An Efficient Method based on Deep Learning Approach for Arabic Text Categorization

Fatima-Zahra El-Alami[1], Said Ouatik El Alaoui[1]
[1]Laboratoire Informatique et Modélisation, FSDM, USMBA, Morocco

Abstract:*In this paper we propose an efficient methodbased on deep learning for Arabic Text Categorization (ATC) thatis considered as a challenging task.We explore deep learning approach to enhance text representation, we use deep stacked autoencoder as deep architecture to produce high level abstraction presentation. We take advantage of short reproducedcodes which enable us to take into account implicit semantic, in addition this representation reduces the dimensionality of representation space.We have conducted several experiment results on CNN Arabic news dataset, two types of stemmers are used: Light and Khoja stemmer. Tree machine learning techniques are explored: the decision tree(DT), the Naïve Bayesian (NB) and Support Vector Machine (SVM).The obtained results illustrate that the proposed methodaffords good performance in Arabic Text Categorization System.*

Keywords:*Arabic text categorization, deep learning, deep autoencoder, RBM, implicit semantic*.

1. Introduction

The large volume of data which are available on internet makes the task of information retrieval difficult. So as a solution the available data must be organized systematically in order to be explored, there are various applications to perform this as documents categorization.Text categorization is the process during which a set of documents are assigned labels; the set of labels are assigned in advance [4].Many algorithms have been proposed and implemented to solve the problem of English text categorization,however few studies have been carried out for ATC, because it is a very complex inflectional language which makes ordinary analysis a very complex task. These studies focused on classic representations such as: TFIDF, n-grams. In this work we investigate a presentation that captures implicit semantic, and it can be appropriate for large corpus.

Deep learning refers to learning models which use feature hierarchies with many layers. Thus, a deep artificial neural network is a multi-layer network composed of input and output layers, in addition to numerous hidden layers between the input and outputthose hidden layers are composed of hidden (or "latent") units that can be used to describe underlying features of the data[7].

Dealing with highly complex semantic tasks requires deep learning architectures which enables us tolearn semantic structures for language inference in such way to have similar representations for semantically similar phrases that capture underlying semantics.Explicit semantic in which we uselexical resources or ontologies can'tenable us to automatically learning semantic structures for language inference.

In this work, we present a novel method for ATC based on deep RBM autoencoder which is considered as the most efficient deep neural network [14]. The idea is to use deep autoencoder to produce short codes for documents representation in such way that the documents which are near in code space will be in the same category. Using this method will enable as to have hidden semantic and reduce the dimensionality of representation space [12].

The rest of this paper is organized as follows: Section 2 presents the related work. In section 3, RBM are presented. Section 4 describes Stacked Autoencoder with RBM pre-training.Section 5describes the proposed method forATC. Section 6presentsresults. Finally, the last section draws the conclusion andoutlines future work.

2. Related Work

In [1], authorsproposed a new method for Arabic text classification in which a document is compared with

pre-defined documents categories based on its content using the TF.IDF method measure, then the document is classified into the appropriate sub-category using Chi Square measure.

In [2], authors investigated the use of Bayesian learning for ATC, Two representation types were used for representing the text: Bag-Of-Word and character-level n-gram,and four feature selection methods were investigated to reduce the feature space dimensionality: Mutual Information, Chi-Square statistic, Odds Ratio and GSS-coefficient. Three classifiers based on Bayesian theorem had been implemented which are Simple Naïve Bayes (NB), Multi-variant Bernoulli Naïve Bayes (MBNB) and Multinomial Naïve Bayes (MNB) models.

In [11], it is introduced a new ATC method using vector evaluation method. The proposed method determines the key words of the document by weighting each of its words, and then comparing these key words with the key words of each category and select the category that had high match percentage.

Authors in [5] have used conceptual representation of the text basing on Arabic WordNet (AWN) as a lexical and semantic resource, and different similarity measures, to comprehend the effect of using WordNet (WN) authors incorporated it in a comparative study with different modes of representation which are bag of words and n-grams.

In [16], authorsproposed a feature reduction method that aims to enhance the effectiveness of the Arabic text classification through artificial neural network and support vector machines. Three stemming strategies were used: the dictionary-lookup stemming, the root-based stemming and lightstemming methods, they are considered as feature reduction methods. In addition the conceptual representation using WN concepts was included.

To enhance ATC authorsin[8] proposed two methods of Arabic word disambiguation to use the most appropriate concept after the integration of semantic, in the first one, they proposed to use both two external resources AWN and WN based on term to term Machine Translation System (MTS). The second one consists of choosing the nearest concept for the ambiguous terms based on more relationships with different concepts in the same local context. As Feature methods they used Chi-Square statistics for feature selection.

In [6], authors proposed a new document representation in Text Mining based on signal representation and spectral processing by Wavelets Transform, they were interested in semantic dependence between descriptors of texts. The signal of descriptor is a sequence of values which indicate the occurrence of a specific descriptor in a particular section of a document. They compared their proposed method to Vector Space Model representation and Latent Semantic.

In [17], authors presented an algorithm based on the Latent Dirichlet Allocation (LDA) to reduce feature set and the Support Vector Machine was used to perform the classification of Arabic texts. The proposed model in this work adopts a “topics” sampled by LDA model as text features, then each text is represented on the vectors of these topics.

Several methods for ATCexplored classic representation methods asTF.IDF and n-gram,some works integrated explicit semantic basing on WN or AWN to solve the problem of semantic,and some othersused hidden semantic methods as LDA method or Wavelets Spectral Analysis. To reduce the dimensionality of representation space some works used feature selection techniques.Althoughwe need to learn the semantic structuresautomatically, for this purpose we propose amethod for ATC that use deep learning approach to produce high level representation, this representation involves the implicit semantic and giveslow-level features.

3. Restricted Boltzmann Machines

A single Restricted Boltzmann Machine(RBM) consists of a layer of unconnected “visible” input units, that have undirected, symmetrical connections with another single layer of hidden units.As shown in figure 1, the network is fully connected between the two layers, yet no units within the same layer are connected to one another, forming a bipartite graph.Each connection between units of the two layers has an associated weight that must be learned, represented by a weight matrix.

Data in RBM can be initialized randomly and then alternating Gibbs sampling are performed to update them. Given the current states of the units in one layer, all the units of the other layer are then updated simultaneously. This process is repeated until the entire system is sampling from its equilibriumdistribution. RBMs are generative models, meaning that the training period results in a probability distribution of the training data being learned. When the model is later used for testing, it may encounter new, unfamiliar data, but the probability distribution

can account for these previously unseen occurrences, yielding a likely, probabilistic output [7].

RBM are composed of two layers of hidden h_j and visible v_i units, and consists of a matrix of weights $w_{i,j}$ associated with the connection between them, as well as bias weights a_i for the visible units and b_j for the hidden units. Given these a joint configuration (v, h) of the visible and hidden units has an energy given by:

$$E(v,h) = \sum_{i \epsilon visible} a_i v_i \sum_{j \epsilon hidden} b_j h_j \sum_{i,j} v_i h_j w_{i,j} \quad (1)$$

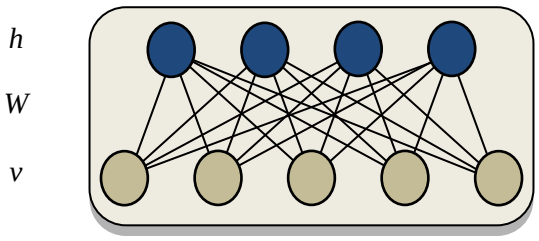

Figure 1. RBM network.

The network assigns a probability to every possible pair of a visible and a hidden vector via this energy function:

$$p(v,h) = \frac{1}{Z} e^{-E(v,h)} \quad (2)$$

where the "partition function", Z, is given by summing over all possible pairs of visible and hidden vectors:

$$Z = \sum_{v,h} e^{-E(v,h)} (3)$$

Since the RBM has the shape of a bipartite graph, with no intra-layer connections, the hidden unit activations are mutually independent given the visible unit activations and conversely, the visible unit activations are mutually independent given the hidden unit activations [3].So we can calculatethe conditional probability of a configuration of the visible units v, given a configuration of the hidden units h: $P(v|h)$, and Conversely the conditional probability of h given v: $P(h|v)$.

For m visible units and n hidden units, the individual activation probabilities are given by:

$$P(h_j = 1|v) = \sigma(b_j + \sum_{i=1}^{m} w_{i,j} v_i) \quad (4)$$

$$P(v_i = 1|h) = \sigma(a_i + \sum_{j=1}^{n} w_{i,j} h_j) \quad (5)$$

$\sigma(x)$: The logistic sigmoid function 1/ (1 + exp (−x))

The probability that the network assigns to a visible vector, v, is given by summing over all possible hidden vectors:

$$p(v) = \frac{1}{Z} \sum_h e^{-E(v,h)} \quad (6)$$

The weights can be adjusted by [9]:

$$\frac{\partial \log p(v)}{\partial w_{i,j}} = < v_i h_j >_{data} - < v_i h_j >_{model} \quad (7)$$

$$\Delta w_{i,j} = \epsilon(< v_i h_j >_{data} - < v_i h_j >_{model}) \quad (8)$$

Where the angle brackets are used to denote expectations under the distribution specified by the subscript that follows, and ϵ is the learning rate, the same learning rule is applied to the biases a_i and b_j.

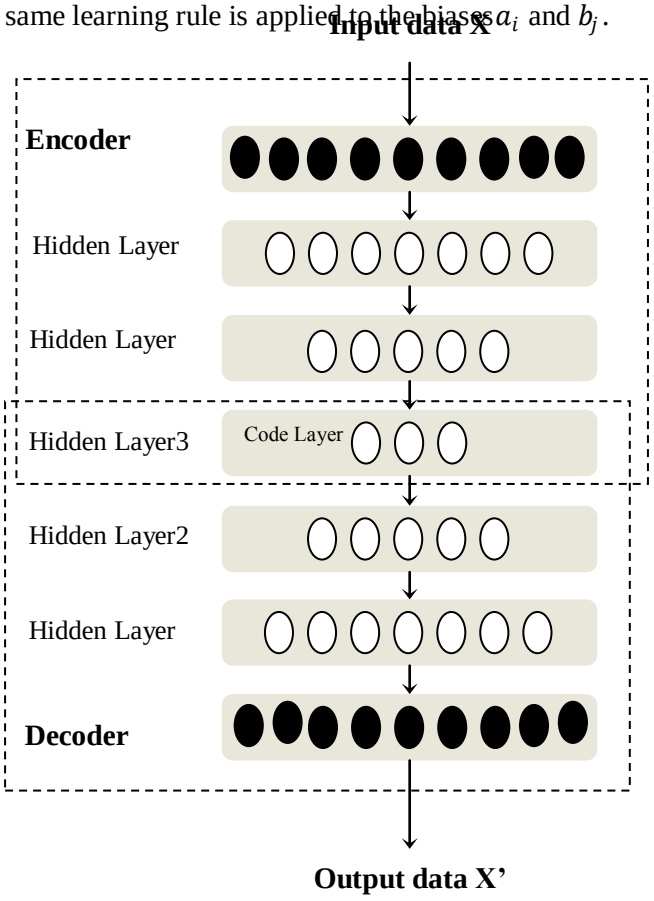

Figure 2. Multilayer autoencoder neural network architecture.

4. Stacked Autoencoder with RBM pre-training

4.1. Autoencoder architecture

Autoencoder networks are feedforward neural networks which can have more than one hidden layer. These networks attempt to reconstruct the input data at

the output layer, the targets at the output layer are the same as the input data, and thus the size of the output layer is also the same as the size of the input layer [14]. They can be used to reduce dimension of data.

As shown in figure 2 Autoencoder contains an encoder which transforms the input data into low-dimensional code, a low-dimensional representation and a decoder that reconstructs the original data from the low-dimensional representation.

Autoencoder can to be trained in order to minimize error reconstruction between the input X and output X' using backpropagation method, but this method isn't a good solution in multi hidden layers networks because the training can be slow. The most appropriate alternative for training is pretraining which will be represented in the next section.

4.2. Pretraining

The pretraining method of multilayer Autoencoders was proposed in [10].This type of Autoencoders uses RBM pre-training for each hidden layer. The idea is to start with a traditional one-hidden layer autoencoder then create a stack of RBM in which the outputs from the hidden layer are the inputs for training the next RBM layer.

4.3. Fine-tuning

After pretraining the individual RBM's at each level are unrolled to have a symmetric topologyof a deep autoencoder, then we fine-tune the weights to reduce reconstruction error of vector count data using backpropagation through the entire network. There are two error functions to perform backpropagation stage which are square error function, and cross entropy function [15].

5. Proposed method for Arabic Text Categorization

In this section, we present a new method for ATCto involve implicit semantic and to reduce dimensionality, this method relies on deep learning architecture. We explored Deep RBM Autoencoder to produce high level abstraction presentation,documents which are similar semanticallywill be near in the code space. Also we take advantage of short codesto have low-dimensional data representation.

As shown in figure 3, our system contains multiple blocs which are preprocessing bloc, deep learning bloc, and categorization bloc, the outputs of each bloc are the inputs for the next bloc.

The first bloc is preprocessing in which we have implemented different steps of Arabic text preprocessing:

- Delete stop word, punctuation marks, numbers, and words written in different languages from text.
- Normalizing the documents by replacing letters (" أ إ آ") with letter ("ا"), letters (" ؤ ئ") with ("ء"), and ("ا") with ("ى")
- Tokenization in which we divide text into units, in this work we take into consideration word as a unit.
- Stemming on words to find basic form of word, it can be explained as the process of removing prefixes, infixes, or/and suffixes from words to reduce these words to their stem or roots [16]. We applied two different techniques of stemming Light stemmer, and Khoja stemmer [16, 13].

After preprocessing, we use feature selection technique to select 2000 frequent words, then we create vector count of each document in the dataset. It is necessary totake into account normalized vector count for deep Autoencoder, for this all the values of each vectorare between 0 and 1.

The second bloc is deep learning,during this steps we have a deep Autoencoder that has a symmetric topology and creates a low dimensional code for input data. To train the deep Autoencoder we used RBM pretraining, during this step a stack of RBM is created. Then each RBM is unrolled to have symmetric model. Afterward we fine-tune using the deep Autoencoder usingsquare error function to reduce the reconstruction error between the input and output data.

Document representations produced from deep learning bloc will be the inputs for the third bloc that is categorization. To confirm the results, we used different machine learning algorithms: SVM, NB, andDT.

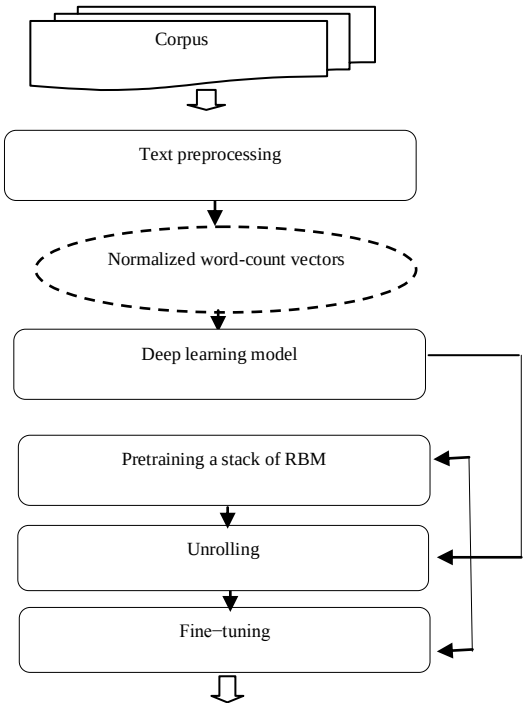

Figure 3. Flowchart of the proposed technique of text representation

6. Experiments and Results

This section concerned the experimental results and the assessment of them.

To evaluate our system a corpus was needed. The dataset used is CNN Arabic news (see table 1), it consists of 5070 documents, and fill into 6 categories:business, entertainment, middle_east, scitech, sport and world.

Table 1. CNN Arabic news corpus.

Categories	Number of documents
business	836
entertainment	474
middle_east	1462
scitech	526
sport	762
world	1010

To measure the effectiveness of our system, we have used four evaluation measures which are accuracy, precision, recall, and f-measure.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (9)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (10)$$

$$F1=2 * \frac{Precision * recall}{Precision + recall} \quad (11)$$

Table 2. Evaluation text categorization.

	Real class	Other classes
Predicted class	TP	FN
Other classes predicted	FP	TN

To study the effectiveness of deep learning representation we havedeveloped a categorization systembasing on tree classifiers which are SVM, NB, and DT. We used K-fold cross-validation to ensure that the system produces reliable results, K was set to 10. For all systems we have a deep autoencoder with 3 layers.The total number of training epochs employed is 200. The weights were updated using a learning rate of 0.1, momentum of 0.9.

In this work, Bag of Word (BoW) is adopted with multiple stemmers, with different architectures of deep Autoencoder.

Results using 32code-vectors

Table 3 and table 6show the macro average of all values of the different evaluation parameters which are precision, recall, and accuracy. We used 2000-500-500-32 architecture. Two stemmers are explored which are Light and Khoja stemmer.We can clearly note that Light stemmer is the best stemmer and we observed thatSVM outperformed the others classifiers.

Results using 64 code-vectors

Table 4 and table 7illustrate the results of using 64 code-vectors with 2000-500-500-64 architecture, two stemmers are explored which are Khoja and Light stemmer for the tree classifiers: SVM, DT, and NB.64 code-vectors is more accurate than 128 code-vectors and 32 code-vectors, it is clear that the SVM classifier has the best accuracy, then the DT comes in the second place, and the worst classifier is NB.

Results using 128 code-vectors

Table 5 and table 8show the result of using 2000-500-500-128 architecture,we notice that Light stemmer outperformed the Khoja stemmer, and SVM classifier has the best accuracy, then the DT comes in the second place, and NB the worst classifier.

By observing figure 4, we can notice that the best system is the system that uses 2000-500-500-64 architecture and Light stemmer, because it gives f1-measure value that equal 0.677 for NB, 0.686 for DT, and 0.753 for SVM.

Table 3. Categorization results using 32- code vectors and Light stemmer.

Classifier/evaluation measure	**NB**	**DT**	**SVM**
Accuracy	0.65	0.687	0.748
Precision	0.662	0.689	0.758
Recall	0.634	0.685	0.742

Table 4. Categorization results using 64- code vectors and Light stemmer.

Classifier/evaluation measure	NB	DT	SVM
Accuracy	0.673	0.686	0.751
Precision	0.685	0.688	0.762
Recall	0.669	0.685	0.745

Table 5. Categorization results using 128- code vectors and Light stemmer.

Classifier/evaluation measure	NB	DT	SVM
Accuracy	0.673	0.681	0.751
Precision	0.684	0.684	0.762
Recall	0.669	0.68	0.749

Table 6. Categorization results using 32- code vectors and Khoja stemmer.

Classifier/evaluation measure	NB	DT	SVM
Accuracy	0.562	0.573	0.717
Precision	0.589	0.589	0.706
Recall	0.587	0.589	0.684

Table 7. Categorization results using 64- code vectors and Khoja stemmer.

Classifier/evaluation measure	NB	DT	SVM
Accuracy	0.567	0.615	0.728
Precision	0.595	0.591	0.718
Recall	0.589	0.593	0.693

Table 8. Categorization results using 128- code vectors and Khoja stemmer.

Classifier/evaluation measure	NB	DT	SVM
Accuracy	0.506	0.619	0.722
Precision	0.594	0.593	0.711
Recall	0.586	0.590	0.689

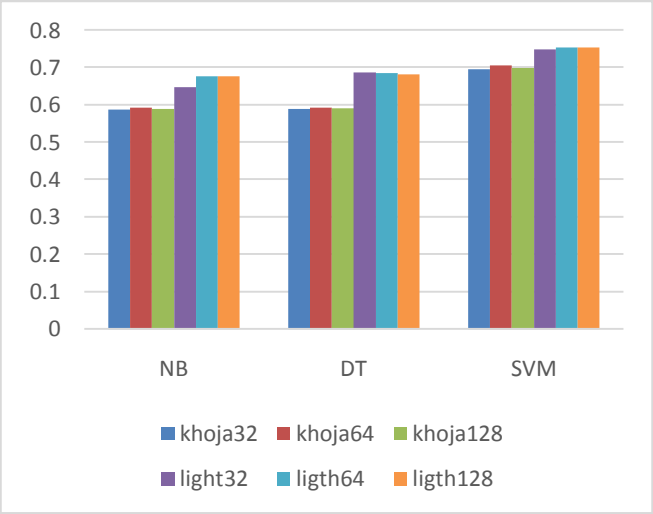

Figure 4. F-measurefor differentevaluated systems.

7. Conclusion and Future Work

In this paper we described an efficient method of ATC using deep learningapproach, we investigated a novel representation method that produced high level representation which learnsautomatically hidden semantic andproduces short codes. We use deep stacked autoencoder which has as input word-count vectors. In the pretraining stage we used RBM, then the model is unrolled to create deep network, during fine-tuning stage backpropagation is used. As classifiers we used SVM, NB, and DT. The obtained results show that exploring deep autoencoder representation performed well in ATC system especially for SVM classifier.

In the future work, we plan to add feature selection bloc and integrate different semantic resources to compare the performance of systems using implicit and explicit semantic.

References

[1] Abu-Errub A., "Arabic Text Classification Algorithm using TFIDF and Chi Square Measurements", *International Journal of Computer Applications*, vol. 93, no. 6, 2014.

[2] Al-Salemi B. and Aziz M. J. Ab., "Statistical Bayesian Learning for Automatic Arabic Text Categorization", *Journal of computer Science*, vol. 7, no. 1, pp. 39, 2011.

[3] Carreira-Perpinan M. A. and Hinton G. E., "On Contrastive Divergence Learning", *AISTATS*, vol. 10, pp. 33-40.

[4] Duwairi R., "Arabic Text Categorization", *Int. Arab J. Inf. Technol.*, vol. 4, no. 2, pp. 125-132, 2009.

[5] Elberrichi Z. and Abidi K., "Arabic Text Categorization: A Comparative Study of Different Representation Modes", *Int. Arab J. Inf. Technol.*, vol. 9, no. 5, pp. 465-470, 2012.

[6] El Hassani I. and Masrour T., "Arabic Semantic Text Classification Based on Wavelet Spectral Analysis", *International Journal of Advanced Research in Computer and Communication Engineering*,vol. 2, Issue 6, 2013.

[7] Gravelines C.,"Deep Learning via Stacked Sparse Autoencoders for Automated Voxel-Wise Brain Parcellation Based on Functional Connectivity", (*Doctoral dissertation, The University of Western Ontario*), 1991.

[8] Hadni M., El Alaoui S., and Lachkar A.," Word Sense Disambiguation for Arabic Text Categorization". *International Arab Journal of Information Technology (IAJIT)*, vol. 13, 2016.

[9] Hinton G. E., "A practical guide to training restricted Boltzmann machines", *Momentum*, 2010, vol. 9, no. 1, pp. 926.

[10] Hinton G. E. and Salakhutdinov R. R., "Reducing the Dimensionality of Data with Neural Networks", Science, vol. 313, pp. 504–507, 2006.

[11] Odeh A., Aymen A., Shambour Q., and Turab N., "Arabic Text Categorization Algorithm using Vector Evaluation Method", *International Journal of Computer Science & Information Technology (IJCSIT)* , vol. 6, no. 6, 2014.

[12] Salakhutdinov R. and Hinton G. E., "Semantic Hashing", RBM, vol. 500, no. 3, pp. 500, 2007.

[13] Sawalha M. and Atwelle S., "Comparative evaluation of arabic language morphological analysers and stemmers", *Proceedings of COLING 2008 22nd International Conference on Comptational Linguistics (Poster Volume))*, Coling 2008 Organizing Committee, pp. 107-110, 2008.

[14] Tan C. C. and Eswaran C., "Performance Comparison of three types of Autoencoder Neural Networks", *2008 Second Asia International Conference on Modelling &# x00026; Simulation (AMS)*,pp. 213-218, 2008.

[15] Tian X., Hérault R., Gasso G., and Canu S., "Pré-apprentissage Supervisé pour les Réseaux Profonds", *Proceedings of Rfia*, pp. 36, 2010.

[16] Yousif S. A., Samawi V. W., Elkabani I., and Zantout R., "The Effect of Combining Different Semantic Relations on Arabic Text Classification", *World of Computer Science & Information Technology Journal*, vol. 5, no. 6, 2015.

[17] Zrigui M., Ayadi R., Mars M., and Maraoui M., "Arabic Text Classification Framework Based on Latent Dirichlet Allocation ". *CIT. Journal of Computing and Information Technology*, vol. 20, no. 2, pp. 125-140, 2012.