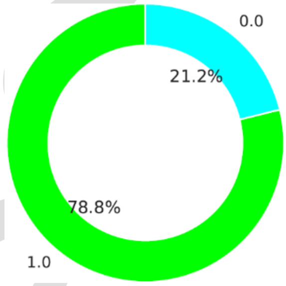


**Fig.16.All domain labels**

## 4.1 | The Dataset

One of the most critical challenges in Persian is the lack of a multi-domain SA evaluation protocol. For this purpose, we collected comments on Digikala's website to create an evaluation protocol. We collected 50799 comments in 10 different domains such as shoes, perfume, phone, cold cream, printer, clothes, books, beds, cars and gold. The labelling process was entirely manual. When choosing the polarity of the comments, we considered comments with a rating of 4 or more to be positive. We also considered comments with a negative rating of less than 2. It should be noted that the data collected is very unbalanced and that the number of negative samples is much smaller than the number of positive samples. Figures 5-16 show the percentage of data frequency per label, data frequency per domain, and labels per domain.

## 4.2 | Implementation & Outcomes.

The Keras library [65] was used to develop the Bi-IndRNN models described in section 3. Keras is a library developed with a high level of programming using Python. This API can run smoothly on both GPU and CPU environments. Keras is compatible with Python versions 2.7-3.x and provides a selection of modules, such as neural layers, cost functions, optimisers, initialisation schemes, activation functions, and regularisation. This API supports TensorFlow[1], CNTK[2], and Theano[3]. Using Keras is simpler to use than its backend, but that doesn't mean it's less flexible.

A confusion matrix is an essential tool for evaluating the performance of a classifier, as it shows the number of correct and incorrect predictions for each class. For a binary classifier, the matrix has two rows and two

[1] https://www.tensorflow.org/

[2] https://cntk.ai/pythondocs/

[3] https://github.com/Theano/Theano

columns and summarises the classification results as true positives, false positives, true negatives and false negatives. Table 2 shows an example of a confusion matrix for a binary classifier.



Table 2. A confusion matrix for a binary classifier

| Class | Classified | |
|---|---|---|
| | POs | Neg |
| POs | TP | FN |
| Neg | FP | TN |

The efficiency of a classifier can be evaluated by applying it to test data, but accuracy is not the only measure used to evaluate its performance. Different of measures can be used to evaluate the efficiency of a classifier. For the evaluation of the proposed model, this paper uses several different metrics such as accuracy, precision, recall, and F1.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{17}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{18}$$

$$\text{F}_1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{19}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{20}$$

$TP$ is the number of positive samples that were correctly classified as positive, while $FP$ is the number of negative samples incorrectly classified as positive. $TN$ is the number of negative samples correctly classified as negative, and $FN$ is the number of positive samples incorrectly classified as negative.

Table 3: The Results Obtained by Different Approaches on Training and Test Data for Polarity Detection.

| | Polarity detection | | | | | |
|---|---|---|---|---|---|---|
| | Accuracy | | Precision | | Recall | |
| Model | Train | Test | Train | Test | Train | Test |
| CNN-Multi Channel [66] | 0.8704 | 0.8775 | 0.8834 | 0.8627 | 0.9331 | 0.9207 |
| Character level CNN [67] | 0.9002 | 0.8823 | 0.9220 | 0.9212 | 0.9312 | 0.9178 |
| NeuroSent [49] | 0.9160 | 0.9183 | 0.9290 | 0.9234 | 0.9222 | 0.9118 |
| Bi-GRUCapsule [68] | 0.9423 | 0.9345 | 0.9231 | 0.9347 | 0.9432 | 0.9336 |
| Bi-IndRNNCapsule | 0.9565 | 0.9489 | 0.9706 | 0.9535 | 0.9744 | 0.9524 |

Table 4: The Results Obtained by Different Approaches on Test Data for Domain Identification.

| | Domain identification | | |
|---|---|---|---|
| Model | Accuracy | Precision | Recall |
| CNN-Multi Channel [66] | 0.6958 | 0.8582 | 0.7255 |
| Character level CNN [67] | 0.7043 | 0.8653 | 0.7589 |
| NeuroSent [49] | 0.7377 | 0.8982 | 0.8290 |
| Bi-GRUCapsule [68] | 0.7809 | 0.8909 | 0.8554 |
| Bi-IndRNNCapsule | 0.8020 | 0.9223 | 0.8829 |

In order to demonstrate the potential of the Bi-IndRNNCapsule for multi-domain sentiment analysis, the Digikala dataset was split into training and test data. We allocated 80% of the data to model training, while the remaining 20% was used for testing. A number of basic models were tested on this data. An overview of each model is given below.

- **CNN-Multi Channel:** In [66], a basic CNN structure is trained with a convolutional layer built on top of word vectors generated by an unsupervised language model. This architecture is both dynamic and static, which determines the training of the model on the embedded vectors. This architecture has been used as another model to compare with the proposed approaches.
- **Character level CNN:** The model processes a sequence of encrypted characters as input. The input language is encrypted by specifying the m-size alphabet and converting each character into a vector using 1-m encryption, also known as "one-hot" encryption. The sequence of characters is then transformed into a long sequence of constant-length vectors of size m and the alphabet used in all models includes 75 characters, including 32 Persian letters, 10 digits, and 33 other characters that are punctuation marks. This model was used for comparison with the proposed models. The structure of this model is given in [67].
- **NeuroSent**: The main idea of this approach [49] is to convert raw input text into an embedded representation, which is then used to predict the polarity of comments. In addition, domain identification is performed in parallel with polarity prediction. The approach uses a skip-gram Word2Vec model to learn embedded words, and LSTM memory cells are at the core of building a deep network. The focus is on identifying common domain features and using them to train the network.
- **Bi-GRUCapsule:** This model is the subject of a paper [68]. This model is very similar to the proposed model, except that in the second step of the model, GRU is used instead of IndRNN. The performance of this model on previous textual and visual data was investigated in [69]. This model is expected to achieve similar results to the proposed approach in some domains.

Table 3 summarises the results obtained by these approaches for polarity detection. The Bi-IndRNNCapsule approach has achieved higher accuracy compared to the other approaches. This correlation between the baseline models further encourages the use of the Bi-IndRNNCapsule model to solve the multi-domain SA problem. In addition, Table 4 shows the domain identification results obtained by these approaches. The Bi-IndRNNCapsule strategy has achieved higher accuracy than alternative approaches. The accuracy is generally low due to the presence of variable domains and linguistic differences between domains. The best result obtained by all approaches in this experiment is 0.8020%.

### 4.3 | Analysis

An analysis was performed to assess the effectiveness of a Bi-IndRNN Capsule approach using fold cross-validation. Table 5 shows the results of the proposed approach on Digikala. Compared to other methods, such as Bi-GRUCapsule, NeuroSent, Char-CNN and Multi-channel CNN, this approach has achieved acceptable results. The Multi-channel CNN model has a lower average accuracy than the other proposed approaches. It is evident that this model has an accuracy of less than 0.96% in all domains, and its average accuracy is 0.8347, the lowest accuracy. On the other hand, the character level model of the CNN achieves greater accuracy than the multi-channel model in most domains. This is shown from a character's point of view. It was shown that NeuroSent, one of the LSTM-based models, achieved relatively good accuracy in a number of domains. Despite this, it produced a lower average accuracy than both the Bi-GRUCapsule and Bi-IndRNNCapsule models. The Bi-GRUCapsule model performs similarly to the Bi-IndRNNCapsule model in many areas. There is no doubt that the Bi-IndRNNCapsule model is more accurate than other approaches in most domains. It was found that this model achieved an average accuracy of 0.9144, which is 0.07% better than the weakest approach and 0.03% better than the Bi-GRUCapsule approach.

Table 5. Accuracy of the data domain.

| Domain | NeuroSent | Multi-channel CNN | Character level CNN | Bi-GRUCapsule | Bi-IndRNNCapsule |
|---|---|---|---|---|---|
| Shoes | 0.8717 | 0.8232 | 0.8514 | 0.8692 | 0.9134 |
| Perfume | 0.8837 | 0.8122 | 0.8420 | 0.8864 | 0.8917 |
| Phone | 0.8718 | 0.8105 | 0.8097 | 0.8693 | 0.9232 |
| Cold cream | 0.8650 | 0.8402 | 0.8728 | 0.8652 | 0.9291 |
| Printer | 0.8866 | 0.8054 | 0.8093 | 0.8915 | 0.9272 |
| Dress | 0.8996 | 0.8434 | 0.8714 | 0.8762 | 0.9075 |
| Book | 0.8941 | 0.8455 | 0.8700 | 0.8926 | 0.9094 |
| Bed | 0.8911 | 0.8612 | 0.8543 | 0.8911 | 0.9114 |
| Shaving machine | 0.9086 | 0.8491 | 0.8704 | 0.8806 | 0.9125 |
| Jewellery | 0.8890 | 0.8567 | 0.8589 | 0.8828 | 0.9193 |
| Average | 0.8861 | 0.8347 | 0.8510 | 0.8804 | 0.9144 |

### 4.4 | Discussion

Several approaches have been proposed in the literature for the analysis of Persian multi-domain SAS. Many of these approaches have been developed in the context of a specific domain. The proposed approach focuses on ten different domains and uses a combination of Bi-IndRNN and CapsuleNet for both polarity detection and domain identification. Bi-IndRNN extracts features while CapsuleNet learns the extracted features. A comparison of the efficiency of the proposed algorithm with four other approaches shows that it has a higher level of accuracy in both tasks.

## 5 | Error Analysis and Future Works

Figures 17-26 display the TP, FP, FN, and TN rates obtained from the proposed Bi-IndRNN+Capsule approach on the test data. As shown in Figures 17-26, the Shoes and Perfume domain exhibit the highest FP rate, while the Perfume and Dress domain display the highest $TN$ rate.

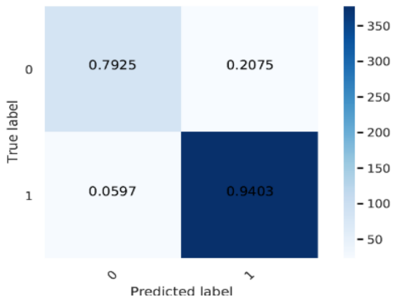


Fig.17.Book

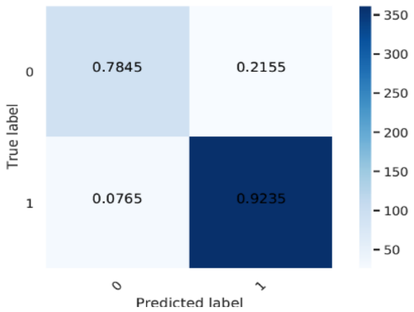


Fig.18.Perfume

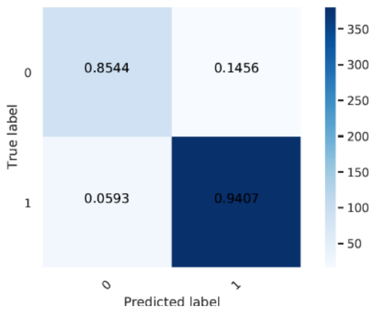


Fig.19.Phone

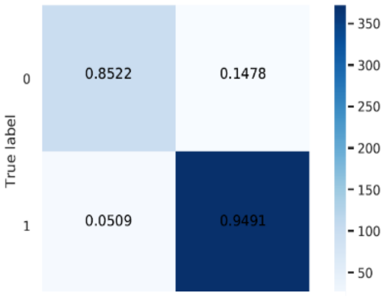


Fig.20.Printer

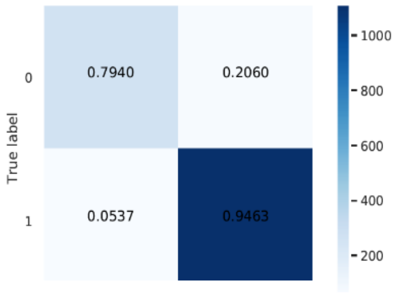


Fig.21.Shaving machine

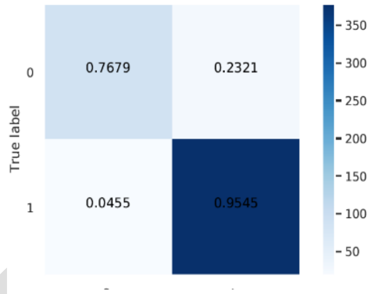


Fig.22. Shoes

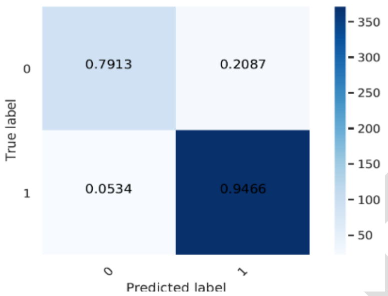


Fig.23. Bed

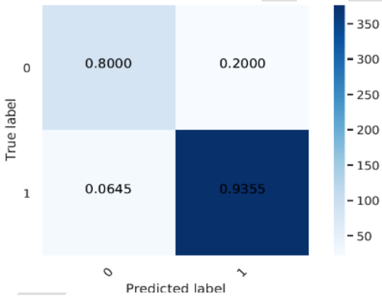


Fig.24. Dress

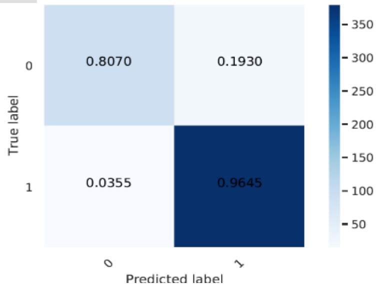


Fig.25.Cold cream

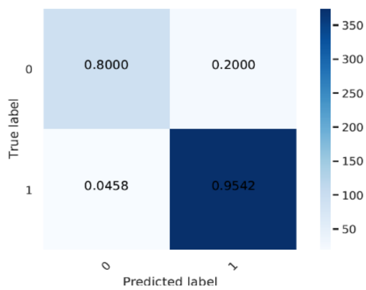


Fig.2. Jewellery

In a review of the 100 comments that were incorrectly categorised, we concluded that the presence of noise in the data and the lack of recognition of the scope of negation caused this error. In addition, the use of non-slang language was another reason for this error. Therefore, our next goal for future work will be to manually integrate negation scope as a feature with the automatically selected features from the Bi-IndRNNCapsule as part of our future work. The main objective of our proposal is to improve the Bi-IndRNN approach by adding more subtasks, which include:

1. Applying some more pre-processing steps for denoising, such as replacing some irregular forms of words with their correct forms, for example, using "خلی" "instead of "خیییلی" "."Use optimisation approaches to optimise the proposed network parameters.
2. Use a sarcasm detection algorithm.

## 6| Conclusion

This paper presents a hybrid approach based on Bidirectional-IndRNN (Bi-IndRNN) and CapsuleNet for multi-domain sentiment analysis. This method proposed the use of Bi-IndRNN to extract the features and CapsuleNet to learn them. This model was implemented for two different tasks, namely domain detection and polarity detection. The proposed approach was evaluated using data obtained from the Digikala website. The dataset consisted of 10 different domains, which were divided into positive and negative polarities. The main results of this study are:

1. In the domain detection mode, Bi-IndRNN achieved an accuracy of 0.8020.

2. In polarity detection, this approach achieved 0.9565.

3. Bi-IndRNN was compared with 5 other approaches that were more accurate in both tasks.

4. One of the most critical problems of the proposed approach was ignoring the negation range of words, and solutions were proposed to improve future work.

The main limitation of this study is the lack of ability of the proposed method to deal with the negation range. For this reason, it is recommended to use methods that can control the negation scope. The author of [4] used 19 negative words with 90 patterns to detect the negation scope of the sentence, which resulted in better classification accuracy for the classifier. In addition, similar patterns can improve the proposed approach of Bi-IndRNN.

## References


[1]. Yadav, A., & Vishwakarma, DK. (2019, December 02). Sentiment analysis using deep learning architectures: a review. *Artificial Intelligence Review,* 53(6), 4335–4385. https://doi.org/10.1007/s10462-019-09794-5

[2]. Syed, TA., Mehmood, F., & Qaiser, T. (2023, July). Brand–SMI collaboration in influencer marketing campaigns: A transaction cost economics perspective. *Technological Forecasting and Social Change*, 192, 122580. https://www.sciencedirect.com/science/article/pii/S0040162523002652?via%3Dihub

[3]. Birjali, M., Kasri, M., & Beni-Hssane, A. (2021,August 17). A comprehensive survey on sentiment analysis: Approaches, challenges and trends. *Knowledge-based Systems*, 226, 107134. https://doi.org/10.1016/j.knosys.2021.107134

[4]. Torabian, B.(2016, September 14). Sentiment classification with case-based approach. *Item–Theses Canada. Library-Archives.canada.ca.* https://library-archives.canada.ca/eng/services/services-libraries/theses/Pages/item.aspx?idNumber=1132135615

[5].Laddha, M., Gaikwad, H., Akarte, H., &Gandhi, S.(2023, February). CognitiveScience: An Insightful Approach. *in Proceedings of International Conference on Data Science and Applications: ICDSA 2022*, *2,*(pp. 199-208). Retrieved from
https://www.researchgate.net/publication/368309892_Cognitive_Science_An_Insightful_Approach

[6]. Blasch, E., Pham, T., Chong, C.Y., Koch, W., Leung, H., Braines, D., & Abdelzaher, T. (2021, July 01). Machine Learning/Artificial Intelligence for Sensor Data Fusion–Opportunities and Challenges. *IEEE Aerospace and Electronic Systems Magazine*, *36*(7), 80–93. https://doi.org/10.1109/maes.2020.3049030

[7] Novak, A., Bennett, D., & Kliestik, T. (2021,June 25). Product Decision-Making Information Systems, Real-Time Sensor Networks, and Artificial Intelligence-driven Big Data Analytics in Sustainable Industry 4.0. *Economics, Management, and Financial Mrkets*, *16*(2), 62–73. https://doi.org/10.22381/emfm16220213

[8]. Zhang, Z., Ning, H., Shi, F., Farha, F., Xu, Y., Xu, J., Zhang, F., & Choo, K.-K.R. (2021, March 13). Artificial intelligence in cyber security: research advances, challenges, and opportunities. *Artificial Intelligence Review*, *55*(2), 1029–1053. https://doi.org/10.1007/s10462-021-09976-0

[9]. Xiang, X., Li, Q., Khan, S., & Khalaf, O.I. (2020, November 12). Urban water resource management for sustainable environment planning using artificial intelligence techniques. *Environmental Impact Assessment Review*, 86, 106515. https://doi.org/10.1016/j.eiar.2020.106515

[10]. Zhan, Z., Tong, Y., Lan, X., & Zhong, B. (2022, September 14). A systematic literature review of game-based learning in Artificial Intelligence education. *Interactive Learning Environments*, 1–22. From https://doi.org/10.1080/10494820.2022.2115077

[11]. Tan, T.E., Anees, A., Chen, C., Li, S., Xu, X., Li, Z., Xiao, Z., Yang, Y., Lei, X., Ang, M., Chia, A., Lee, S Y., Wong, E.Y.M., Yeo, I.Y.S., Wong, Y.L., Hoang, Q.V., Wang, Y.X., Bikbov, M.M., Nangia, V., & Jonas, J.B. (2021, May). Retinal photograph-based deep learning algorithms for myopia and a blockchain platform to facilitate artificial intelligence medical research: a retrospective multicohort study. *The Lancet Digital Health*, *3*(5), e317–e329. https://doi.org/10.1016/S2589-7500(21)00055-8

[12]. Babu, N.V., & Kanaga, E.G.M. (2021, November 19). Sentiment Analysis in Social Media Data for Depression Detection Using Artificial Intelligence: A Review. *Springer Nature Computer Science*, *3*(1),1-20. https://doi.org/10.1007/s42979-021-00958-1

[13]. Singhal, A., & Sharma, D. (2021, March 19). Keyword extraction using Renyi entropy: A statistical and domain independent method. *In 2021 International Conference on Advanced Computing* , Coimbatore, India,(pp.1-5). IEEE. https://ieeexplore.ieee.org/document/9441909

[14]. Bayer, M., Kaufhold, M.A., & Reuter, C. (2022, December 15). A Survey on Data Augmentation for Text Classification. *Association for Computing Machinery Computing Surveys*, *55*(7), 1–39. https://doi.org/10.1145/3544558

[15]. Ha, S., & Grubert, E. (2023,March 07). Hybridizing qualitative coding with natural language processing and deep learning to assess public comments: A case study of the clean power plan. *Energy Research & Social Science*, *98*, 103016. https://doi.org/10.1016/j.erss.2023.103016

[16]. Ghorbanali, A., & Sohrabi, M.K. (2023, February 20). Exploiting bi-directional deep neural networks for multi-domain sentiment analysis using capsule network. *Multimedia Tools and Applications, 82*, 22943–22960. https://doi.org/10.1007/s11042-023-14449-3

[17]. Yuan, Z., Wu, S., Wu, F., Liu, J., & Huang, Y. (2018, September 01). Domain attention model for multi-domain sentiment classification. *Knowledge-Based Systems*, *155*, 1–10. doi:https://doi.org/10.1016/j.knosys.2018.05.004

[18] .Shepelenko, O., Hadachi, A., & Lind, A. (2017, May18). Opinion Mining and Sentiment Analysis using Bayesian and Neural Networks Approaches. *Univerity of Tartu, Estonia*, Retrieved April 25, 2023, from https://core.ac.uk/download/pdf/237084898.pdf

[19]. Hailong, Z., Wenyan, G., & Bo, J. (2014, September 12). Machine Learning and Lexicon Based Methods for Sentiment Classification: A Survey. *Paper presented at the 2014 11th Web Information System and Application Conference.* Tianjin, China,(pp. 262-265). https://doi.org/10.1109/wisa.2014.55

[20]. Cambria, E. (2016,March17). Affective Computing and Sentiment Analysis. *Institute of Electrical and Electronics Engineers Intelligent Systems*, *31*(2), 102–107. https://doi.org/10.1109/mis.2016.31

[21]. Liu, B., & Zhang, L. (2012, January 01). A Survey of Opinion Mining and Sentiment Analysis. *In Mining Text Data* . 415-463. Springer, Boston, MA.https://doi.org/10.1007/978-1-4614-3223-4_13

[22]. Vohra, M., & Teraiya, P. (2013, November 12). A comparative study of sentiment analysis techniques. *Journal of Information, Knowledge and Research in Computer Engineering*. Retrieved from http://www.ejournal.aessangli.in/aseejournals/ce63.pdf

[23]. Pang, B., Lee, L., & Vaithyanathan, S. (2002, July 06). Thumbs up?: sentiment classification using machine learning techniques.*Proceedings of the ACL-02 Conference On Empirical Methods In Natural Language Processing - EMNLP '02*. Stroudsburg, USA, *10*, (pp.79–86) .https://doi.org/10.3115/1118693.1118704

[24].Ye, Q., Zhang, Z., & Law, R. (2009, April 01). Sentiment classification of online reviews to travel destinations by supervised machine learning approaches. *Expert Systems with Applications*, *36*(3), 6527–6535. https://doi.org/10.1016/j.eswa.2008.07.035

[25]. Zhang, Z., Ye, Q., Zhang, Z., & Li, Y. (2011, June). Sentiment classification of Internet restaurant reviews written in Cantonese. *Expert Systems with Applications*, *38*(6), 7674–7682. https://doi.org/10.1016/j.eswa.2010.12.147

[26].Prabowo, R., & Thelwall, M. (2009,April). Sentiment analysis: A combined approach. *Journal of Informetrics*, *3*(2), 143–157. https://doi.org/10.1016/j.joi.2009.01.003

[27].Riloff, E., Patwardhan, S., & Wiebe, J. (2006, July). Feature subsumption for opinion analysis. *Proceedings Of The 2006 Conference On Empirical Methods In Natural Language Processing*. Sydney, Australia,(pp.440-448). Retrieved from https://aclanthology.org/w06-1652.pdf

[28]. Whitelaw, C., Garg, N., & Argamon, S. (2005, October). Using appraisal groups for sentiment analysis. *Proceedings of the 14th ACM International Conference on Information and Knowledge Management - CIKM '05*. (pp.625–631). https://doi.org/10.1145/1099554.1099714

[29]. Medhat, W., Hassan, A., & Korashy, H. (2014, December). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, *5*(4), 1093–1113. https://doi.org/10.1016/j.asej.2014.04.011

[30]. Turney, P.D. (2002, December 11). Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics,* Philadelphia, Pennsylvania, (pp.417-424).*ArXiv:cs*/0212032. https://arxiv.org/abs/cs/0212032

[31].Church, K., & Hanks, P. (1990, March 01). Word association norms, mutual information, and lexicography. *Computational Linguistics*, *16*(1), 22–29. https://doi.org/10.5555/89086.89095

[32]. Fernández-Gavilanes, M., Álvarez-López, T., Juncal-Martínez, J., Costa-Montenegro, E., & González-Castaño, F.J. (2015, June 1). GTI: An Unsupervised Approach for Sentiment Analysis in Twitter. *In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)* , Denver, Colorado. *ACLWeb; Association for Computational Linguistics*. 533–538. https://doi.org/10.18653/v1/S15-2089

[33]. Taras, Z., & Carroll, J.M. (2008,August18). Automatic seed word selection for unsupervised sentiment classification of Chinese text. *COLING '08:Proceedings of the 22nd International Conference on Computational Linguistics. 1*, (pp.1073–1080). https://doi.org/10.3115/1599081.1599216

[34]. Otter, D.W., Medina, J.R., & Kalita, J.K. (2020, April 21). A Survey of the Usages of Deep Learning for Natural Language Processing. *Institute of Electrical and Electronics Engineers Transactions on Neural Networks and Learning Systems*, *32*(2), 1–21. https://doi.org/10.1109/TNNLS.2020.2979670

[35]. Zhang, Y., Meng Joo Er, Venkatesan, R., Wang, N., & Mahardhika, P. (2016, July 24). Sentiment classification using Comprehensive Attention Recurrent models. *2016 International Joint Conference on Neural Networks.*Vancouver, BC, Canada,(pp.1562-1569). https://doi.org/10.1109/ijcnn.2016.7727384

[36].Vyas, A., & Nextlabs, M. (2017, April 19). Deep Learning in Natural Language Processing. *Retrieved April 26, 2023*, from https://www.mphasis.com/content/dam/mphasis-com/global/en/downloads/whitepaper/Deep-Learning-In-Natural-Language.pdf

[37].LeCun, Y., Haffner, P., Bottou, L., & Bengio, Y. (1999, October 22). Object Recognition with Gradient-Based Learning. *Shape, Contour and Grouping in Computer Vision*, *1681*, 319–345. https://doi.org/10.1007/3-540-46805-6_19

[38]. Tompson, J., Jain, A., Lecun, Y., & Bregler, C. (2014, June 11). Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation. *Advances In Neural Information Processing Systems, volume 27.*Retrieved April 26, 2023, from https://jonathantompson.github.io/others/joint-training-convolutional.pdf

[39]. Kim, H., Jung, W.K., Park, Y.C., Lee, J.W., & Ahn, S.H. (2022, February). Broken stitch detection method for sewing operation using CNN feature map and image-processing techniques. *Expert Systems with Applications*, *188*, 116014. https://doi.org/10.1016/j.eswa.2021.116014

[40]. Deng, L.(2014, November 21). 2014 Index IEEE/ACM Transactions on Audio, Speech, and Language Processing Vol.22.*IEEE/ACM Transactions on Audio, Speech, and Language Processing, 22*(12), 2264–2288. https://doi.org/10.1109/taslp.2014.2373645

[41]. Yang, H., & Schell, K.R. (2022, January). GHTnet: Tri-Branch deep learning network for real-time electricity price forecasting. *Energy*, *238*, 122052. https://doi.org/10.1016/j.energy.2021.122052

[42] Kim, Y. (2014, September 03). Convolutional neural networks for sentence classification. *ArXiv preprint arXiv*:1408.5882. Retrieved from https://arxiv.org/abs/1408.5882

[43]. Rojas-Barahona, L.M. (2016, December 29). Deep learning for sentiment analysis. *Language and Linguistics Compass*, *10*(12), 701–719. https://doi.org/10.1111/lnc3.12228

[44]. Zeiler, M.D., & Fergus, R. (2013, January 16). Stochastic Pooling for Regularization of Deep Convolutional Neural Networks. *ArXiv*:1301.3557 [Cs, Stat]. https://arxiv.org/abs/1301.3557

[45]. He, K., Zhang, X., Ren, S., & Sun, J. (2014, September 06). Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *European Conference on Computer Vision 2014* Springer. (pp.346-361). https://doi.org/10.1007/978-3-319-10578-9_23

[46]. Ouyang, W., Luo, P., Zeng, X., Qiu, S., Tian, Y., Li, H., Yang, S., Wang, Z., Xiong, Y., Qian, C., Zhu, Z., Wang, R., Loy, C.C., Wang, X., & Tang, X. (2014, September 11). DeepID-Net: Multi-stage and deformable deep convolutional neural networks for object detection. *ArXiv.org*. https://doi.org/10.48550/arXiv.1409.3505

[47]. Lipton, Z.C., Berkowitz, J., & Elkan, C. (2015, October 17). A Critical Review of Recurrent Neural Networks for Sequence Learning. *ArXiv. org*. https://arxiv.org/abs/1506.00019

[48]. Hochreiter, S., & Schmidhuber, J. (1997, January 01). Long Short-Term Memory. *Neural Computation*, *9*(1), 1–42. https://doi.org/10.1162/neco.1997.9.1.1

[49]. Dragoni, M., & Petrucci, G. (2017,June 21). A Neural Word Embeddings Approach for Multi-Domain Sentiment Analysis. *Institute of Electrical and Electronics Engineers Transactions on Affective Computing*, *8*(4), 457–470. https://doi.org/10.1109/taffc.2017.2717879

[50]. Bahdanau, D., Cho, K., & Bengio, Y. (2014,December19). Neural Machine Translation by Jointly Learning to Align and Translate. *ArXiv.org*. https://arxiv.org/abs/1409.0473

[51]. Li, S., Li, W., Cook, C., Zhu, C., & Gao, Y. (2018, June 18). Independently Recurrent Neural Network (IndRNN): Building A Longer and Deeper RNN. *2018 Institute of Electrical and Electronics Engineers and Conference on Computer Vision and Pattern Recognition,* Salt Lake City, UT, USA, 2018, (pp.5457-5466 ).https://doi.org/10.1109/CVPR.2018.00572

[52]. Tai, K. S., Socher, R., & Manning, C.D. (2015, May 30). Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. *ArXiv*:1503.00075 [Cs]. https://arxiv.org/abs/1503.00075

[53]. Sabour, S., Frosst, N., & Hinton, G. (2017, November 7). Dynamic routing between capsules. *Proceedings of the 31st Neural Information Processing Systems Conference,*Long Beach, CA, USA, Retrieved April 26, 2023, from https://proceedings.neurips.cc/paper/2017/file/2cad8fa47bbef282badbb8de5374b894-Paper.pdf

[54]. Bhowmik, N.R., Arifuzzaman, M., & Mondal, MRH. (2022, January 11). Sentiment analysis on Bangla text using extended lexicon dictionary and deep learning algorithms. Array, *Volum13,* 100123. https://doi.org/10.1016/j.array.2021.100123

[55].Yang, K., & Liu, J. (2021, November 12). Weibo Sentiment Analysis Based on Advanced Capsule Network. *Paper presented at the2021 International Symposium on Computer Science and Intelligent Controls,* Rome, Italy. (pp.211-216). doi:https://doi.org/10.1109/iscsic54682.2021.00046

[56]. Obied, Z., Solyman, A., Fat'hAlalim, A., & Alsayed, A. (2021, February 26). BERT Multilingual and Capsule Network for Arabic Sentiment Analysis. *Paper presented at the2020 International Conference on Computer, Control, Electrical, and Electronics Engineering* ,Khartoum, Sudan,(pp.*1-6).* https://doi.org/10.1109/iccceee49695.2021.9429568

[57] .Habib, M.K. (2021, January 20). The challenges of Persian user-generated textual content: A, Machine Learning-Based Approach. *ArXiv preprint arXiv:2101.08087*. https://arxiv.org/abs/2101.08087

[58]. Akhoundzade, R., & Hashemi Devin, K. (2019, October 24). Persian sentiment lexicon expansion using unsupervised learning methods. *Paper presented at 9th International Conference on Computer and Knowledge Engineering ,* Mashhad, Iran, 2019, (pp. 461-465). doi: https://doi.org/10.1109/iccke48569.2019.8964692.

[59]. Asgarian, E., Kahani, M., & Sharifi, S. (2017, November 07). The Impact of Sentiment Features on the Sentiment Polarity Classification in Persian Reviews. *Cognitive Computation*, *10*(1), 117–135. https://doi.org/10.1007/s12559-017-9513-1

[60]. Basiri, M.E., Kabiri, A., Abdar, M., Mashwani, W.K., Yen, N.Y., & Hung, J.C. (2019, October 13). The effect of aggregation methods on sentiment classification in Persian reviews. *Enterprise Information Systems*, *14*(9-10), 1394–1421. https://doi.org/10.1080/17517575.2019.1669829

[61]. Vamerzani, H.A., & Khademi, M. (2015, January). Increase Business Intelligence Based on opinions mining in the Persian Reviews. *International Academic Journal of Science and Engineering*, *2*(2), 164–174. https://www.iaiest.com/abstract.php?Id=7&archiveid=1109

[62]. Pennington, J., Socher, R., & Manning, C. (2014, January). GloVe: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* , Doha, Qatar.(pp.1532–1543).https://doi.org/10.3115/v1/d14-1162

[63]. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013, September 07). Efficient Estimation of Word Representations in Vector Space. *ArXiv.org*. https://arxiv.org/abs/1301.3781

[64]. Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016, August 09). Bag of tricks for efficient text classification. *ArXiv preprint arXiv:1607.01759*. https://arxiv.org/abs/1607.01759

[65]. Chollet, F. (2015) .keras, GitHub. - References - *Scientific Research*. https://github.com/fchollet/keras

[66]. Zhang, Y., & Wallace, B. (2016, April 06). A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. *ArXiv.org*. https://arxiv.org/pdf/1510.03820.pdf

[67]. Zhang, X., Zhao, J., & LeCun, Y. (2016, April 04). Character-level Convolutional Networks for Text Classification. *ArXiv:1509.01626 [Cs]*. https://arxiv.org/abs/1509.01626

[68]. Khayi, N.A., & Rus, V. (2019, August 04). BI-GRU Capsule Networks for Student Answers Assessment. *In 2019 KDD Workshop on Deep Learning for Education*, Retrieved April 26, 2023, from http://ml4ed.cc/attachments/Khayi.pdf

[69]. Mousa, R., Nazari, S., Abadi, A.K., Shoukhcheshm, R., Pirzadeh, M.N., & Safari, L. (2021, February 15). Ti-Capsule: Capsule Network for Stock Exchange Prediction. *ArXiv:2102.07718 [cs]*. https://doi.org/10.48550/arxiv.2102.07718