How to cite this article:

El-Alami, F., El Mahdaouy, A., El Alaoui, S. O., & En-Nahnahi, N. (2020). A deep autoencoder-based representation for Arabic text categorization. *Journal of Information and Communication Technology, 19(3),* 381-398. https://doi.org/10.32890/jict2020.19.3.4

# A DEEP AUTOENCODER-BASED REPRESENTATION FOR ARABIC TEXT CATEGORIZATION

**[1]Fatima-zahra El-Alami, [1]Abdelkader El Mahdaouy,**
**[1,2]Said Ouatik El Alaoui & [1]Noureddine En-Nahnahi**
*[1]Laboratory of Informatics and Modeling, FSDM, Sidi Mohamed Ben Abdellah University, Morocco*
*[2]National School of Applied Sciences, Ibn Tofail University, Morocco*

*fatimazahraelalami1@gmail.com; abdelkader.elmahdaouy@usmba.ac.ma; s_ouatik@yahoo.com; nahnnourd@yahoo.fr*

## ABSTRACT

Arabic text representation is a challenging assignment for several applications such as text categorization and clustering since the Arabic language is known for its variety, richness and complex morphology. Until recently, the Bag-of-Words remains the most common method for Arabic text representation. However, it suffers from several shortcomings such as semantics deficiency and high dimensionality of feature space. Moreover, most existing methods ignore the explicit knowledge contained in semantic vocabularies such as Arabic WordNet. To overcome these shortcomings, we proposed a deep Autoencoder based representation for Arabic text categorization. It consisted of three stages: (1) Extracting from Arabic WordNet the most relevant concepts based on feature selection processes (2) Features learning via an unsupervised algorithm for text representation (3) Categorizing text using deep Autoencoder. Our method allowed for the consideration of document semantics by combining both implicit and explicit semantics and reducing feature space dimensionality. To evaluate our method, we conducted several experiments on the standard Arabic dataset, OSAC. The obtained

**Received:** 2/7/2019 **Revised:** 31/1/2020 **Accepted:** 12/2/2020 **Published:** 11/6/2020

results showed the effectiveness of the proposed method compared to state-of-the-art ones.

**Keywords:** Arabic text representation, deep autoencoder, feature selection, machine learning, text categorization.

## INTRODUCTION

Text categorization consists of automatically assigning textual documents to their most relevant categories (Swesi & Bakar, 2019). This process affords a conceptual view of document collection and plays a key role in numerous tasks of data organization and management. Arabic text categorization suffers from several problems ranging from high dimensionality of feature representation space to the lack of semantics. The Arabic language has a rich morphology and complex orthography due to its inflectional and derivational nature. Moreover, the diacritical marks and absence of capital letters make it a challenging language. Hence, producing a high-quality text representation is a challenging task. Text representation is a fundamental assignment in text mining applications such as text categorization, document clustering, and automatic summarization. Therefore, to enhance Arabic text categorization, it is necessary to build an efficient text representation reducing the feature space dimensionality and reflecting text semantics. Several text categorization techniques have been studied for other languages (e.g. English and French). Yet, research on Arabic text categorization is rather limited. The Bag-of-Words and character-level n-gram approaches have been widely used and still accomplish highly competitive results (Abu-Errub, 2014; Odeh et al., 2015). However, these representations fail to extract similarities between words and phrases leading to feature space sparsity and curse of dimensionality. Moreover, by handling words as independent tokens, semantic dependencies cannot be captured.

One of the real achievements in the advancement of neural networks are deep learning representation models (Bengio et al., 2006; Le & Mikolov, 2014; Mikolov et al., 2013a; Salakhutdinov & Hinton, 2009). These models learn appropriate and consistent input item representation based on unsupervised algorithms. They are defined as learning models that extract text features utilizing several hidden layers. Since these models provide highly discriminative features, semantically similar texts have close vectors in the representation space. Moreover, while these deep architectures allow learning of pertinent implicit text semantics, the explicit semantic incorporated in lexicon resources should be explored to further enhance text representations.

In this paper, we propose an Arabic text categorization method based on deep autoencoder to deal with the aforementioned shortcomings such the high dimensionality of feature representation space and lack of semantics. The main contributions in this work are as follows:

- Combining explicit and implicit semantics using deep learning architecture and Arabic lexicons to enrich text representations.
- Integrating feature selection techniques to highlight more discriminant features within texts.
- Adopting Restricted Boltzmann Machines autoencoder to learn low dimensional text representations and capture hidden semantics.
- Exploring deep autoencoder for Arabic text categorization.
- Conducting several evaluations on the standard Open Source Arabic Corpora (OSAC). To show that our Arabic text categorization method has outperformed state-of-the-art methods.

The remainder of this paper is organized as follows: Section two presents the related work. Section three introduces Stacked Autoencoder based on Restricted Boltzmann Machines. Section four describes the proposed Arabic text categorization method. Section five presents the experimental results. Finally, the last section concludes and outlines the main prospects.

## RELATED WORK

Most existing studies on Arabic text categorization have relied on the Bag-of-Words and/or character-level n-gram representations using mainly stemming techniques to deal with the rich morphology of the Arabic language. Abu-Errub (2014) proposed an Arabic text categorization method which compares a document with pre-defined texts classes using the Term Frequency-Inverse Document Frequency (TF-IDF) measure. Then, documents are assigned to the appropriate subcategory based on Chi-Square measure. Besides, a vector-based evaluation method for Arabic text categorization was introduced by Odeh et al. (2015). This method extracts document keywords and then compares the keywords with each category's keywords. After that, it selects the category having the higher rank of matching keywords. Yousif et al. (2015) proposed a features reduction method to improve Arabic text categorization performance employing support vector machines (SVM) and artificial neural networks classifiers. They used three stemming strategies as feature reduction methods which include: the dictionary-lookup, root-based and light stemming. In another research, Al-Salemi et al. (2019), presented a new benchmark dataset for multi-label Arabic news articles. They assessed various multi-label transformation algorithms including Binary Relevance, Classifier Chains, Calibrated Ranking by Pairwise Comparison and Label Powerset. They utilized three classical classifiers such SVM, k-Nearest Neighbors (kNN) and Random Forest. Furthermore, they employed different adaptation-based algorithms. The results showed the effectiveness of RFBoost and Label Powerset merged with SVM compared to other algorithms.

Besides, numerous models for capturing low dimensional and latent document representations were proposed. Zrigui et al. (2012) introduced a Latent Dirichlet Allocation (LDA) based algorithm to reduce vector space dimensionality and extract document latent topics, and an SVM classifier to assign each vector to its category. Al-Anzi and AbuZeina (2017) presented a method to enhance Arabic text categorization utilizing the cosine distance. They explored the Latent Semantic Indexing (LSI) technique based on Singular Value Decomposition to represent textual information as numerical vectors. They compared several categorization algorithms such as SVM, Naïve Bayes, kNN, Random Forest and Decision Trees.

The major drawbacks of the previous representations ranged from the inference problem inasmuch as they could not handle nonlinear dependencies and the high memory cost in processing time. To overcome these limitations, distributed word and document representations were proposed by Mikolov et al. (2013b). They explored the word embedding model (Word2vec) presented by Mikolov et al. (2013a). This technique considers the average of all word vectors in the text to produce distributed and continuous text representations although it cannot preserve word order. Seeking to resolve this problem, Le and Mikolov (2014) introduced a more sophisticated approach combining word and document embeddings, namely Doc2vec. It relies on an unsupervised algorithm to learn fixed-length vector representations for variable-length texts. The basic idea is to represent each text by a dense vector, which is trained to foresee words in it. In the context of Arabic language, El Mahdaouy et al. (2016) explored both word vectors averaging and document embeddings to enhance Arabic text categorization using a large Arabic text corpus. Similarly, El-Alami and El Alaoui (2018) also investigated the use of document embeddings, word sense and word sense disambiguation to train text representations. These latter are categorized involving a Multi-Layer-Perceptron (MLP) classifier. Alayba et al. (2018a), exploited the strengths of the Word2vec model. They used a 1.5 billion word corpus to train their model which comprises various words from Modern Standard Arabic (MSA) and their dialectical variants. For a sentiment classification task, they explored this model in conjunction with different algorithms such as Multinomial Naive Bayes, Bernoulli Naive Bayes (BNB), Nu-Support Vector Classification, Linear Support Vector Classification, Logistic Regression, Stochastic Gradient Descent, Ridge Classifier and Convolutional Neural Network (CNN). In the same context, Abdullah and Shaikh (2018) employed Word2vec, Doc2vec models and a psycholinguistic feature set to extract tweet vectors. These vectors were fed to deep neural networks such as Dense-Network and LSTM-Network to obtain document label predictions. Al-Smadi et al. (2018) implemented and trained two approaches of deep Recurrent Neural Network (RNN) and SVM. These approaches were evaluated using a reference annotated Arabic hotels' review dataset.

In the same context of deep learning, El-Alami and El Alaoui (2016) studied the use of deep neural networks for Arabic text categorization. They proposed a deep Autoencoder architecture based on Bag-of-Words representations and explored machine learning based classifiers such as SVM, Decision Trees and Naïve Bayes in the categorization phase. The work of Al-Sallab et al. (2017) presented a recursive Autoencoder for opinion mining. Their method was evaluated on three Arabic corpora representing different genres and writing styles. A combined CNN and LSTM model for Arabic sentiment analysis was proposed by Alayba et al. (2018b). This model considered different levels of tweets and comprised different layers. Al-Smadi et al. (2019) compared the usage of deep Recurrent Neural Network model with SVM for aspect-based sentiment analysis of Arabic hotels' reviews. Elnagar et al. (2020) introduced single-label and multi-label datasets. Furthermore, they presented a comparative study of several deep learning models for Arabic text categorization. Nevertheless, most of these deep learning-based research were limited to specific domains such as sentiment analysis and emotion classification using Twitter data.

The literature review shows that the most existing methods for Arabic text categorization utilized classical text representations such as the Bag-of-Words and character-level n-grams. Further methods employed latent semantic representations like the Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation. However, these methods still suffered from the lack of semantics and high dimensionality of their feature space. Thus, in resolving the mentioned drawbacks, we propose an Arabic text categorization method adopting Restricted Boltzmann Machines deep Autoencoder and Bag-of-Concepts to produce a viable representation. In addition, we investigate deep learning neural networks for Arabic text categorization.

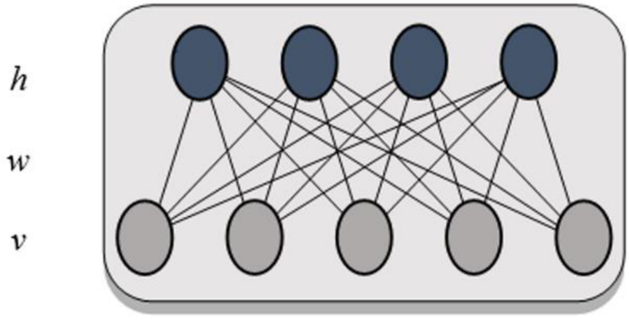

**Figure 1.** Restricted Boltzmann Machine.

## Restricted Boltzmann Machines

A Restricted Boltzmann Machine (RBM) is a network that characterizes a layer of unconnected "visible" input units having undirected and symmetrical associations with another single layer of hidden units. It refers to a graph with no intra-layer connections (Carreira-Perpinan & Hinton, 2005). As shown in Figure 1, this network comprises two layers of hidden $h_j$ and visible $v_i$ units. A weight is assigned to each connection between the units of the two layers and thus a weight matrix $w_{i,j}$ is composed. After that, alternative Gibbs sampling is applied on this matrix to update weights.

The visible unit configuration probability, given a configuration of the hidden units $h$, is $p(v|h)$. The configuration probability of $h$ given $v$ is $p(h|v)$. More formally, given $m$ visible units and $n$ hidden units, the activation probabilities for these units are defined in Equations 1 and 2:

$$p(h_j = 1 | v) = \sigma(b_j + \textstyle\sum_{i=1}^{m} w_{i,j} v_i) \tag{1}$$

$$p(v_i = 1 | h) = \sigma(a_i + \textstyle\sum_{j=1}^{n} w_{i,j} h_j) \tag{2}$$

Where,
$\sigma$ denotes the logistic sigmoid.
$v_i$ and $h_j$ are the binary states of the visible unit $i$ and hidden unit $j$. $a_i$ and $b_j$ are their corresponding biases.

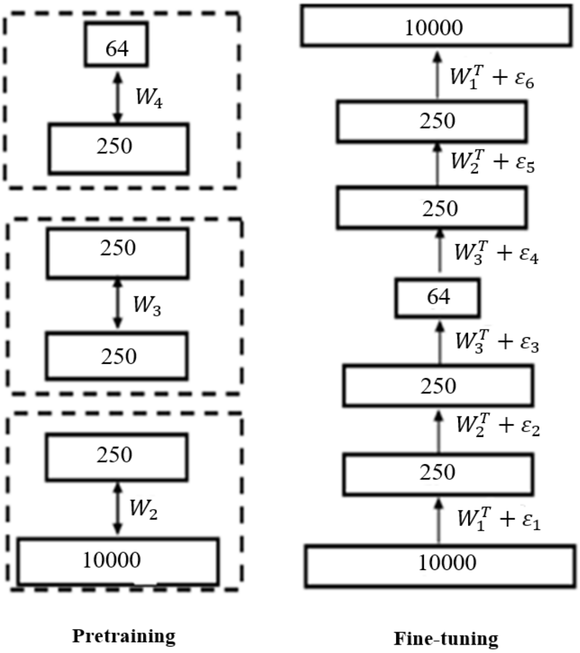

**Figure 2.** Left panel: Pretraining by learning a stack of RBMs. Right panel: Fine-tuning (Salakhutdinov & Hinton, 2009).

## RBMs Autoencoder Architecture

An Autoencoder is a feedforward neural network containing one or multiple hidden layers. The main purpose is reconstructing the input data in an unsupervised manner (Tan & Eswaran, 2008). The Autoencoder consists of an encoder which reverses the input information into low-dimensional representations and a decoder that reconstructs the original information from the encoder output. It is essential to alleviate the distance between the Autoencoder input and output utilizing backpropagation. However, this latter is not appropriate for deep neural systems seeing that the training will be moderate. Hence, RBMs pretraining for each hidden layer by Hinton and Salakhutdinov (2006) to accelerate the training procedure through parameters initialization was employed. It begins with a traditional one-hidden layer then creates a pile of RBMs, where outputs from the concealed layer are the contributions for preparing the following RBM layer. This greedy layer-by-layer training can be repeated a few times to pretrain a deep model.

After the pretraining, the individual RBMs at each level is unrolled to have a symmetric deep Autoencoder topology. Then, as shown in Figure 2, fine-tuning is used to diminish the input vectors reconstruction error through the whole system utilizing backpropagation. This backpropagation can be performed via square error function or cross entropy function (Tian et al., 2010). In this work, we utilized the square error which is characterized as follows in Equation 3:

Where,

$j$ is a variable

$$Square\ Error = \sum_j \left( \hat{x}_j - x_j \right)^2 \qquad (3)$$

$\hat{x}_j$ is the vector of observed values being predicted.

$x_j$ is the vector of the predicted values.

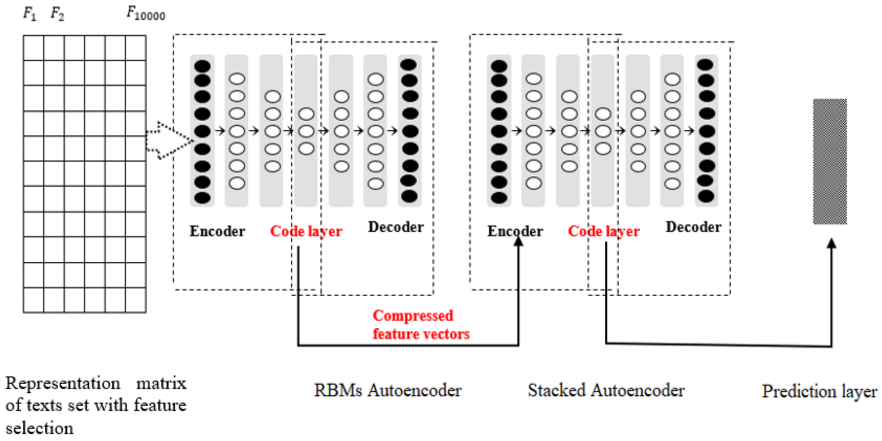

**Figure 3.** The proposed Arabic text categorization system architecture.

## ROPOSED METHOD FOR ARABIC TEXT CATEGORIZATION

As illustrated in Figure 3, our method consisted of various modules. First, we incorporated the knowledge contained in Arabic WordNet using Bag-of-Concepts representation to enrich the vocabulary and feature selection techniques to enhance text vector quality. Then, we used a deep RBMs Autoencoder to learn higher-level features representation. After that, the resulting representations were fed to a deep Autoencoder classifier.

### Preprocessing and Feature Selection Module

The Arabic text preprocessing consisted of three steps:

- Tokenization;
- Removing stop words, punctuation marks, numbers and words written in different languages;

•Normalizing words by replacing letters (“آ إ أ “) with (“ا”), letters (“ؤ ء”) with (“ا”) and (“ا”) with (“ى”).

After text preprocessing and as illustrated in Figure 4, we mapped Bag-of-Words to Bag-of-Concepts using Arabic WordNet 2.0.1 to capture the semantics and enrich the Arabic text representations. The mapping strategy added concepts to terms using a threshold to select only words having important frequencies. After that, we performed Chi-Square (khi2) and Variance Threshold techniques in order to rank the best features by eliminating non-discriminant terms. The Chi-Square ( $\chi^2$ ) is a technique for measuring the association level between a word and a category (Yang & Pedersen, 1997). The main hypothesis is that a word whose recurrence emphatically relies upon the category where it happens will be increasingly valuable for discriminating it. Chi-Square measures the independence degree between a term $t$ and category $c$. Depending on the two-way contingency table of a $t$ and $c$, where $A$ is the number of times $t$ and $c$ co-occur, $B$ is the number of times the $t$ occurs without $c$, $C$ is the number of times $c$ happens without $t$, $D$ is the number of times neither $c$ nor $t$ happens and $N$ is the total number of texts in the corpus. The word-goodness measure is calculated using Equation 4 as follows:

$$\chi^2(t,c) = \frac{N*(A*D-C*B)^2}{(A+C)*(B+D)*(A+B)*(C+D)} \tag{4}$$

$\chi^2$ statistic equals zero if $t$ and $c$ are autonomous.

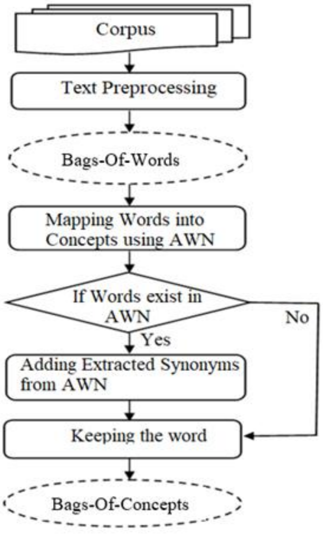

**Figure 4.** Building feature representation space for texts.

In this work, the number of features was fixed empirically at 10000 since it gave the best results.

Variance Threshold (VT) is a basic standard technique to select features by expelling all features whose variance does not attain some threshold (Guyon & Elisseeff, 2003). It discards each of the zero-variance features, for instance, features that have similar value in all texts. In this work, we fixed the threshold at 0.001.

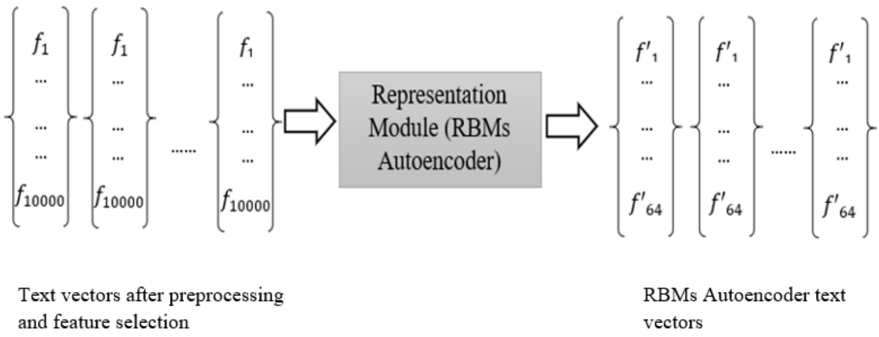
$f_1$
...
$f_{10000}$
......
Representation Module (RBMs Autoencoder)
$f'_1$
$f'_{64}$
Text vectors after preprocessing and feature selection
RBMs Autoencoder text vectors

**Figure 5.** Input and output vectors of proposed representation module

## Representation Module

As shown in Figure 5, after selecting the best features using Chi-Square or Variance Threshold, the obtained vectors are fed to the Autoencoder module. To train the deep Autoencoder, we used RBMs pre-training as mentioned in the previous section. A pile of RBMs was generated during this step. Then, each RBM was unfolded to obtain a symmetric architecture in order to produce the resulting input vectors from the previous module. After that, weights were fine-tuned using square error function to reduce reconstruction error between the input and output data. We constructed a deep Autoencoder with seven (7) layers. We evaluated different architectures and then adopted 10000-250-250-64-250-250-10000 architecture (see Figure 2) based on its good results. After freezing the decoder layer, we obtained low-dimensional 64-length vectors for each textual document. To accelerate the pretraining process, we subdivided the training set into small mini batches. For the fine-tuning, we employed the conjugate gradient method.

Table 1

*OSAC Corpus.*

| Category | Document number |
|---|---|
| Economy | 3102 |
| History | 3233 |
| Education and family | 3608 |
| Religion and Fatwa | 3171 |
| Sports | 2419 |
| Health | 2296 |
| Astronomy | 557 |
| Law | 944 |
| Stories | 726 |
| Food Recipes | 2372 |

### Categorization Module

After learning text representations based on the deep RBMs Autoencoder, the latter are fed to another deep Autoencoder in order to be categorized. We adopted the deep Autoencoder classifier owing to its ability to reconstruct inputs without the need for labels and its effectiveness in classifying applications. We trained our Autoencoder using a stack of layers to reconstruct the inputs. After the training, we dropped the decoder layer and connected a *Softmax* layer to the encoder features layer to map vector representations to a suitable class among ten categories. After several experiments, we used a stack of layers (three layers for the encoder and three layers for the decoder: 64-700-64-64-64-700-64). We utilized *Relu* activation function and *Adam* optimizer in the training phase.

## EXPERIMENTAL RESULTS

The corpus description and text categorization performance indicators are presented in this section including description of the experiments and analysis of the results.

## The Corpus

We performed our experiments on the standard OSAC corpus, which is the largest free public Arabic text corpus. The corpus contains 22.428 text documents; each document belongs to one of 10 categories. There are 18 million words and more than 449.600 distinct words after removal of stop words. The dataset statistics are described in Table 1.

## Performance Measures

To measure the effectiveness of our Arabic text categorization system, we utilized three indicators including precision, recall and F-measure. For a given category $y$, the precision ($p$) and the recall ($r$) are defined in Equations 5 and 6:

$$p = \frac{TP}{TP + FP} \tag{5}$$

$$r = \frac{TP}{TP + FN} \tag{6}$$

$TP$ : the set of texts affected correctly to $y$ .
$FP$ : the set of texts incorrectly affected to $y$ .
$FN$ : the set of texts not affected incorrectly to $y$ .
F-measure represents the harmonic mean of $p$ and $r$, it is calculated using Equation 7:

$$F - measure = \frac{2p.r}{p + r} \tag{6}$$

We evaluated our method against the leading text representation models since they are well-known and the most used in Arabic. Moreover, they enabled production of fixed-length vectors for text categorization tasks. These representations included Doc2vec, Latent Semantic Analysis (LSA) and Bag-of-Words using TF-IDF, and they were integrated with the SVM classifier. The illustrated results in Table 2 demonstrate that the proposed method improves the performance of Arabic text categorization. Our model adopted the Bag-of-Concepts representation and the Chi-Square outperformed all baselines and achieved up to 93% and 94% in terms of F-measure and precision, respectively. This was not surprising since the classical method based on Bag-of-Words representation, afforded only 88% of F-measure, by ignoring the semantics and word contexts. Moreover, the proposed method captured exact inference and handled complex semantic tasks since it considered nonlinear dependencies in comparison with the Latent Semantic representation which could not. The

Latent Semantic representation achieved only 89% in terms of F-measure. Apart from this, a significant reduction of feature space dimensionality was derived from our technique (64 fixed-length vectors) compared with other methods, 300 dimensions for Doc2vec and LSA. Overall, the obtained results showed that the proposed technique enhanced Arabic text categorization by reducing the dimensions of the feature representation space and represented each textual document by a vector containing only 64 features. In addition, our proposed method reflected the hidden document structures which was attributed to the Autoencoder architecture.

Table 2

*Categorization Results Using Different Text Representations.*

| Text representation | F-measure | Precision | Recall |
|---|---|---|---|
| **Proposed method** | 0.93 | 0.94 | 0.91 |
| **Classical Bag-of-Word method** | 0.88 | 0.91 | 0.88 |
| **Doc2vec method** | 0.89 | 0.91 | 0.88 |
| **Latent Semantic Analysis method** | 0.89 | 0.91 | 0.89 |

To study the feature selection impact on the effectiveness of our system, we utilized two different feature selection methods such the Chi-Square and VT. The results are shown in Figure 6. From this figure, it can be seen that the Chi-square was more effective than the VT for both Bag-of-Words and Bag-of-Concepts representations since it selected the best features based on the probability of interdependence between the term and category. Chi-Square using Bag-of-Concepts performed 94%, 91% and 93% for precision, recall and F-measure, respectively. In contrast, the VT using Bag-of-Concepts achieved only 92%, 89% and 90% for precision, recall and F-measure, respectively. In respect of the Bag-of-Words representation, the Chi-Square reached less than 91% for all measures in comparison with the VT which achieved 86% and 83% for precision and F-measure, respectively.

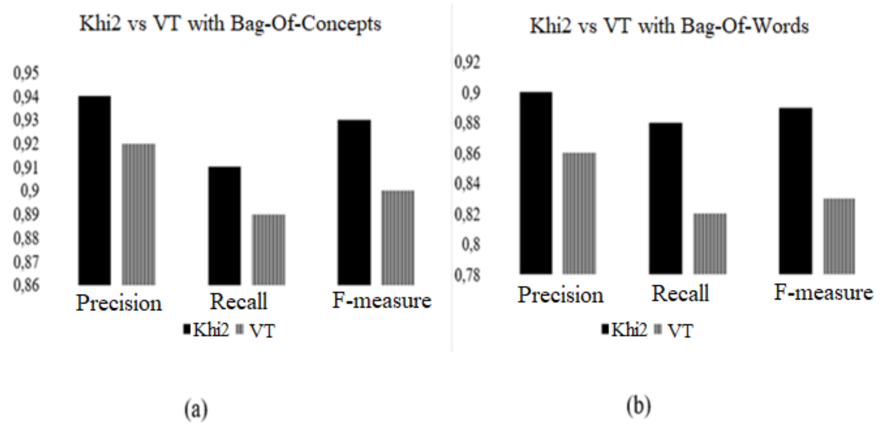

**Figure 6.** Performance analysis of khi2 vs VT. (a) Using the Bag-of Concepts representation. (b) Using the Bag-of-Words representation.

In order to show the Bag-of-Concepts representation effect on system performance, we conducted different experiments using Bag-of-Concepts and Bag-of-Words. The results are shown in Figure 7. It can be noted that our model using the Bag-of-Concepts representation surpassed the model using Bag-of-Words representation regardless of the used feature selection method. This was due to the mapping of words to their corresponding concepts.

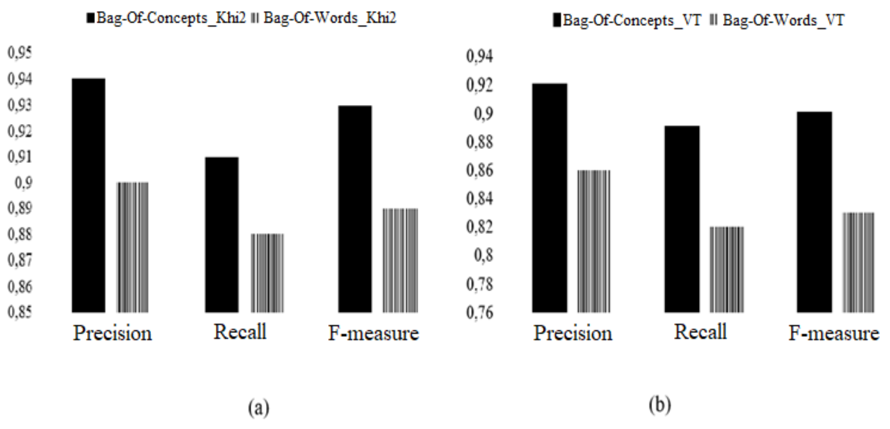

**Figure 7.** Performance analysis of Bag-Of-Words representation vs the Bag-Of-Concepts representation. (a) Performance based on khi2. (b) Performance using VT.

To evaluate the impact of adopting the deep Autoencoder classifier, we compared its performance with the Multi-Layer Perceptron and SVM by applying the khi2 measure and the Bag-of-Concepts representation. The results of this comparison are illustrated in Table 3. We observed that the deep Autoencoder slightly outperformed the other classifiers and achieved the best performance specifically, the best F-measure by a score of 93%.

Table 3

*Comparison between Autoencoder, MLP and SVM classifiers using Bag-of-Concepts and Khi2.*

| | **Deep Autoencoder** | **MLP** | **SVM** |
|---|---|---|---|
| **Precision** | 0.94 | 0.93 | 0.93 |
| **Recall** | 0.91 | 0.90 | 0.91 |
| **F-measure** | 0.93 | 0.91 | 0.92 |

## CONCLUSION AND FUTURE WORK

We proposed an Arabic text categorization method based on Bag-of-Concepts and deep Autoencoder representations. It incorporates explicit semantics relying on Arabic WordNet and exploits Chi-Square measures to select the most informative features. In sum, successive stacks of RBMs were applied to text vectors to produce high-level representations. The learned features were fed to another deep Autoencoder for categorization. An exhaustive set of experiments was carried out and has shown that using the Autoencoder as text representation model combined with Chi-Square and classifier outperformed state-of-the-art techniques and achieved the best results by 94% and 93% for precision and F-measure, respectively. The principal advantages are: (1) Integrating explicit semantics in order to improve the quality of text vectors; (2) Modeling semantic structure within texts; (3) Reducing representation dimensionality and (4) Exploring deep learning networks for Arabic text categorization. In our future work, we aim to tackle the issue of Arabic language ambiguity and enhance the performance of our system utilizing sense embedding techniques.

## ACKNOWLEDGMENT

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

## REFERENCES

Abdullah, M., & Shaikh, S. (2018, June). Teamuncc at semeval-2018 task 1: Emotion detection in English and Arabic tweets using deep learning. In *Proceedings of the 12th International Workshop on Semantic Evaluation* (pp. 350-357).

Abu-Errub, A. (2014). Arabic Text Classification Algorithm using TFIDF and Chi Square Measurements. *International Journal of Computer Applications*, *93*(6).

Alayba, A. M., Palade, V., England, M., & Iqbal, R. (2018a, March). Improving sentiment analysis in Arabic using word representation. In *2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR)* (pp. 13-18). IEEE.

Alayba, A. M., Palade, V., England, M., & Iqbal, R. (2018b, August). A combined CNN and LSTM model for Arabic sentiment analysis. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction* (pp. 179-191). Springer, Cham.

Al-Anzi, F. S., & AbuZeina, D. (2017). Toward an enhanced Arabic text classification using cosine similarity and Latent Semantic Indexing. *Journal of King Saud University - Computer and Information Sciences*, *29*(2), 189-195.

Al-Salemi, B., Ayob, M., Kendall, G., & Noah, S. A. M. (2019). Multi-label Arabic text categorization: A benchmark and baseline comparison of multi-label learning algorithms. *Information Processing & Management*, *56*(1), 212-227.

Al-Sallab, A., Baly, R., Hajj, H., Shaban, K. B., El-Hajj, W., & Badaro, G. (2017). Aroma: A recursive deep learning model for opinion mining in Arabic as a low resource language. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, *16*(4), 1-20.

Al-Smadi, M., Qawasmeh, O., Al-Ayyoub, M., Jararweh, Y., & Gupta, B. (2018). Deep Recurrent neural network vs. support vector machine for aspect-based sentiment analysis of Arabic hotels' reviews. *Journal of Computational Science*, *27*, 386-393.

Al-Smadi, M., Talafha, B., Al-Ayyoub, M., & Jararweh, Y. (2019). Using long short-term memory deep neural networks for aspect-based sentiment analysis of Arabic reviews. *International Journal of Machine Learning and Cybernetics*, *10*(8), 2163-2175.

Bengio, Y., Schwenk, H., Senécal, J.-S., Morin, F., & Gauvain, J.-L. (2006). Neural Probabilistic Language Models. *Innovations in Machine Learning*, 137-186.

Carreira-Perpinan, M. A., & Hinton, G. E. (2005). On Contrastive Divergence Learning. In *Aistats* (Vol. 10, pp. 33-40).

El Mahdaouy, A., Gaussier, E., & El Alaoui, S. O. (2016, October). Arabic text classification based on word and document embeddings. In *International Conference on Advanced Intelligent Systems and Informatics* (pp. 32-41). Springer, Cham.

Elnagar, A., Al-Debsi, R., & Einea, O. (2020). Arabic text classification using deep learning models. *Information Processing & Management*, *57*(1), 102-121.

El-Alami, F. Z., & El Alaoui, S. O. (2016, December). An Efficient Method based on Deep Learning Approach for Arabic Text Categorization. In *International Arab Conference on Information Technology*.

El-Alami, F. Z., & El Alaoui, S. O. (2018, November). Word sense representation based-method for Arabic text categorization. In *2018 9th International Symposium on Signal, Image, Video and Communications (ISIVC)* (pp. 141-146). IEEE.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, *3*(Mar), 1157-1182.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the Dimensionality of Data with Neural Networks. *Science, 313*(5786)*,* 504-507.

Le, Q., & Mikolov, T. (2014, January). Distributed representations of sentences and documents. In *International Conference on Machine Learning* (pp. 1188-1196).

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

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems* (pp. 3111-3119).

Odeh, A., Abu-Errub, A., Shambour, Q., & Turab, N. (2015). Arabic Text Categorization Algorithm Using Vector Evaluation Method. *International Journal of Computer Science and Information Technology*, *6*(6), 83-92.

Salakhutdinov, R., & Hinton, G. (2009). Semantic hashing. *International Journal of Approximate Reasoning*, *50*(7), 969-978.

Swesi, I. M. A. O. & Bakar, A. B. (2019). Feature clustering for pso-based feature construction on high-dimensional data. *Journal of Information and Communication Technology*, *18*(4), 439-472.

Tan, C. C., & Eswaran, C. (2008, May). Performance comparison of three types of autoencoder neural networks. In *2008 Second Asia International Conference on Modelling & Simulation (AMS)* (pp. 213-218). IEEE.

Tian, X., Hérault, R., Gasso, G., & Canu, S. (2010, January). Pré-apprentissage supervisé pour les réseaux profonds. In *Proceedings of Rfia* (Vol. 2010, p. 36).

Yang, Y., & Pedersen, J. O. (1997, July). A comparative study on feature selection in text categorization. *International Conference on Machine Learning* (Vol. 97, No. 412-420, p. 35).

Yousif, S. A., Samawi, V. W., Elkabani, I., & Zantout, R. (2015). The Effect of Combining Different Semantic Relations on Arabic Text Classification. *World of Computer Science & Information Technology Journal*, *5*(1), 12-118.

Zrigui, M., Ayadi, R., Mars, M., & Maraoui, M. (2012). Arabic Text Classification Framework Based on Latent Dirichlet Allocation. *Journal of Computing and Information Technology*, *20*(2), 125-140.