



Text Classification Using Convolutional Neural Networks

Sara Muslih Mishal ^{1*} and Murtadha M. Hamad²

¹ Department of Computer Science, University of Anbar, Ramadi, Iraq

² Department of Computer Science, University of Anbar, Ramadi, Iraq

Emails: sar19c1006@uoanbar.edu.iq , dr.mortadha61@uoanbar.edu.iq

Abstract

Most of the information (more than 80%) is stored as text, and text mining is a very important process as it is an initial step in the process of text classification, and this is especially the case in the Arabic language. The Aim of The Study is to classify Arabic texts according to specific categories using advanced performance indicators We used Data Templates as a platform for managing and organizing Apache Spark to solve big data challenges. Apache Spark offers several integrated language APIs. nlp lib was used for text processing. The data is pre-processed through several steps, namely separating the words into one text on the basis of the space between words, cleaning the text of unwanted words, restoring the words to their roots, as well as the feature selection process is a critical step. in text classification. It is a preprocessing technology. In this paper, one way to determine which TF attributes are used how often each feature appears in the document is that they consider the first level of the feature selection process. Then we use TF-IDF to determine the significance of the feature in the document, and this is the last step in the preprocessing Outcomes Text classification . Results were evaluated using advanced performance indicators such as accuracy, Precision and recall. A high accuracy of 96.94% was achieved. The main objective of this paper is to classify basic texts quickly and accurately, according to the results as long as the feature size is suitable, the most advanced technology is superior to other pass rate methods due to the reasonable reliability and perfect pruning level.

Keywords: Text Mining, Text Classification, CNN, Apache Spark, Databricks.

1. Introduction

Text Classification is one of the most important elements in the field of Text Mining (TM). The classification of Arabic texts has its own problems and limitations deriving from the nature of the Arabic language . Four classifiers are available, including a suggested algorithm. The Arabic corpus applied. The classifiers used will follow Arabic text (tokenization, stemming, stopping) preprocessing Word removal) calculation and tfidf weight calculation of each feature

Because of the different form, structure, and components in the Arabic language, the challenges of text classification are obvious. Moreover, there are not enough studies dealing with the classification of the Arabic script. Autotext classification has been a topic of application and important research since digital documents were first produced. Today, due to the very large volume, text classification is a must, and we have to deal with the number of text documents every day

The CNN algorithm was used in this study to classify Arabic text. The work is focused on proposing a classification algorithm applied to different texts from the Arabic language: economic, cultural, political and technical. Considering that the goal of our study is to take use of the algorithm's proven benefits in other disciplines, in Arabic text.

The trained models on datasets of various sizes (data 27k, data 55k, data 83k, data 111k), and the number of documents in each size of the dataset (the bigger the dataset we use, the greater accuracy we acquire). Using CNN's model, we can see that it performs well for all sizes. Traditional approaches relying on the bag-of-words model fail to capture as much contextual information as the CNN model, which is shown to be more successful. In addition, the training and testing data is gathered from a variety of sources and distributions, and in many instances the documents may be categorized in numerous categories. Each of SANAD's three training datasets was used to train the suggested CNN models. As a final step, we applied the trained model to each of the datasets: On Arabiya, Khaleej, and Akhbarona testing datasets. The main contribution of this study are:

- Online Arabic news items have no uniform categorization, which makes it impossible to explore them in the form of an aggregate collection (as opposed to a traditional print edition).
- The news services utilize their own unique taxonomies, which do not have enough consistency to be of use to other organizations. Not only was it unclear how best to categorize Arabic news articles according to a particular taxonomy, but it was unclear what would be the optimum approach. Even yet, stemming is intended to be a component of the preprocessing processes, which help classify information better.
- We used Databricks as a platform for managing and curating Apache Spark to solve big data challenges. The `nlp` lib was used to pre-process texts.
- The work is focused on proposing a classification algorithm applied to different texts from the Arabic language: economic, cultural, political and technical.
- Apache Spark offers various integrated language APIs. The proposed classification algorithm is applied to these transcripts to obtain the highest value for accuracy and recording time. This function was well performed in the Sanad Group.

The rest organization of this study are: Section 2 is related work that reviews the recent studies in text classification. The proposed Text Classification System Design presented in Section 3. Section 4: the results discussion on text classification using CNN. Finally, the conclusion and future work presented in section 5/

2. Related Works

Nidaa F. et al, 2018 Presented, an algorithm involving audio features (mean, standard deviation, zero crossing, Amplitude) and Support Vector Machines (SVMs) has been presented to perform speaker gender recognition. For every audio, the highlights vector has been used as an info vector in the Support Vector Machines (SVM) algorithm. An example of 2270 audio files, include 1132 female audio with 1138 male audio have been analyzed according to this algorithm. With only four features, the average error of prediction is 5% [2]. Bashar M. 2017 presented the preprocessing signal for speech emotion recognition was introduced. The discrimination between speech and music files was performed depending on a comparative between more than one statistical indicator such as mean, standard deviation, energy and silence interval. The preprocessing include silence removal, pre-emphasis, normalization and windowing

so it is an important phase to get pure signal which is used in the next stage (feature extraction). The wave files (male, female) and the music file which are used in this paper have sample rate [3].

Rajeswari P. and Juliet K. 2017 focus on text categorization using the Naive Bayes and K-Nearest Neighbor classifiers, with an emphasis on performance and accuracy utilizing the Rapid miner for Student Data Set. Opinion mining, sentiment analysis, labeling, and text object identification from news documents are examples of text or document classification applications. The results of the experiment reveal that the Naives Bayes classifier is a better classifier than the KNN classifier, which has an accuracy of 38.89.[4]. Soumya G. and Shibily J. 2014 This study presented a method for detecting co-occurrence features in wikipedia pages' anchor text, as well as a method for incorporating co-occurrence features into the BOW model. Finally, the approach is evaluated to see how well it performs in text categorization tasks. The results suggest that the co-occurrence feature at the document level has aided classification and that text categorization has improved. In Information Retrieval, co-occurrence may also give useful indexes[5].

Menaka S.et al 2013 examines this Text classification is the process of categorizing text documents according to a set of predetermined categories using words, phrases, and word combinations. Keywords are a group of words that carry the most crucial information about a document's content. TFIDF and WordNet are used in the proposed method to extract keywords from texts. The performance of the Naive Bayes, Decision Tree, and K-Nearest Neighbor (KNN) algorithms was tested, and the results were assessed. When compared to other algorithms, the decision tree approach provides superior text categorization accuracy [6]. V. Srinivas, et al 2013 In This work they presented a simple text dependent speaker identification method which was based on the Symlet wavelets for extracting features. Those features were afterwards classified with the use of algorithms of data mining. In this work, Naive Bayesian(NB), J48 and Support Vector Machines (SVM) have been utilized for feature classification[7] .

The proposed system is suitable for applications with large data sets, providing efficient access to this data. To evaluate the DL models, they are given accuracy, recall, F1-score, and a false alarm rate as well as a true negative rate and a false negative rate. the confusion matrix includes characteristics that provide the foundation for the assessment measures. In the confusion matrix, TP and TN values correspond to feature classes occurrences being correctly predicted.

3. The Proposed Text Classification System Design

The design of any system is very important because it shows how the system is working and explains the exact steps and processes that will be performed to get the desired need from it. The proposed system in this thesis is the text

classification especially Arabic news text depended on previous knowledge by using proposed CNN model. This diagram demonstrate the planned feature extraction and classification step in the CNN process shown below in Figure 1 [1].

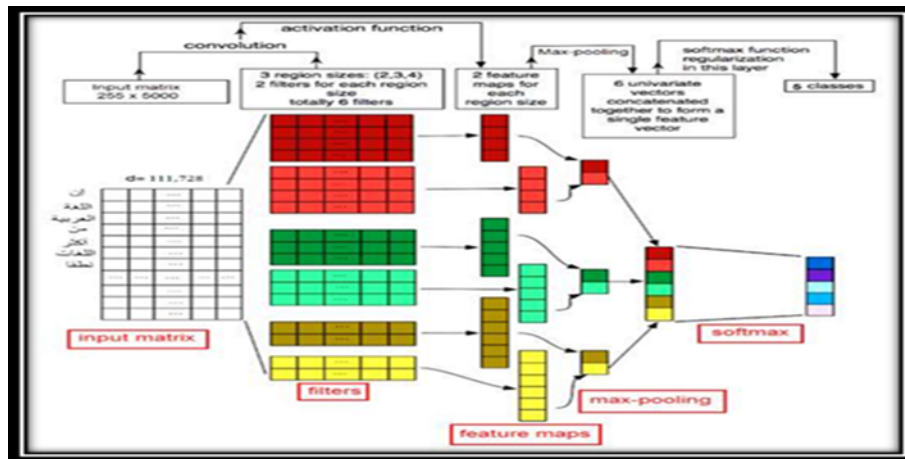


Figure 1. The proposed Convolutional Neural Network (CNN) Architecture for Arabic Text Classification

The proposed system will pass through multi phases and process as the following:

A. Data Collection/Creation: is the collection of Arabic news materials with the goal of categorizing them into various groups based on their topics.

B. Text Preprocessing: preprocess the text to make it acceptable for categorization by employing preprocessing activities. Processing procedures like as tokenization, Stemming and lemmatizing and removing stop words are used in this phase.

C. Feature Weighting and Selection: these approaches will be used to weight and choose each word in the text, and the location of the word or its meaning will have no effect on the classification process..

D. Data Splitting: Typically, the dataset is divided into two parts, one for training the data and the other for testing the developed classifier model.

- **D.1 Training:** An algorithm is used to classify the data into a classifier model from the first section.
- **D.2 Testing:** Using the second set of data, the classifier model is tested to see whether it accurately predicts the class of each input. After that, it calculates the error rate as well as other important performance metrics including precision, accuracy, recall, and the F1-measure.

E. Classification: Building a model based on CNN. The proposed deep CNN model and work to classify Arabic news text. Figure 5 show the block diagram of the proposed system.

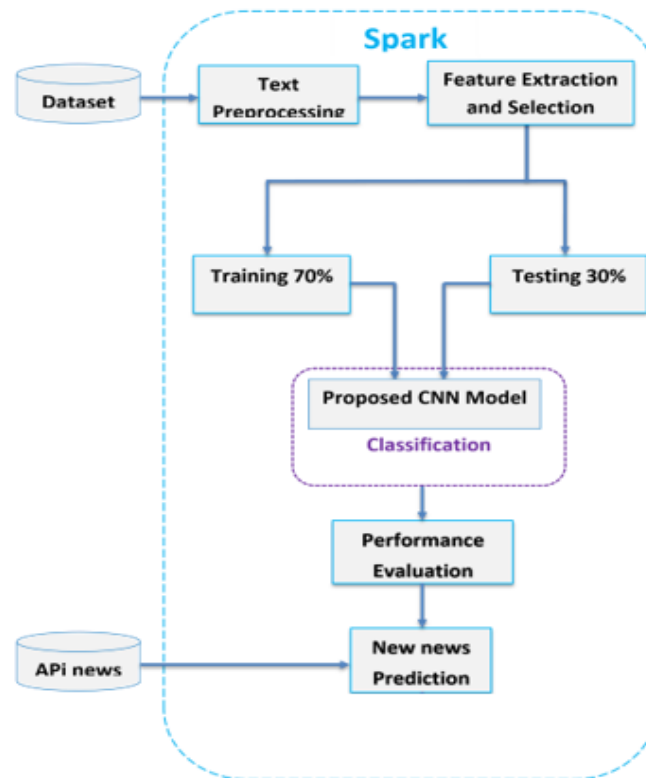


Figure 2. Block Diagram of The Proposed System

3.1 Data Collection

In this part, the data set is clarified and how to deal with it and pre-process stage.

3.1.1 SANAD

Arabic News Articles Dataset for Automatic Text Classification. NLP activities like Text Classification and Word Embedding may be performed using the SANAD Dataset, a vast collection of Arabic news stories. Three well-known news websites' stories were gathered using Python scripts: AlKhaleej, AlArabiya, and Akhbarona. One exception to this rule is the dataset AlArabiya which does not contain a category for religion. SANAD has more than 190,000 articles. with the exception of Since the data were collected from the website of the news, the articles are expressed through MSA, so no dialects are concerned. From that time on, we grouped the data sets in a single tag and called SANAD. The data sets are divided into training and test sets, Bulk articles are preprocessed to provide a clean copy after removal punctuation and Latin alphabet[7].

3.1.2 Dealing With Datasets

The data is formatted as follows: there are three folders; a separate folder for each source of news website. Each folder has sub-folders that carry the title of the categories or labels. Each sub-folder contains a list of text files numbered sequentially, in which a file corresponds to one whole article. Figure 6 shows an example of an article that is categorized as —Finance and belongs to —Al-Arabiya dataset.

1. Unzipping compressed files is the first step.
2. The packed file must be unzipped.
3. There are three primary folders: Akhbarona, Khaleej, and Arabiya, which include the three datasets.
4. Subfolders for each category are identified with the category's name and include 6-7 subfolders.
5. In addition, each category has its own subfolder containing a collection of articles.



Figure 3. An Example of Text Document

3.2 Text Preprocessing Stage

Preprocessing is the second step of the proposed system. Many meaningless terms are found in papers therefore it is required to remove them from the document's text in order to improve categorization and thus minimize the dimension. For preprocessing the text, tokenization, lemmatization, and deletion of stop words are three sub-steps[8].

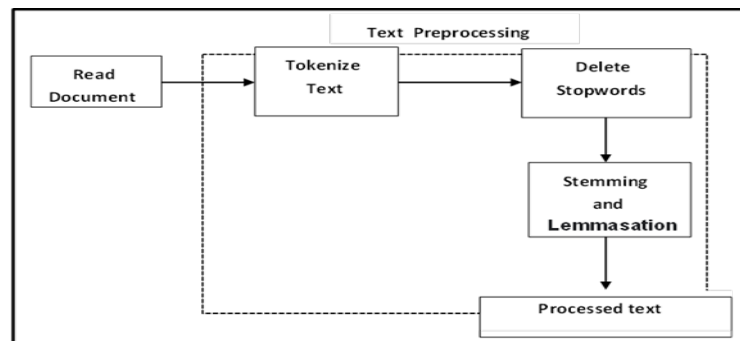


Figure 4. Text Preprocessing Phase

3.2.1 Tokenization

Tokenization is the initial stage in text preparation. By separating words with spaces, the text will be tokenized, or broken up into smaller chunks. Symbols, phrases, or numbers may all be used as tokens. Tokenization is mostly used to identify relevant terms[9].

3.2.2 Delete Stopwords

The second stage in text preparation is to remove stop words. Natural Language Processing (NLP)-based text preprocessing is efficient since its primary function is to eliminate nonsensical and unnecessary words from a document in order to produce a more usable text document. In the categorization process, a stop word may be characterized as a term that is neither significant nor crucial. It's common knowledge that eliminating stop words results in a loss in dimension[10].

3.2.3 Stemming and lemmatization

Stemmed words are words that have been stripped of their suffixes in order to reduce the number of words, find the stem, and save time and memory space. Stemming is the process of doing this. To acquire the Arabic token's stem, all of the suffixes must be removed. These steps are explained in algorithm (1). This algorithm involves several steps after reading the text document, the first step is to cut or separate the words from each other, depending on the space between the words, and then the numbers, symbols and punctuation are removed; the second step is to read the result of the first step and remove unnecessary and illogical words; the third and final step is to return the words to their roots, i.e. strip the word of its suffixes to reduce the number of words [11].

Algorithm (1): Preprocessing Algorithm

Input: Collection of documents (T).

Output: (Tokenization-List ((T)), Stop word removing-List (T), Stem-List.

Begin

Step1: Read the document from data set as follow:

While not (EOF)

-separate every word from other based on space for obtaining (tokens).

-eliminate all numbers & symbols token:

 if no. of token > 0 then i= i+1

 End if

 Tokenization-List = Token[i] \ \ tokenize the data

 End While

- Return (Tokenization-List (T))

Step 2: Read the result from previous step then do as follow:

- take the text which obtain from previous step then:

 While (Tokenization-List != null) do

 Put word in new list called stop word removing-List (T) \ \ set stop words for separators

 End while

-For each word in T do

 If word = stop word then

 Remove word

 Else

 Stop word removing-List (T) = word

 End if

-Return (Stop word removing-List (T))

```
Step 3: for each token in (stop word removing(list) import
(ISRI)algorithm in python library then return (stem-list)
#( using heavy kind in algorithm to return the root)
End
```

3.3 .Feature Selection Phase

The fourth step in the text categorization system is feature extraction. It is an important step to extract characteristics from documents after they have been preprocessed. This model, which is also known as a bag of words model, is used to extract the characteristics. The procedure for extracting the characteristics is as follows[12]:

3.3.1 Building the Vocabulary

A vocabulary may be constructed by constructing a list of features retrieved from the training set, such as (key, string) for each feature. Each feature has a unique Key and a unique String to identify it. Bag-of-words models have a tendency to reorder words in a random fashion, thus this list doesn't keep the original phrases' word order.

3.3.2 Documents Representation

To represent each document as an array of features, the whole text of the documents must be converted to a documents vector. As a result, a matrix with one row for each document and one column for each characteristic in the list of vocabulary may be used to describe the collection of text documents. The length of this vector is the same for all documents since it is dependent on the length of the vocabulary. Documents are represented as (w1, w2, w3, w4,..., wn) where wi is the weight of feature I in document D. The weight of each feature defines the document's relevance. The techniques for assigning weights to features are outlined in the following paragraphs:

a)Term Frequency (TF)

TF is determined for each feature in the document by counting the number of times each feature appears in the document. Make the feature's significance more apparent in a document.

b)Inverse Document Frequency (IDF)

A typical feature weighting system is IDF, which is used to evaluate the value of a feature throughout the collection of documents rather than just one document. When a characteristic is found in just a few documents, the IDF relies on the assumption that this feature will be a good discriminator of the documents in the collection.

3.3.3Term Frequency – Inverse Document Frequency TF_IDF

It works by comparing the inverse percentage of a feature's occurrence in a given text to the relative frequency of that feature in the training set. Intuitively, how important a certain characteristic in a text really is determined by this formula. The goal of the step is to extract text properties from documents after the preprocessing step to get a word bag where the entry is set of tokens for each document (T). A list of properties is generated for each property that has a unique key and string. Then the document is converted to a vector containing the word weight and the word is specified TF word weight, which specifies the number of times the feature appears in the document while IDF calculates the weight of the word in the set of documents instead of a single document The last step in the algorithm is to calculate TF_IDF where the percentage is compared to the inverse percentage. The presence of a feature in a given text with relative frequency for that feature in the training set. The output will return an array with one row for each document and one column for each feature that occurs in the vocabulary list, and each feature associated with it. Weight (TFIDF) determines the significance of the feature in the document.

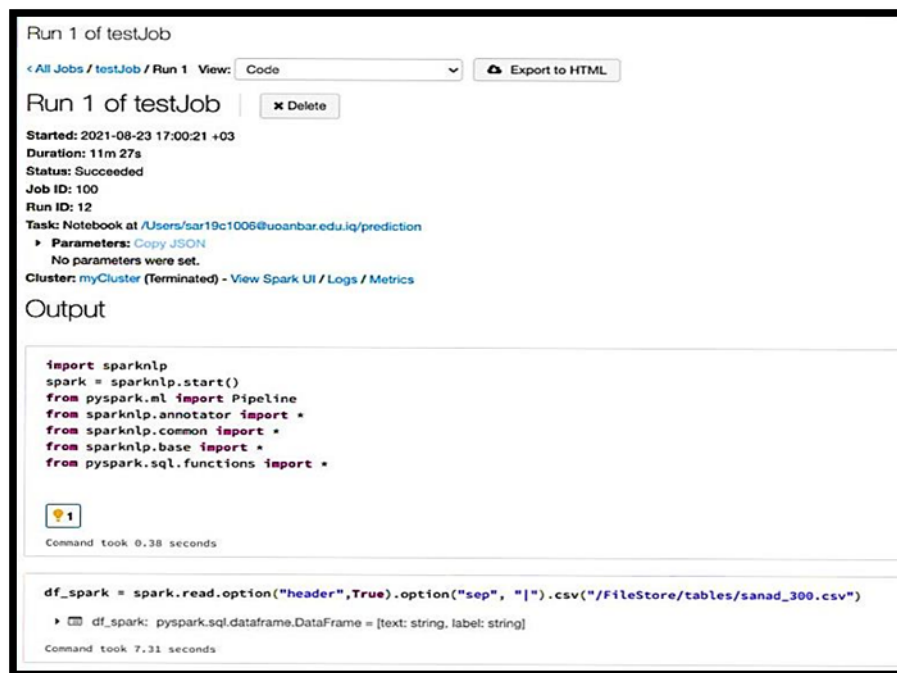
3.4. Model Training

Text classification relies heavily on this step. The training set is selected from the corpus, the learning is performed on it, and the model is generated. To reiterate the findings from researching the experts above, this section provides unambiguous definitions of the three terms. Training Dataset: The sample of data used to fit the model. The actual dataset that we use to train the model (weights and biases in the case of a Neural Network). The model sees and learns from this data[13].

- Training Dataset: The sample of data used to fit the model.
- Validation Dataset: The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyper parameters. The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration.
- Test Dataset: The sample of data used to provide an unbiased evaluation of a final model fit on the training dataset.

3.5. Testing and Evaluation

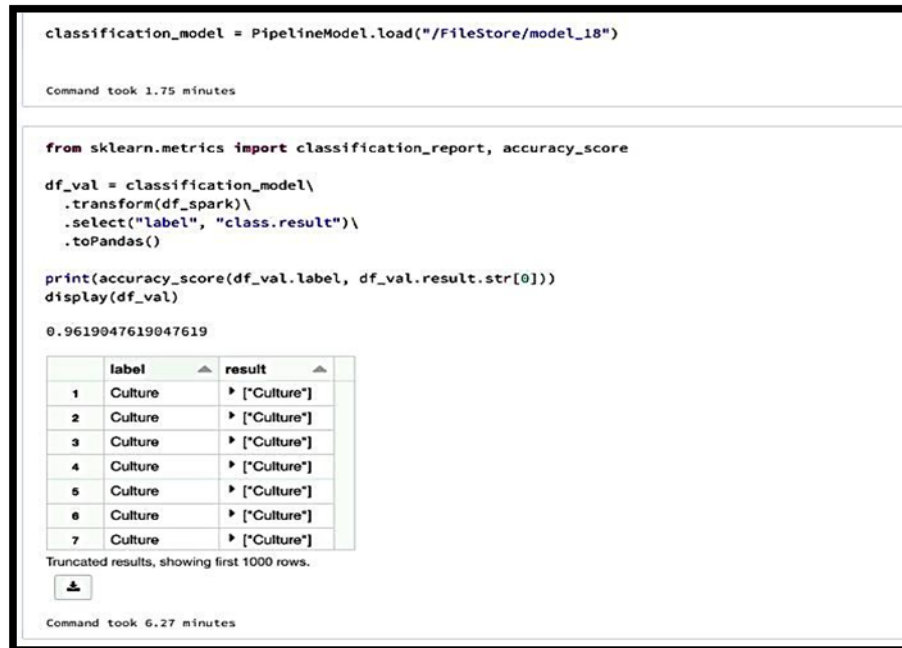
The evaluation of a model skill on the training dataset would result in a biased score. Therefore, the model is evaluated on the held-out sample to give an unbiased estimate of model skill. This is typically called a train-test split approach to algorithm evaluation. Classification is carried out on a portion of the corpus known as the testing set, which was prepared in the preceding stage. Next, we choose an appropriate index to evaluate the model's performance. The validation dataset is different from the test dataset that is also held back from the training of the model, but is instead used to give an unbiased estimate of the skill of the final tuned model when comparing or selecting between final models. A validation dataset is a sample of data held back from training your model that is used to give an estimate of model skill while tuning model's hyper parameters.



The screenshot displays the 'Run 1 of testJob' interface. At the top, there are navigation links for '< All Jobs / testJob / Run 1' and a 'View:' dropdown set to 'Code', along with an 'Export to HTML' button. Below this, the job title 'Run 1 of testJob' is shown with a 'Delete' button. The job details include: Started: 2021-08-23 17:00:21 +03, Duration: 11m 27s, Status: Succeeded, Job ID: 100, and Run ID: 12. The task is identified as 'Notebook at /Users/sar19c1006@uoanbar.edu.iq/prediction'. Under 'Parameters', it states 'Copy JSON' and 'No parameters were set.'. The cluster is 'myCluster (Terminated)'. The 'Output' section shows two code blocks. The first block contains import statements for sparknlp, spark, Pipeline, and various classes from sparknlp and pyspark.sql.functions. The second block shows the execution of 'df_spark = spark.read.option("header", True).option("sep", "|").csv("/FileStore/tables/sanad_300.csv")' and its output as a DataFrame with columns 'text' and 'label'.

Figure5. Screen Shot of The Spark Data Testing

The development of automatic segmentation algorithms is essential for the creation of an automated diagnostic system for the diagnosis of skin lesions in the context of skin lesion detection. This paper addresses some procedures for segmenting skin lesions, including dermatoscopy. We have provided a quick summary of the various segmentation algorithms that are currently being utilized for dermoscopic image analysis, which are included in Table 1.



```

classification_model = PipelineModel.load("/FileStore/model_18")

Command took 1.75 minutes

from sklearn.metrics import classification_report, accuracy_score

df_val = classification_model\
    .transform(df_spark)\
    .select("label", "class.result")\
    .toPandas()

print(accuracy_score(df_val.label, df_val.result.str[0]))
display(df_val)

0.9619047619047619



|   | label   | result      |
|---|---------|-------------|
| 1 | Culture | ["Culture"] |
| 2 | Culture | ["Culture"] |
| 3 | Culture | ["Culture"] |
| 4 | Culture | ["Culture"] |
| 5 | Culture | ["Culture"] |
| 6 | Culture | ["Culture"] |
| 7 | Culture | ["Culture"] |



Truncated results, showing first 1000 rows.

Command took 6.27 minutes

```

Figure 6. Screen Shot of The Spark Data Testing

3.6. Classification Phase use Convolutional Neural Networks

In the text classification system, classification is the fifth step of the process. When a text (a "class label") is sent to the proposed system, it is evaluated against a set of pre-defined classes in order to determine its classification. For classification, the retrieved features are sent into the CNN classifier. In most classification algorithms, a feature vector is utilized as an input to the algorithm. Therefore, specialists are required to identify the intended operation's feature vectors. Instead of requiring an expert to build the feature vector, a learning algorithm may be used to automatically extract it. It is possible for deep learning algorithms to automatically discover characteristics that are relevant for the classification job. Deep learning algorithms include a variety of versions, including convolutional neural networks (CNN). Neurons in Convolutional Neural Networks feature learnable biases and weighted weights, like conventional Neural Networks. Dot products and non-linear functions are performed by each neuron based on the inputs it receives. From inputs to output classes, the complete system represents a single differentiable scoring function. The last layer still has a loss function. The outputs of each layer are often used as inputs to the following layer in a CNN. There are three types of layers: convolutional, pooling, and fully linked. The CNN algorithm was used to classify Arabic text. Given that the goal of this project is to take use of the algorithm's benefits, which have been shown in other disciplines, the algorithm will output features of the texts classified either as stop words or features of the text type, these features will later be used to help detect the type of the text content (Sport, News, Politicsetc)[14].

4. Result and Discusion

We trained models on datasets of various sizes (data 27k, data 55k, data 83k, data 111k), and the number of documents in each size of the dataset (the bigger the dataset we use, the greater accuracy we acquire). Using CNN's model, we can see that it performs well for all sizes. Traditional approaches relying on the bag-of-words model fail to capture as much contextual information as the CNN model, which is shown to be more successful. In addition, the

training and testing data is gathered from a variety of sources and distributions, and in many instances the documents may be categorized in numerous categories. It is more than satisfactory to get an accuracy rate over 96 percent when dealing with such difficult cross-domain categorization assignments as these. Each of SANAD's three training datasets was used to train the suggested CNN models. As a final step, we applied the trained model to each of the datasets. On Arabiya, Khaleej, and Akhbarona testing datasets. we describe the final results of the proposed system that works on the data splits (90/10). Where the work is focused on proposing a classification algorithm applied to different texts from the Arabic language. economic, cultural, political and technical. We used databricks as a platform for managing and curating Apache Spark to solve big data challenges. The nlp lib was used to pre-process texts. Apache Spark offers various integrated language APIs. The proposed classification algorithm is applied to these transcripts to obtain the highest value for accuracy and recording time. This function was well performed in the Sanad Group.

When it came to classifying Arabic document texts, the proposed system performed well in data split (70/30) achieving The highest value of accuracy 96% in sport, and less value 91% in Economy ,and highest value of Precision is 91% in sport and less value is 80% in sport , and highest value of recall is 94% in Politics and less value is 88% in sport , and highest value of f_score is 91% in Science and less value in sport is 77% as shown in table(1)

Table(1) Average Accuracy, Precision, Recall, And F-Score Of CNN Algorithm Without Using Arabic Phrases As Features At 70/30 Data Split

Category	Accuracy	Precision	Recall	F-score
Sport	0.96	0.804	0.883	0.776
Culture	0.935	0.871	0.913	0.805
Politics	0.923	0.902	0.942	0.793
Science	0.921	0.893	0.931	0.911
Economy	0.911	0.915	0.904	0.907

5. Conclusion

The adopted method to implement our system is to get Arabic text documents, preprocess them, build a parallel computing model with Apache Spark, implement using MLlib over Apache Spark, and then run a parallel classification using Apache Spark. The tests are run on a solitary Apache Spark cluster that has 16 worker driver nodes. Multiple measures have been used to assess the classification of the suggested method, we have utilized the classification metrics accuracy, precision, recall, and f-measure. We created a standardized Arabic classification system (taxonomy) to enable browsing services for online Arabic newspapers in order to improve information retrieval and natural language processing. The data is organized according to the same structure, which assists us with categorizing it. To develop this taxonomy, experts in the area, as well as volunteers, put together a taxonomy evaluation panel that worked from a widely used news taxonomy. This means that the most commonly used standard for media-based content metadata is what this taxonomy is based on. For future works can this extension can operate with bigger computer clusters that have greater memory capabilities and higher volume of Arabic texts in excess of 10 gigabytes. Testing several classifying algorithms in the proposed method using real-time data that is found in many forms, including pictures, videos, and documents. The methodology may be utilized with many different cloud-based software platforms, including big data analytics and web services, where data mining techniques are preferable over frameworks that support the MapReduce paradigm to provide excellent outcomes in systems and services.

References

- [1] S. Salloum, R. Dautov, X. Chen, P. X. Peng, and J. Z. Huang, "Big data analytics on Apache Spark," *Int. J. Data Sci. Anal.*, vol. 1, no. 3–4, pp. 145–164, 2016, doi: 10.1007/s41060-016-0027-9.
- [2] Hassin and Salah, "Gender Classification Based On Audio Features," *Mamoun Journal*, p. 196, 2018, doi: 10.36458/1253-000-031-011.

- [3] B. M. Nema and A. A. Abdul-Kareem, "Preprocessing signal for Speech Emotion Recognition," *Al-Mustansiriyah J. Sci.*, vol. 28, no. 3, pp. 157–165, 2018, doi: 10.23851/mjs.v28i3.48.
- [4] R. R.P, K. Juliet, and A. hana, "Text Classification for Student Data Set using Naive Bayes Classifier and KNN Classifier," *Int. J. Comput. Trends Technol.*, vol. 43, no. 1, pp. 8–12, 2017, doi: 10.14445/22312803/ijett-v43p103.
- [5] S. George K and S. Joseph, "Text Classification by Augmenting Bag of Words (BOW) Representation with Co-occurrence Feature," *IOSR J. Comput. Eng.*, vol. 16, no. 1, pp. 34–38, 2014, doi: 10.9790/0661-16153438.
- [6] K. Grabczewski, *Meta-learning in decision tree induction*, vol. 498, no. June. 2014.
- [7] M. Biniz, "Arabic Text Classification Using Deep Learning Technics," no. April 2019, 2018, doi: 10.14257/ijgcd.2018.11.9.09.
- [8] M. Kannan, S., Gurusamy, V., Vijayarani, S., Ilamathi, J. & Nithya, "Preprocessing Techniques for Text Mining Preprocessing Techniques for Text Mining," *Int. J. Comput. Sci. Commun. Networks*, vol. 5, no. October 2014, pp. 7–16, 2015.
- [9] L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, and J. M. Zurada, "Artificial intelligence and soft computing: 15th international conference, ICAISC 2016 Zakopane, Poland, June 12-16, 2016 proceedings, Part I," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, no. ML, pp. 621–630, 2016, doi: 10.1007/978-3-319-39378-0.
- [10] A. El Kah and I. Zeroual, "The effects of Pre-Processing Techniques on Arabic Text Classification," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 10, no. 1, pp. 41–48, 2021, doi: 10.30534/ijatcse/2021/061012021.
- [11] N. Pavlopoulou, A. Abushwashi, F. Stahl, and V. Scibetta, "A Text Mining Framework for Big Data," *Expert Updat.*, vol. 17, no. 1, 2017, [Online]. Available: <https://www.exonar.com/platform/%0Ahttp://centaur.reading.ac.uk/70108/>.
- [12] P. Grover and A. K. Kar, "Big Data Analytics: A Review on Theoretical Contributions and Tools Used in Literature," *Glob. J. Flex. Syst. Manag.*, vol. 18, no. 3, pp. 203–229, 2017, doi: 10.1007/s40171-017-0159-3.
- [13] H. Sayed, M. A. Abdel-Fattah, and S. Kholief, "Predicting potential banking customer churn using Apache Spark ML and MLlib packages: A comparative study," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 11, pp. 674–677, 2018, doi: 10.14569/ijacsa.2018.091196.
- [14] M. Alhwarat and A. O. Aseeri, "A Superior Arabic Text Categorization Deep Model (SATCDM)," *IEEE Access*, vol. 8, pp. 24653–24661, 2020, doi: 10.1109/ACCESS.2020.2970504.