Multimedia Tools and Applications
https://doi.org/10.1007/s11042-022-12155-0

Deep text clustering using stacked AutoEncoder

Soodeh Hosseini[1,2] • Zahra Asghari Varzaneh[1,2]

Received: 10 June 2021 / Revised: 4 September 2021 / Accepted: 3 January 2022
Published online: 16 February 2022

Abstract
Text data is a type of unstructured information, which is easily processed by a human, but it is hard for the computer to understand. Text mining techniques effectively discover meaningful information from text, which has received a great deal of attention in recent years. The aim of this study is to evaluate and analyze the comments and suggestions presented by Barez Iran Company. Barez is an unlabeled dataset. Extracting useful information from unlabeled large textual data by human to manually be very difficult and time consuming. Therefore, in this paper we analyze suggestions presented in Persian using BERTopic modeling for cluster analysis of the dataset. In BERTopic, each document belongs to a topic with a probability distribution. As a result, seven latent topics are found, covering a broad range of issues such as Installation, manufacture, correction, and device. Then we propose a novel deep text clustering based on hybrid of a stacked autoencoder and k-means clustering to organize text documents into meaningful groups for mining information from Barez data in an unsupervised method. Our data clustering has three main steps: 1) Text representation with a new pre-trained BERT model for language understanding called ParsBERT, 2) Text feature extraction based on based on a new architecture of stacked autoencoder to reduce the dimension of data to provide robust features for clustering, 3) Cluster the data by k-means clustering. We employ the Barez dataset to verify our work's effectiveness; Silhouette Score is used to evaluate the resulting clusters with the best value of 0.60 with 3 clusters grouping. Experimental evaluations demonstrate that the proposed algorithm clearly outperforms other clustering methods.

Keywords AutoEncoder · Clustering · BERT word embedding · Topic modelling

✉ Soodeh Hosseini
so_hosseini@uk.ac.ir

Zahra Asghari Varzaneh
z.asghari@math.uk.ac.ir

1 Department of Computer Science, Faculty of Mathematics and Computer, Shahid Bahonar University of Kerman, Kerman, Iran

2 Mahani Mathematical research center, Shahid Bahonar University of Kerman, Kerman, Iran

1 Introduction

With the advancement of technology, the amount of generated data is increasing, which nearly 80% of the world's data is in unstructured textual form. Therefore, it has become essential to develop efficient and effective techniques to discover useful and interesting patterns from a large amount of textual data. Text mining techniques effectively discover meaningful information from text, which has received a great deal of attention in recent years. Text mining is the process of Natural Language Processing (NLP) to extract and analyze valuable insights from unstructured text data and transforming unstructured documents into structured data. Text mining combines concepts of linguistics, statistics, machine learning, and deep learning to learn training data and prediction results on new information based on their previous experience [36].

In NLP, text data preprocessing is traditionally the first and most important step in the process of building Machine Learning or Deep learning models. In this step, the given input document is processed to convert the document contents in numerical form to make them mathematically computable and remove any unnecessary or unwanted data. Various text preprocessing steps are widely applied for dimensionality reduction. In different NLP tasks, the vector representation of words is very useful to take the semantic meaning of words. Each word/term weight in the Vector Space Model (VSM) is an axis/dimension. The text/document is represented as a vector in the multi-dimensional space. Clustering techniques' efficiency is affected by the uninformative features and the size of dimension space [5]. Turian et al. discussed unsupervised methods to obtain word features before utilizing them in supervised approaches [48].

Word embedding is a type of word representation for text where words with similar meanings have a similar representation. Each word is represented as a real-valued vector, often tens or hundreds of dimensions. Actually, embeddings are dense vector dimensional representations of a point in a higher dimensional vector space. Also, word embeddings can learn some interesting traits about words in a vocabulary [4]. Pre-trained word representations are a key component in many neural language understanding models [27, 37]. In this paper, we applied a BERT (Bidirectional Encoder Representations from Transformers) to extract features, namely word and sentence embedding vectors, from text data [12]. BERT, released in late 2018, is a pre-training language representation method, which supports transfer learning and fine-tuning on specific tasks based on Transformer's encoder and represents a general language modeling. Farahani et al. [14], propose a monolingual BERT for the Persian language called ParsBERT. This model is lighter than multilingual BERT and improves state-of-the-art results by outperforming other prior works in different tasks.

Text feature extraction is a selection of text feature item from a text message to reduce the dimension of feature space [46]. As a new feature extraction method, deep learning can automatically learn feature representation from big data, including millions of parameters. Autoencoders were first introduced in the 1980s by Rumelhart et al. [42]. Autoencoder is a back propagation network that can learn without a teacher (unsupervised learning), by using the input data as the teacher to dimensionality reduction or manifold learning.

A stacked autoencoder is the deep autoencoder, which is built by stacking up layers. In every layer, the input is the learned representation of the former layer, and it learns a more compact representation of the existing learned representation. Gravelines et al. [17], proposed stacked sparse autoencoder. It is a stacked autoencoder where sparsity regularizations are introduced into the autoencoder to learn a sparse representation.

Topic modeling is an unsupervised text clustering that groups the text into categories; it's just that the categories are not known before-hand [3, 49]. Topic models provide a process for organizing, classifying, and managing documents. Topic models are a set of probabilistic models in which each document can be represented by distribution over topics. Each topic has a set of keywords, with each keyword in the set having a probability of occurrence for the subject topic. Different topics have their own sets of keywords with corresponding probabilities, and topics may share some keywords, but most likely with different probabilities [6]. Topic modeling could be effectively used in Barez dataset. In this paper, we proposed to apply BERTopic modeling for cluster analysis of the dataset. The application of the topic modeling approach to cluster analysis of large datasets can greatly improve subgroup identification accuracy.

Cluster analysis is a technique to explore the relationships between attributes and samples that assign samples or attributes to clusters based on their similarity [47]. Text clustering can be used as a preliminary method for grouping similar texts and documents in the same group (cluster) that are more similar to each other than to those in other clusters. This algorithm receives a set of unlabeled texts and returns the list of detected clusters. Each cluster is assigned a descriptive name indicating the cluster's relative importance with respect to all clusters. Each document may be assigned to one or several clusters. Both Text clustering and topic modeling are unsupervised tasks that attempt to organize documents for better information retrieval. On the other hand, Text clustering forms similar clusters of these documents, but in topic modeling, we don't determine document similarity. Instead, we treat a document as a mixture of topics in which a topic is a probability distribution of words.

Our motivation of doing this research is to propose a new method for clustering unlabeled text data in Persian. The raw data used in this paper are suggestions in Persian to increase the quality of Barez's products, which exceeds 27,000 suggestions. For this purpose, at first, we used BERTopic modeling for cluster analysis of the dataset. In BERTopic, each document belongs to a topic with a probability distribution. Furthermore, a new pre-trained language representation model called ParsBERT is used to represent Persian textual data as numerical features. Furthermore, an autoencoder is proposed with a new architecture to reduce the dimension of the data. Finally, the extracted features are used to cluster the textual data used in this research. The proposed autoencoder is designed to produce the best clusters based on the Silhouette evaluation criteria. This paper uses several different methods such as fast-text and tf-idf to represent text data and the features extracted from these models are used for k-means clustering. The results of comparing these models with the proposed method shows that the efficiency of the proposed method is better than other models. Also, the proposed method has acceptable performance against state-of- the- art algorithms.

Our main contributions of this paper can be summarized as follows:

- We apply a pre-trained language representation model for the Barez dataset in the Persian Language, called ParsBERT.
- In this paper, we propose a Deep Stack Autoencoder based feature extraction method which extracts discriminative subset of features by eliminating irrelevant features to improve clustering efficiency.
- We group the extracted features with the k-means clustering algorithm and determine the optimal clusters.

- We apply BERTopic modeling for cluster analysis of the dataset. BERTopic organizes and manages documents and builds a set of probabilistic models in which that each document can be represented by distribution over topics.

The rest of this paper is organized as follows: In section 2, we present the related works. In section 3, we discuss Theoretical Background in detail. In section 4, we introduce the proposed method. Section 5 and 6 present a series of experiments and the conclusion of the paper, respectively.

2 Related work

In recent years, the increasing amount of digital text information in several applications and Internet web pages has affected the text analysis process. Text mining using various techniques such as Natural Language Processing (NLP), statistical analysis, and data mining aims to mine helpful, attention grabbing, and non-trivial information or patterns from unstructured data. Textual Data Mining (TDM) has widespread applications in analyzing and processing textual documents [8]. In this section, we review different studies particularly related to text mining.

Ombabi et al. [34] proposed a deep learning model for Arabic language sentiment analysis based on one-layer CNN architecture for local feature extraction and two layers LSTM to maintain long-term dependencies. The feature maps learned by CNN and LSTM are passed to the SVM classifier to generate the final classification. In [33], a model for feature extraction in images called the Joint Weak Saliency mechanism and the Attention Awareness model (JWSAA) is presented in which by weakening the salient features, more complete universal features are obtained. By weakening the high-response features, the model gradually pays attention to all the valuable feature areas, not just the most important features of an image. A multi-purpose subnet is then created through the loss of diversity so that the model pays attention to the different features of the area and makes the features more abundant and diverse.

In [7], authors improved the ATE task with a two-step mixed unsupervised model by combining linguistic patterns with deep learning techniques. They, at the first step, used rules-based methods to extract the single word and multi-word aspects. In the second step, the first step's extracted aspects were used as label data to train the attention-based deep learning model for aspect-term extraction. A recommendation model based on weighted aspect-based opinion mining using the MCNN model is proposed in [11]. The authors used the MCNN model for extracting aspect terms using different channels of the model. The extracted aspects from the user text review were then used to generate aspect ratings using the lexicon method. Ning et al. [31] proposed a real-time 3D face-alignment method that uses an encoder-decoder network with an effective deconvolution layer. The integration of encryption and decryption features adds more features to this network. An effective deconvolution layer in the decoding step applies the L1 norm to select useful features and create multiple features through linear operations. Furthermore, these authors in [32] proposed a model in which valuable and complete features of individuals are extracted by weakening the salient features and obtaining various fine-grained features after removing the interfering information. Second, by locating and intercepting the high response areas of persons, interference from prior information is eliminated and the response of the model to the complete features of individuals is strengthened.

Li et al. [24] proposed a text regression model based on conditional GAN with a generator and a discriminator, with an attempt to associate textual data and social outcomes in a semi-supervised manner. To overcome the problem of the unavailability of the labeled dataset in the classification approach, Saumya & Singh [43], proposed an unsupervised learning model by combining LSTM-autoencoder and EM- clustering algorithm to distinguish spam reviews from other real reviews. Ali et al. [2], proposed a fuzzy ontology and word embedding as a text representation model to improve the task of transportation entities or features extraction and text classification using the Bi-directional Long Short-Term Memory (Bi-LSTM) approach.

Sentiment analysis offers a solution to computationally understand and classify subjective information from user-generated feedback. In [10], the authors proposed a framework based on Persian dependency-based rules that consider the dependency relations between keywords. In addition, they proposed a hybrid framework for concept-level sentiment analysis in the Persian language that integrates linguistic rules and deep learning to optimize polarity detection. Abualigah et al. [1] proposed a feature selection method using a hybrid of particle swarm optimization algorithm with genetic operators to find the more informative features in each document. This method is used to improve the performance of the text clustering and reduce its computational time. The k-means clustering is used to evaluate the effectiveness of the obtained feature subsets. Text-mining fundamentally recognizes biomedical named entities (NER). In [19], authors improved recognition of NER (Named Entity Recognition) based on statistical word embedding by deep learning. Their deep learning method is called Long Short-Term Memory network-Conditional Random Field (LSTM-CRF).

Yousefi-Azar and Hamey [51] investigated the effect of adding random noise to local as the input representation of AE. Then they proposed Ensemble Noisy Auto-Encoder (ENAE) that added noise to the input text and selected the top sentences from an ensemble of noisy runs. In each experiment, a different randomly generated noise was added to the input. Architecture changed the application of the AE from a deterministic feed-forward network to a stochastic runtime model. The authors in [23] introduced a new clustering algorithm based on a genetic algorithm for MEDLINE abstract. They used the genetic algorithm combined with the vector space model to deal with the textual data structure, and cleaned abstracts were the input of the agglomerative clustering algorithm. Lima & de Castro [25] proposed a new system called PERSOMA to predict specific personality traits present in Tweets extracted from social media based on the Big Five Model. They applied a semi-supervised learning method to classify the texts by using a small sample of labeled documents. Their proposed system was trained with the Naïve Bayes classifier, Support Vector Machine, and a Multilayer Perceptron neural network. Table 1. Summarizes some of the researches on text mining problem.

3 Theoretical background

3.1 Text mining

Knowledge Discovery in Databases is “the non-trivial process of identifying valid, novel, useful and ultimately understandable patterns in data” [15]. The data set is divided into three forms: structured, semi-structured and unstructured. The process of extracting useful information from structured data is called data mining. Structured data are completely independent of each other but have the same structure. But textual data is unstructured or semi-structured and must first be structured by methods and then used to extract information and knowledge from

Table 1 An Overview of the Literature on text mining methods

Author (Year) [reference no.]	Data set	Embedding model	Objective	Classification/ Clustering method	Learning method	Evaluation metrics
Ombabi et al. (2020) [34]	Arabic sentiment text	Fast Text words embedding	Arabic sentiment analysis using textual information shared in social networks	Deep learning CNN–LSTM framework	Supervised (Classification)	Precision (0.88), Recall(0.83), F-score(0.86), accuracy(0.86)
Chauhan et al. (2020) [7]	SemEval-16	Fine-tuned word embedding+ POS tagging	A two-step hybrid unsupervised model to overcome the limitations of existing techniques for aspect extraction	Attention-based deep learning model for aspect-term extraction	Unsupervised linguistic rule-based	Precision (0.82), Recall(0.81), F-score(0.81)
Da'u et al. (2019) [11]	Amazon Datasets (MI, Auto, IV, Yelp)	Word2vec based on the CBOW + POS	A weighted **A**spect-based **O**pinion mining using **D**eep learning method for **R**ecommender system (**AODR**)	Deep learning MCNN	Unsupervised (Clustering)	F-score(0.88), RMSE(0.97) and MAE (0.76)
Saumya & Singh (2020) [43]	YouTube	Glove, Word2Vec and OneHot embedding	Spam review detection using LSTM autoencoder	LSTM autoencoder+ EM clustering	Unsupervised (Clustering)	MCC (0.98), precision(0.98), Recall(1.0), F-score(0.99)
Ali et al. (2019) [2]	Social Network	Word2vec	A Transportation Network Monitoring System for Assisting Travel	Fuzzy Ontology and LSTM	Supervised (Classification)	Precision (0.88), Recall(0.86), F-score(0.87), and accuracy(0.84)
Dashtipour et al. (2019) [10]	Persian product and hotel reviews	Fast Text words embedding	Persian Sentiment Analysis	CNN, LSTM, SVM, Logistic Regression	Supervised (Classification)	Precision(0.76), Recall(0.95), F-score(0.84), accuracy(0.81)
Abualigah & Tajudin Khader (2017) [1]	Reuters21578, 20Newsgroups, Dmoz-Business, …	TFIDF	Improve the performance of the text clustering and reduce its computational time with feature selection	PSO+ GO+ K-means	Unsupervised (Clustering)	Precision(0.48), Recall(0.48), F-score(0.49), accuracy(0.52)
Habibi et al. (2017) [19]	Biomedical	PubMed-PMC, Wiki-PubMed--PMC, Patent	recognition of biomedical named entities (NER)	Deep learning LSTM-CRF	Supervised	Precision (0.81), recall(0.81), F-score(0.81)
Yousefi-Azar & Hamey (2017) [51]	BC3 & SKE email	TFIDF/ tf bag-of-words	Text summarization	Ensemble Noisy Auto-Encoder (ENAE)	Unsupervised	Recall (0.68)
Abdessalem Karaa et al. (2016) [23]	MEDLINE abstract	TFIDF	MEDLINE abstract clustering based on a genetic algorithm	Agglomerative algorithm+ genetic algorithm	Unsupervised (Clustering)	Precision (0.98), recall(0.98)
Lima & de Castro (2014) [25]	Obama-McCain Debate (OMD), Sanders, SemEval2013	–	A Personality Prediction System for Social Media Data Analysis	Naïve Bayes classifier, a Support Vector Machine, and a Multilayer Perceptron neural network	Semi-Supervised Classification	Precision(0.87), and accuracy(0.84)

them. Text mining, also called “text analysis”, is the process of converting unstructured text data into meaningful and practical information, or in other words, text analysis is a way to extract knowledge from texts [16]. Text mining allows users to gain knowledge and useful information from unstructured textual data without the need for manual review of vast amounts of information by identifying “topics”, “patterns” and “keywords” [9].

Common text mining tasks include text classification, text clustering, information retrieval etc. Text classification is a supervised technique that assign natural language documents to predefined categories according to their contents [30]. This technique finds counts of words and from that count decides the topic of the document. Text clustering is an unsupervised technique. In text clustering no input- out patterns are predefined. This method groups similar documents into different clusters based upon dividing the similar text into the same cluster [29]. Figure 1 shows an overview of text mining technique.

3.2 Word embedding

Human dictionaries are plain texts, but to use a machine learning model to understand and process natural language, words are converted from plain text to numerical values. One of the simplest conversion methods is the one-hot encoding process, in which each word has a specific position on a vector and the binary values ‘0’ and ‘1’ indicate the absence or presence of a word in a vector, respectively. However, one-hot encoding will impose a lot of computational

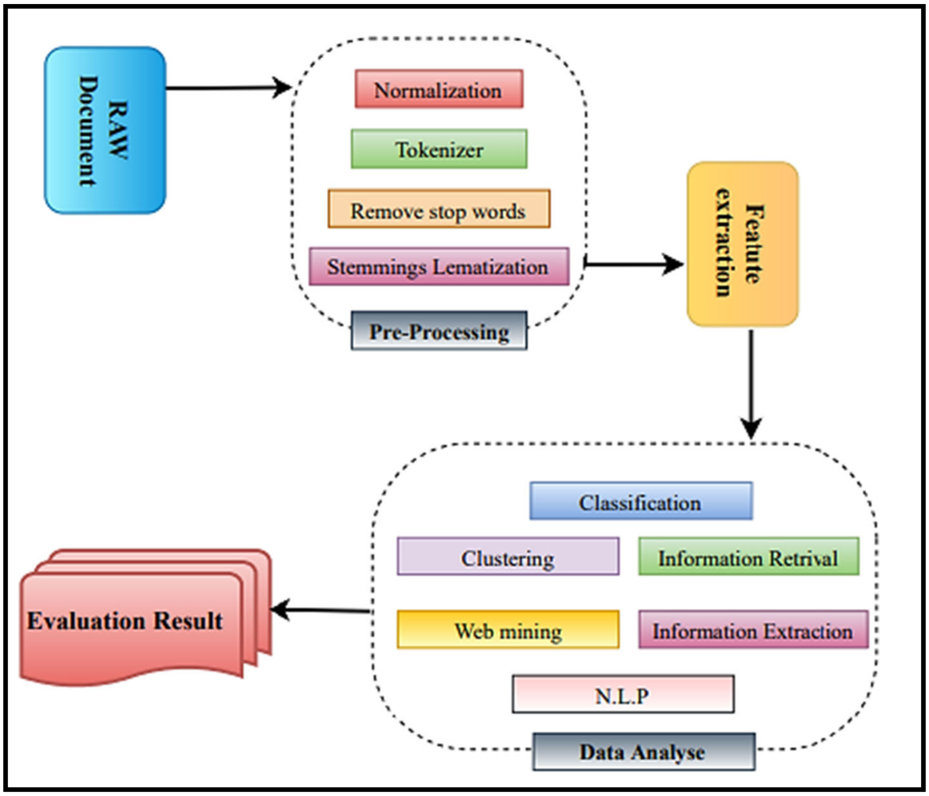


Fig. 1 An overview of text mining technique

overhead when faced with an entire dictionary, consequently, it is an impractical process as the word representation would require hundreds of thousands of dimensions. One of the most popular representations of document vocabulary is word embedding. Word embedding refers to a set of language learning and modeling techniques in Natural Language Processing (NLP) in which words and phrases are mapped from a dictionary to numerical vectors [18]. Conceptually, this technique requires mathematical embedding of words from a large space to a continuous vector space with much smaller dimensions. Word embedding is a dense representation of words in the form of numerical vectors that can be learned by different language models. Bengio et al. [4] refer to word embedding as distributed representations of words in 2003 and train them in a neural language model jointly with the model's parameters. Word embedding representation can reveal many secret relationships between different words. Each set of numbers is considered a valid "word vector" that will be useful when they have acquired the meaning of words, the relationship between them, and the content of different words.

Some of the best popular techniques of word embedding are Word2vec [28], GloVe [38], Fast Text [22], ELMO [39] and BERT. In this paper, BERT word embedding method is used to display Persian words, which is explained below.

3.2.1 BERT word embedding

Pre-trained neural language models (LMs) have achieved significant results in various natural language processing tasks in different languages. A particularly striking example is the performance of BERT. Bidirectional Encoder Representations from Transformers (BERT) as a method of fine-tuning was proposed by Devlin et al. in late 2018 [12]. BERT is a method for (NLP) pre-training which effectively can exploit the deep semantic information of a sentence. In previous word embedding methods, such as GloVe and word2vec, each word is assigned a property vector. Therefore, in word embedding, the same vector is considered for two different words with the same text. But in BERT, a large part of the text is given to the network at the same time, and to produce the vector of each word, the previous and next words are also considered. As a result, different vectors may be generated for the word with different meanings. BERT is a deeply bidirectional, unsupervised language representation with two pre-trained general types: The BERTBASE model, a 12-layer, 768-hidden, 12-heads, 110 M parameter neural network architecture, and the BERTLARGE model, a 24-layer, 1024-hidden, 16-heads, 340 M parameter neural network architecture; models were trained on a version of the with 2500 M words and with 800 M words. The pre-training objective consists of two tasks: employing a Masked Language Model (MLM) to train the model. This model predicts randomly masked tokens by using cross-entropy loss. Implementing the Next Sentence Prediction (NSP) task, in which the model learns to predict whether the second sentence in a pair of sentences is the actual next sentence of the first one or not. BERT can be used to extract high quality language features from textual data or it can be used with model fine-tuning for specific classification tasks, such as question-answering and sentiment analysis, etc. to generate prediction status.

4 Proposed method

This section details the process flow of the proposed method. According to Fig. 2, the proposed method for Barez dataset analysis can be divided into four main phases: Text pre-

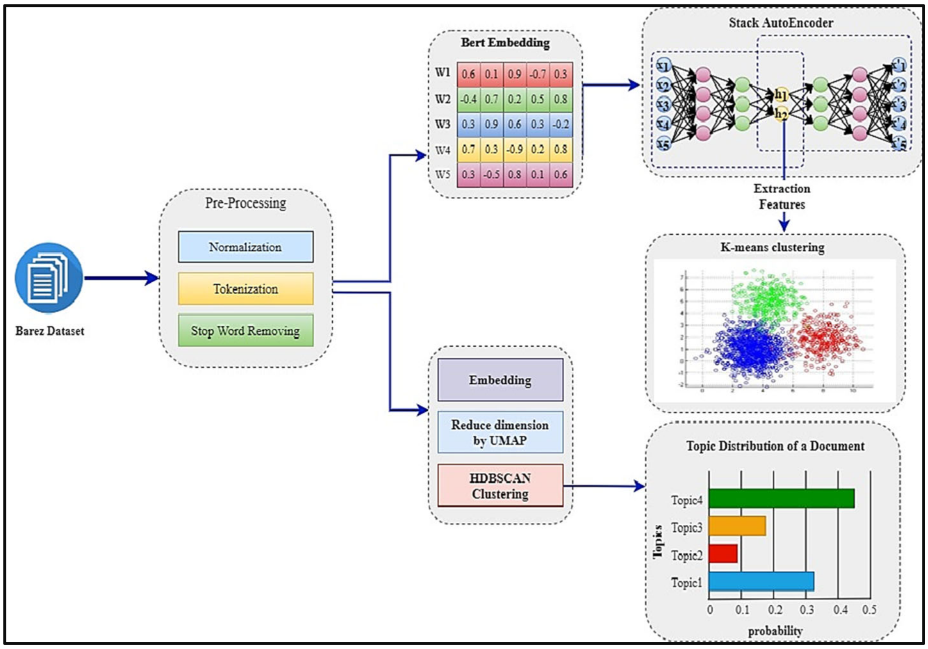


Fig. 2 Architecture of proposed method

processing, Word Embedding, Feature Extraction, and K-means clustering. The raw data used in this paper are suggestions in Persian to increase the quality of Barez's products, which exceeds 27,000 suggestions. Since this textual data is unlabeled, we must classify the data using data clustering techniques so that suggestions that have a similar issue are in the same group. To do this, we must first divide the suggestions into separate sentences. Then use text preprocessing methods to clean up the raw data. Then, to convert Persian words into a language understandable by computer, a new pre-trained language representation model called ParsBERT is used to represent Persian textual data as numerical features. In this step, data features are extracted with numerical values. To reduce the dimension of the data and extract the appropriate features to improve the quality of clustering, stack autoencoder is used. The extracted features are fed to the k-means clustering algorithm, and the more similar suggestions are placed in one cluster so that they differ from the documents in the other clusters. The silhouette criterion is used to evaluate the clustering results. Algorithm 1 shows the pseudo-code of the proposed algorithm.

```
Algorithm 1 : Proposed text clustering

Input: K(number of clusters) , Text documents
Output: K clusters

Method:
  1. Text preprocessing
  2. Apply pre-trained Bert model to convert text documents into Numerical Embedding matrix
  3. Feature extraction by a Stack AutoEncoder
  4. Apply K-means on extracted features to cluster text documents

Return the set of clusters
```

4.1 Text pre-processing

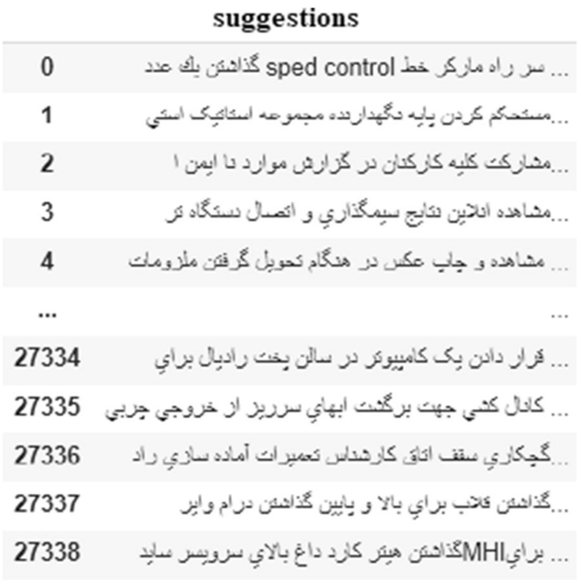
In the first step, we applied standard preprocessing steps before using a text clustering algorithm to convert document contents into the numerical style. Examples of sentences from the Barez raw dataset are shown in Table 2. These Persian sentences are a collection of suggestions presented to the Barez Company in order to increase the quality of products.

First, we remove duplicate sentences and shorter than 2 words in the dataset. Then, we preprocessed the sentences by eliminating suggestions, including numbers and punctuation marks, and URL links. Next, we tokenized each suggestion into single words. Tokenization is breaking down text documents apart into words, terms, symbols, or some other meaningful elements that are called a token. Tokens are used to detect Keywords and Key phrases, calculate each Term frequency in the document and then apply it to further processing [50]. Further, stop words (e.g., “ ب ا“ ,”با“ ,” د“ ,”به“ ,” ا”, etc.) are removed. Stop words are very common words that probably do not carry particular information of their own. Hence, stop words are removed from documents to save computing time and increase the performance of the text clustering technique [44]. All characters are encoded in UTF-8 format.

4.1.1 Word embedding

Text representation is one of the main steps to extract the pattern from the document. Each document is represented as a vector by Vector Space Model (VSM). In this model, each word in the document corresponds to the feature with weight values. A single document generates a large number of features. The size of dimension space and uninformative features affects the performance of the clustering algorithm. Word embedding is a type of word representation for text. Actually, embeddings are dense vector dimensional representations of a point in a higher

Table 2 Barez raw dataset



	suggestions
0	... سر راه مارکر خط sped control گذاشتن یك عدد
1	...مستحکم کردن پایه نگهدارنده مجموعه استاتیک استي
2	...مشارکت کلیه کارکنان در گزارش موارد نا ایمن ا
3	...مشاهده انلاین نتایج سیمگذاري و اتصال دستگاه تر
4	... مشاهده و چاپ عکس در هنگام تحویل گرفتن ملزومات
...	...
27334	... قرار دادن یک کامپیوتر در سالن پخت رادیال براي
27335	... کانال کشي جهت برگشت ابهاي سرریز از خروجي جربي
27336	...گچکاري سقف اتاق کارشداس تعمیرات آماده سازي راد
27337	...گذاشتن قالب براي بالا و پایین گذاشتن درام وایر
27338	... برايMHIگذاشتن هیتر کارد داغ بالاي سرویس ساید

dimensional vector space. Pre-trained word representations are an important component in NLP.

In this step, we utilize the pre-trained ParseBERT as the word embedding algorithm to generate contextualized sentence embeddings. We use this model to extract features, namely word and sentence embedding vectors, from data. ParseBERT includes a multi-layer bidirectional Transformer: 12 hidden layers, 12 attention heads, and 768 hidden sizes. The total number of parameters in this configuration is 110 M. Therefore, the number of features extracted from this step is 768.

4.2 Topic modeling

In this step, we proposed to employ BERTopic modeling for cluster analysis of the dataset. A document treats as a mixture of topics in which each topic is a probability distribution of words. BERTopic contains three stages: 1) Creating documents/sentence embeddings from a set of documents using a pre-trained sentence-transformer to compute dense vector representations for sentences based on BERT embedding. 2) Reducing documents/sentence embeddings dimensions using UMAP and cluster them using HDBSCAN. 3) Extract and reduce topics with c-TF-IDF. The more important words are within a cluster, the more it is representative of that topic. In other words, if we extract the most important words per cluster, we get descriptions of topics.

$$c{-}TF{-}IDF_i = \frac{t_i}{w_i} \times \log \frac{m}{\sum_j^n t_j} \tag{1}$$

Where **t** represents every word in class **i**, **w** is total number of words, total number of classes is **n** and number of documents is **m**.

4.3 Feature extraction with stack AutoEncoder

Feature extraction is one of the most important parts of solving a problem in machine learning. The extracted features should be able to be used for classification and clustering methods. The biggest advantage of autoencoders is the automatic selection of these features. These neural networks are used to reduce the size and reduce processing time and memory costs. Autoencoders play an essential role in unsupervised learning and deep networking.

In this step, AutoEncoder is used to reduce the extracted features' dimensions from the previous step. That is an unsupervised feedforward neural network that employs backpropagation to generate output, which is almost similar to the input. That means Autoencoder can learn to map the input to itself with the goal of dimensionality reduction or manifold learning. Still, the autoencoder concept became more widely used for learning the Generative model of data. Autoencoder consists of two main units: encoder and decoder. Encoder unit maps the input into the codes or features, which is known as the bottleneck layer, and on the other hand decoder unit maps the code to reconstruct the input from the bottleneck layer. The input to the proposed autoencoder in this paper is the embeddings obtained from embedding layers, i.e., ParsBert embedding. Figure 3 shows the basic structure of Autoencoder. As you can see, the network is trained in a way that the weights produced in the layers, make the output with the input as little as possible and equal in the most ideal case possible.

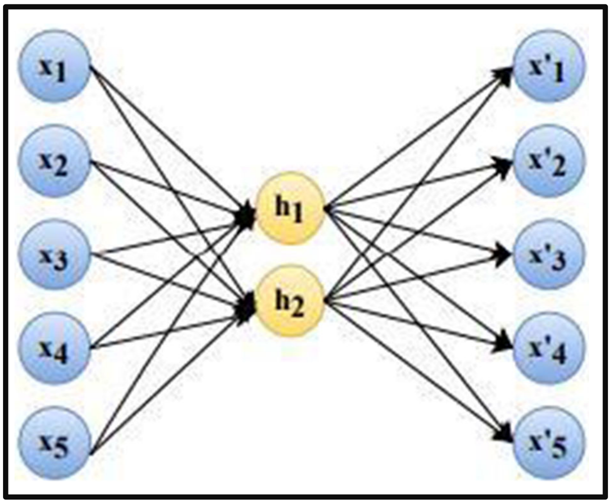


Fig. 3 Basic structure of autoencoder

An autoencoder takes the input X = {x_1, x_2,…, x_m}, and the encoder maps it to a hidden representation h = {h_1,h_2,…, h_k,} calculated from x_n, and x'_n is the decoder vector of the output layer. Therefore, decoder process is:

$$h = \sigma(wX + b) \tag{2}$$

Where σ is encoding function, w is weight matrix of the encoder, and b is the bias unit.

Decoder maps output of the hidden layer to produce output x' of the autoencoder.

$$X^{'} = \sigma^{'}\left(w^{'}h + b^{'}\right) \tag{3}$$

Where σ' is decoding function, w' is weight matrix of the decoder, and b' is the bias unit.

Autoencoders minimize reconstruction errors, often referred to as the loss function defined as follows:

$$L\,(x, x') = \|x{-}x'\|^2 \tag{4}$$

In this paper, we use ReLU activation function because its gradient will not decrease with the independent variables increasing. So the network with ReLU does not suffer from gradient diffusion or vanishing. ReLU function and its structure is shown in Fig. 4.

In this paper, we employ a stacked autoencoder to reduce the dimensions of the embedding matrix. After obtaining the embedding matrix from the previous step, we propose stacked autoencoder architecture to obtain the text features. In this type of autoencoder, a set of layers are stacked. A stacked autoencoder is the deep autoencoder consist several layers of sparse autoencoders where the output of each hidden layer is connected to the input of the successive hidden layer. In the stacked autoencoder, each layer is trained using one layer at a time. An example of a stacked autoencoder is shown in Fig. 5.

We reduced the input vector dimension with each layer during encoding. We started with 256 dimensions at the input layer corresponding 256-dimensional output was expected. The

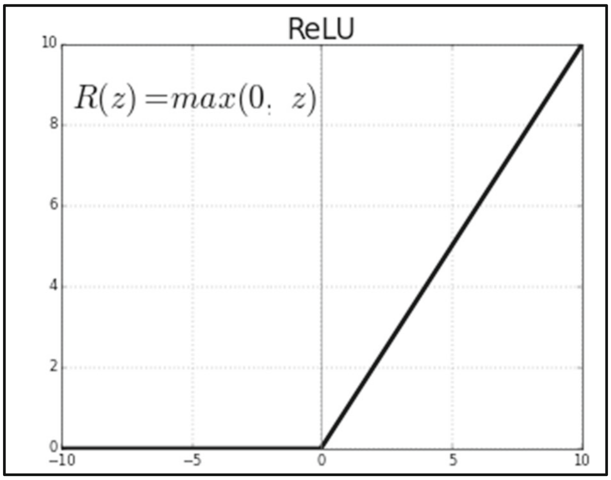


Fig. 4 ReLU activation function

second layer is 128 dimensional. The next layer has 64 dimensions, the 4th and 5th layers have 32 and 16 bits, respectively, and the next layer is reduced to 2 dimensions in the proposed autocoder structure. In the decoding step, the 2D vector was given as input to the next layer, which was extended to a 16D vector. It was further expanded into 32 dimensions and 64 dimensions in 8th and 9th layers respectively. The next layer has 128 dimensions and the final layer has 256 dimensions. The output of the final layer is stored and used to calculate the loss between the input and output. The loss function used in the architecture of proposed autoencoder model is Mean Squared loss. The model was trained for 30 epochs and 256 batch size. The parameters used in each layer is shown in Table 3. In the following clustering step, the output of the encoder is fed to the K-Means algorithm for clustering. The low-dimensional representation vector contains the key information of the given input, and thus yield better clustering results.

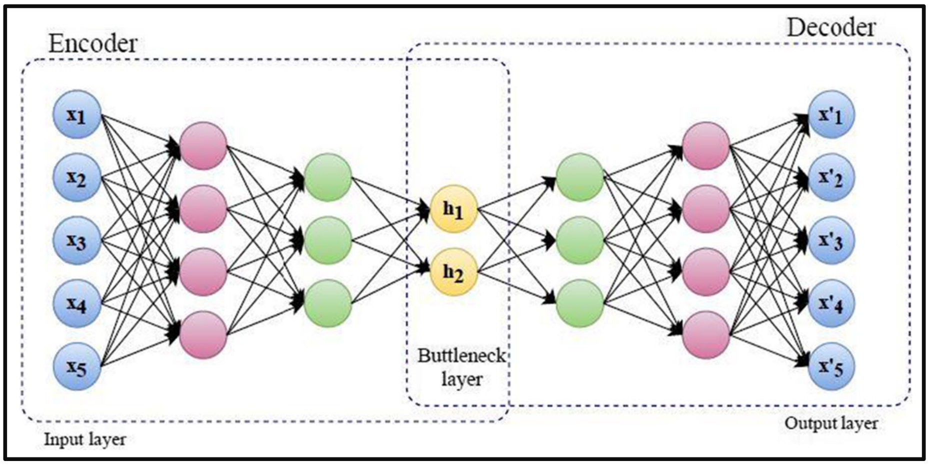


Fig. 5 Structure of stack autoencoder

Table 3 The parameters of the model

Parameter	Value
Optimizer	Adam
Batch size	256
Activation function	Relu
Hidden layer	6
Neurons in layers	256,128,64,32,16,2
Loss function	Mean Squared loss
Epochs	20

4.4 K-means clustering

One of the techniques known in text mining is Text Clustering that groups documents according to their similarities. Its purpose is to place similar documents in one cluster so that they are different from documents in other clusters.

In the last step of the proposed model, we feed Features extracted from the previous step to the k-means clustering algorithm. K-Means is one of the simplest and most popular clustering algorithms. This algorithm is unsupervised because it does not use labeled data. For textual data, this means that no single text belongs to a class or group. K-means is widely used in text clustering [26]. K-means text clustering algorithm divides n text documents $T = \{t_1, t_2, ..., t_n\}$ into a subset of k clusters. This algorithm works iteratively, where initially assigns random initial cluster centroids $Z = (z_1, z_2,. .., z_K)$. Where z_K is cluster centroid of k. Distance between centroids and the points are recalculated in each iteration and centroids move to the center of the points which are closer to them. K-means algorithm is presented in Algorithm 2.

Algorithm 2. K-means clustering algorithm

```
Input: k (the number of clusters), text documents ($T = \{t_1, t_2, ..., t_n\}$)
Output: A set of k clusters

Method:
Randomly choose K document as clusters centroid $Z = (z_1, z_2, . . . , z_K)$.
for all t in T do
    Assign item $t_i$ to the cluster which has the closest centroid using Euclidean distance;
        $C_j = \arg\min_j d(t_i, Z_j)$
    Update cluster centroids for each cluster;
        $Z_j = \frac{1}{|C_j|} \sum_{t_i \epsilon C_j} t_i$
end for
```

5 Experimental results

In this section we present experimental settings on Barez dataset. Experiments done using Python code on the TensorFlow framework with i5 core processor in the development environment: colab with GPU in 64-bit Windows 10 environment on a machine with 4GB RAM.

5.1 Data set

In this section, we present the details of the data used in this paper. To evaluate the performance of the proposed clustering method, we conducted experiments on Barez text data. This data is an experimental dataset consisting of 27,000 suggestions presented to improve the quality of products collected by Barez Company. Each sentence contains between 2 to 15 words. Figure 6 shows a histogram diagram of the distribution of the word counts in sentences. Also, the frequency of words in the document is shown in Fig. 7.

5.2 Evaluation metrics

In this section performance metric used to measure the performance of the proposed method is presented.

Silhouette score is a metric to calculate the separation distance between the resulting clusters. This measure has a range of −1 to 1 [41].

Silhouette coefficients near +1 indicate the neighboring clusters are well far away from each other. A value of 0 means clusters are on or very close to the decision boundary between themselves and − 1 value indicate that those samples are assigned to the wrong cluster. Silhouette score is calculated by the following equation:

$$\text{Silhouette Score} = (\mathrm{b}-\mathrm{a})/\max\left(\mathrm{a},\mathrm{b}\right) \tag{5}$$

Where, a is mean distance between each point within other data points in a cluster and b is mean distance between all clusters.

5.3 Comparative analysis of experimental results

In this section, we evaluate the results obtained through the research experiments of the proposed method in terms of clustering performance and compare the results with the other algorithms in the domain of word embedding and feature extraction. Empirically, different hyper-parameters and settings have been tested. The parameters for the proposed method are fixed, as shown in Table 3. We use the elbow method to pick the number of clusters in k-means. Figure 9 compares the SSE measure to find optimal k. The best k value is 3.

Table 4 shows the topics with the highest frequency of keywords. This table contains the topics with 5 of the top words in them. According to the keywords in each topic, we can specify a label for that topic, which is done by experts in this field. Expert read the keywords generated by the topic modeling and assigns a title to each topic. With BERTopic, we can

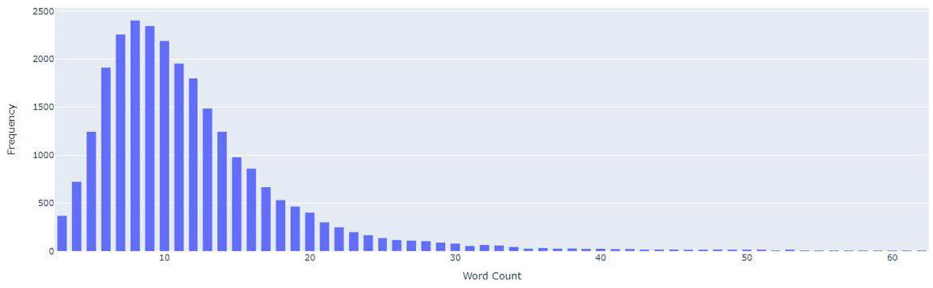


Fig. 6 Distribution of word counts in Barez dataset

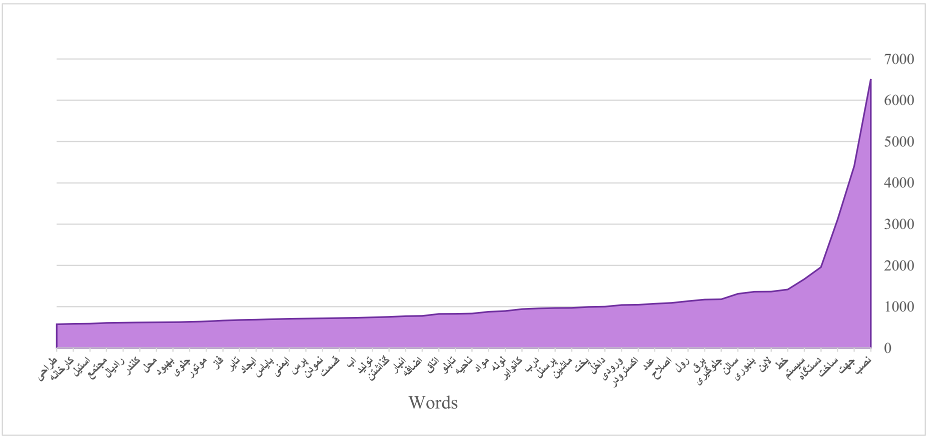


Fig. 7 Frequency of words in Barez dataset

visualize the probability of each document belong to each possible topic. Figure 8 shows an example of the probability distribution of a document in topics.

Evaluation of all tested models is based on silhouette criteria. In fact, this criterion determines the distribution of data in clusters. The higher the amount of silhouette, the higher the clustering quality. To compare the performance of the proposed method with other models, at first three different model for representing textual data are considered. The tf-idf method, which is based on word weighting, as well as two pretrained models are tested to represent Barez Persian text data. Pre-trained models include fast-text and ParsBert. The proposed hybrid method in this paper uses the ParsBert model. The features extracted from each of these three models are given to the k-means clustering algorithm and the clustering results of the Barez text data are displayed based on the models presented in Table 5. In the proposed

Table 4 Top 5 words in topics

Topic	Top 5 words
1	لاین، نصب، دستگاه، ساخت، رول
2	پرسنل، گزارش، فرم، ثبت، ایجاد
3	ایجاد، ثبت، سیستم، عنوان، بارز
4	جهت، نصب، تست، نشانی، آتش
5	سالن، نصب، جلو، اتاق، درب
6	نصب، برق، پرس، جلوگیری
7	تعداد، سایز، ابعاد، ساخت، حمل

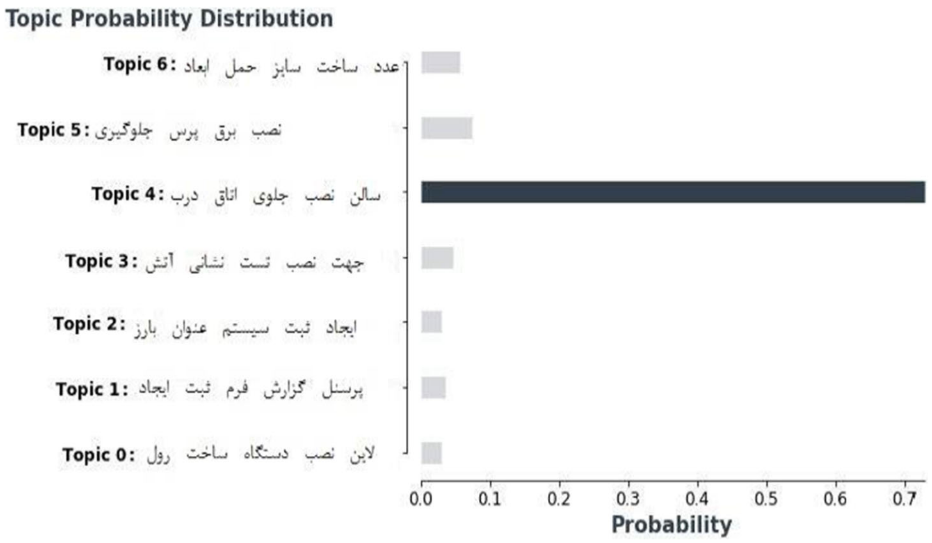


Fig. 8 Topics probability distribution for documents

method, the stacked autoencoder is used to extract effective features in order to reduce the dimensions of text data. The PCA model is also used to compare the efficiency of the proposed method with other feature extraction methods. As shown in Table 5, the amount of silhouette obtained from the Bert + PCA + KM clustering algorithm is much lower than the proposed algorithm. These results prove that the proposed autoencoder model for feature extraction has a much higher efficiency than the PCA model. Figure 10 compares different models according to the silhouette criterion. As can be seen, the proposed model has a higher efficiency than other models. According to the results, Bert + AE + KM and Fasttext + AE + KM use AE to extract features. On the other hand, the Bert + PCA+ KM use PCA to extract features. Bert + AE + KM model achieve superior performance with a Silhouette Score of 0.60, which outperformed Fasttext + AE + KM with 0.49 score and Bert + PCA+ KM with 0.17 score. Also, performance of the proposed model is higher than other algorithms without using AE.

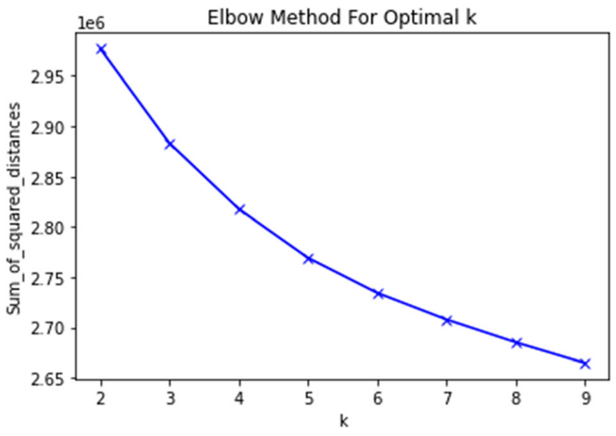


Fig. 9 Find the parameter K for K-means clustering

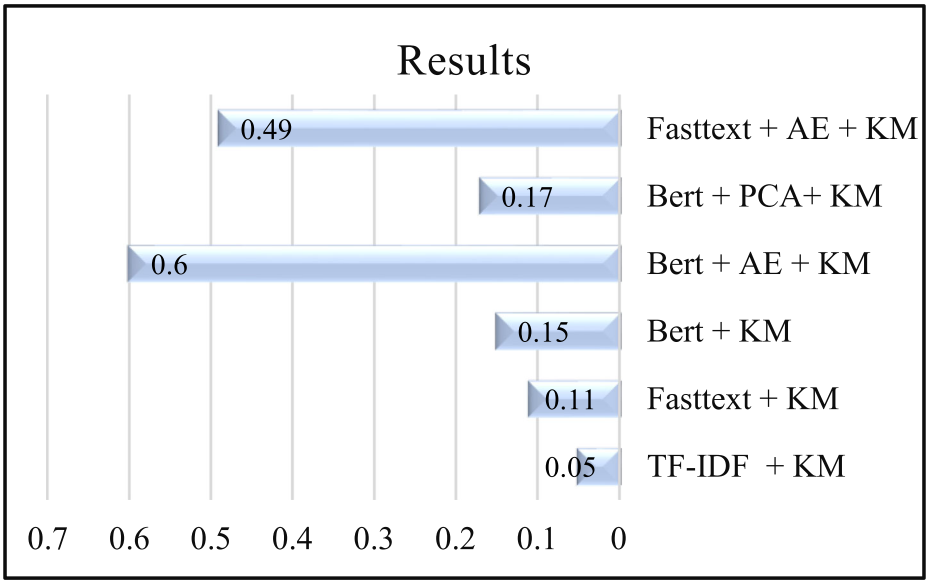


Fig. 10 Comparison of clustering algorithms in terms of Silhouette score

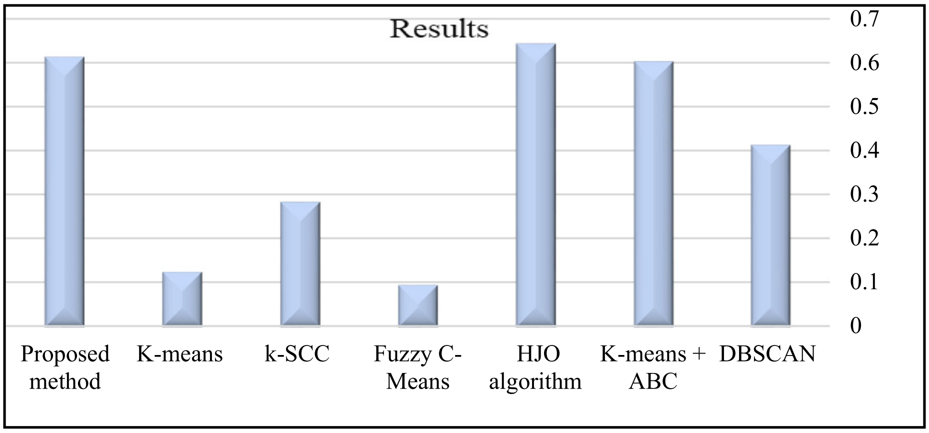


Fig. 11 Comparison of Silhouette Score with state-of-the-art algorithms

Table 6 shows the number of members of the clusters created by the K-means algorithm based on the proposed model. As we can see, textual data are placed in clusters with approximately equal distribution, and increasing the amount of silhouette score in the proposed

Table 5 Clustering results of the k-Means-related methods in terms of Silhouette Score

Data	TF-IDF+KM	Fasttext + KM	Bert + KM	Bert + AE+KM	Bert + PCA+ KM	Fasttext + AE+KM
Barez	0.05	0.11	0.15	**0.60**	0.17	0.49

* KM: K-means

** AE: AutoEncoder

Table 6 Size of clusters

Cluster label	Size
0	12,526
1	9209
2	4395

Table 7 Compare the proposed method with state-of-the-art algorithms in terms of Silhouette score

Clustering method	Silhouette Score
DBSCAN [21]	0.41
K-means + artificial bee colony [35]	0.60
Hybrid Jaya Optimization algorithm (HJO) [45]	0.64
Fuzzy C-Means [20]	0.09
k-SCC [13]	0.28
K-means [40]	0.12
Proposed method	**0.60**

algorithm cannot negatively affect the size of the clusters created. The size of the clusters created is appropriate and the amount of silhouette is desirable.

In addition, the proposed model is compared with a number of studies in the field of text clustering based on the silhouette criterion. The results presented in Table 7 and Fig. 11 show that the efficiency of the proposed model is higher than the models proposed in [13, 20, 21, 40] and is equal to the model proposed in [35]. According to the evaluations performed on different models, we conclude that the use of the autoconductor model proposed in this paper to reduce the extracted features, in addition to reducing the execution time, also increases the quality of clustering.

6 Conclusion and future works

In this paper, we proposed a new deep clustering model based on stacked autoencoder and k-means algorithm to analyze suggestions presented in Barez Company. At first, we used BERTopic modeling for cluster analysis of the dataset. In BERTopic, each document belongs to a topic with a probability distribution. The proposed method was presented in three main steps. In the first step, text preprocessing methods were used to clear the Barez's raw data, and the text data extracted from this step used a new pre-trained language representation model called ParsBERT to convert Persian words into numerical values. The features extracted from this step were fed to a stacked autoencoder with a new architecture. This stacked autoencoder can choose the best features and increase the quality of the clustering algorithm. In the final stage, the k-means clustering algorithm divided the text data into 3 separate clusters. So that similar suggestions were placed in the same group and different suggestions were placed in other clusters. The criterion used to measure the quality of clustering was silhouette score. The results of comparing the proposed method with other models demonstrate that the proposed algorithm clearly outperforms other models. Also, the proposed method has acceptable performance against state-of- the- art algorithms.

In future research, we will use the other clustering algorithms such as hierarchical clustering and GMM to classify unlabeled Barez dataset. We can also employ experts to labeling data and use supervised or semi-supervised algorithms to increase classification performance.

Declarations

Conflict of interest Authors declare that they have no conflict of interest.

References

1. Abualigah LM, Khader AT (2017) Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering. 73(11):4773–4795
2. Ali F, El-Sappagh S, Kwak D (2019) Fuzzy ontology and LSTM-based text mining: A transportation network monitoring system for assisting travel. 19(2):234
3. B. V. Barde and A. M. Bainwad, "An overview of topic modeling methods and tools," in 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), 2017, pp. 745–750: IEEE.
4. Y. Bengio, R. Ducharme, P. Vincent, and C. J Jauvin, "A neural probabilistic language model," vol. 3, no. Feb, pp. 1137–1155, 2003.
5. Bharti KK, Singh PK (2015) Hybrid dimension reduction by integrating feature selection with feature extraction method for text clustering. Expert Syst Appl 42(6):3105–3114
6. Blei D, Carin L, Dunson D (2010) Probabilistic topic models 27(6):55–65
7. Chauhan GS, Meena YK, Gopalani D, Nahta R (2020) A two-step hybrid unsupervised model with attention mechanism for aspect extraction. 161:113673
8. Choudhary AK, Oluikpe P, Harding JA, Carrillo PM (2009) The needs and benefits of Text Mining applications on Post-Project Reviews. 60(9):728–740
9. Choudhary AK, Oluikpe P, Harding JA, Carrillo PM (2009) The needs and benefits of Text Mining applications on Post-Project Reviews. 60(9):728–740
10. Dashtipour K, Gogate M, Li J, Jiang F, Kong B, Hussain AJN (2020) A hybrid Persian sentiment analysis framework: Integrating dependency grammar based rules and deep neural networks. 380:1–10
11. Da'u A, Salim N, Rabiu I, Osman A (2020) Weighted aspect-based opinion mining using deep learning for recommender system. 140:112871
12. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: pre-training of deep bidirectional transformers for language understanding," 2018.
13. D.T. Dinh, T. Fujinami, and V. N. Huynh, "Estimating the optimal number of clusters in categorical data clustering by silhouette coefficient," in International Symposium on Knowledge and Systems Sciences, 2019, pp. 1–17: Springer.
14. M. Farahani, M. Gharachorloo, M. Farahani, and M. Manthouri, "ParsBERT: transformer-based model for Persian language understanding," 2020.
15. Fayyad U, Piatetsky-Shapiro G, Smyth P (1996) From data mining to knowledge discovery in databases. 17(3):37–37
16. R. Feldman, and I. Dagan, "Knowledge Discovery in Textual Databases (KDT)." pp. 112–117.
17. C. Gravelines, "Deep learning via stacked sparse autoencoders for automated voxel-wise brain parcellation based on functional connectivity," 2014.
18. Gupta V, Lehal GS (2009) A survey of text mining techniques and applications. 1(1):60–76
19. Habibi M, Weber L, Neves M, Wiegandt DL, Leser UJB (2017) Deep learning with word embeddings improves biomedical named entity recognition. 33(14):i37–i48
20. Hariri FR (2021) Implementation of fuzzy C-means for clustering the Majelis Ulama Indonesia (MUI) fatwa documents. Jurnal Online Informatika 6(1):79–87
21. R. N. G. Indah et al., "DBSCAN algorithm: twitter text clustering of trend topic pilkada pekanbaru," in Journal of Physics: Conference Series, 2019, vol. 1363, no. 1, p. 012001: IOP Publishing.
22. A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," 2016.
23. W. B. A. Karaa, A. S. Ashour, D. B. Sassi, P. Roy, N. Kausar, and N. Dey, "Medline text mining: an enhancement genetic algorithm based approach for document clustering," Applications of Intelligent Optimization in Biology and Medicine, pp. 267–287: Springer, 2016.
24. T. Li, X. Liu, and S. Su, "Semi-supervised text regression with conditional generative adversarial networks," in 2018 IEEE International Conference on Big Data (Big Data), 2018, pp. 5375–5377: IEEE.

25. Lima ACE, De Castro LN (2014) A multi-label, semi-supervised classification approach applied to personality prediction in social media. 58:122–130
26. J. MacQueen, "Some methods for classification and analysis of multivariate observations," in Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, 1967, vol. 1, no. 14, pp. 281–297: Oakland, CA, USA.
27. T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013.
28. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. 26:3111–3119
29. Nasa D, Engineering S (2012) Text mining techniques-A survey. 2(4):50–54
30. Niharika S, Latha VS, Lavanya D, Technology (2012) A survey on text categorization. 3(1):39–45
31. Ning X, Duan P, Li W, Zhang SJISPL (2020) Real-time 3D face alignment using an encoder-decoder network with an efficient de-convolution layer. IEEE Signal Process Lett 27:1944–1948
32. Ning X, Gong K, Li W, Zhang L, Bai X, Tian S (2020) Feature refinement and filter network for person re-identification. IEEE Trans Circ SystVideo Technol 31:3391–3402
33. Ning X, Gong K, Li W, Zhang LJN (2021) JWSAA: joint weak saliency and attention aware for person re-identification. Neurocomputing 453:801–811
34. Ombabi AH, Ouarda W, Alimi AM, Mining (2020) Deep learning CNN–LSTM framework for Arabic sentiment analysis using textual information shared in social networks. 10(1):1–13
35. K. Orkphol, W. J. Yang, and Applications, "Sentiment analysis on microblogging with K-means clustering and artificial bee colony," International Journal of Computational Intelligence and Applications, vol. 18, no. 03, p. 1950017, 2019.
36. Patel FN, Soni NR (2012) Text mining: A Brief survey. 2(4):243
37. J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.
38. J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.
39. M. E. Peters et al., "Deep contextualized word representations," 2018.
40. Rachman DAC, Goejantoro R, Amijaya FDT (2021) Implementasi Text Mining Pengelompokkan Dokumen Skripsi Menggunakan Metode K-Means Clustering. Jurnal Eksponensial 11(2):167–174
41. P. Rousseeuw and A. Mathematics, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," vol. 20, pp. 53–65, 1987.
42. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science 1985.
43. S. Saumya and J. P. Singh, "Spam review detection using LSTM autoencoder: an unsupervised approach," pp. 1–21, 2020.
44. C. Silva and B. Ribeiro, "The importance of stop word removal on recall values in text categorization," in Proceedings of the International Joint Conference on Neural Networks, 2003, vol. 3, pp. 1661–1666: IEEE.
45. Thirumoorthy K, Muneeswaran KJESWA (2021) A hybrid approach for text document clustering using Jaya optimization algorithm. 178:115040
46. Trier D, Jain AK, Taxt T (1996) Feature extraction methods for character recognition-a survey. 29(4):641–662
47. R. C. Tryon, "Cluster analysis: correlation profile and orthometric (factor) analysis for the isolation of unities in mind and personality. Edwards brother, incorporated," 1939.
48. J. Turian, L. Ratinov, and Y. Bengio, "Word representations: a simple and general method for semi-supervised learning," in Proceedings of the 48th annual meeting of the association for computational linguistics, 2010 pp. 384–394.
49. Vayansky I, Kumar SA (2020) A review of topic modeling methods. 94:101582
50. J. J. Webster and C. Kit, "tokenization as the initial phase in NLP," in COLING 1992 volume 4: the 15th international conference on computational linguistics, 1992.
51. Yousefi-Azar M, Hamey L (2017) Text summarization using unsupervised deep learning. 68:93–105