# HYBRIDIZATION OF DBN WITH SVM AND ITS IMPACT ON PERFORMANCE IN MULTI-DOCUMENT SUMMARIZATION

Karari Kinyanjui, Malanga Ndenga and H.O Nyongesa

Department of Computer Science, Dedan Kimathi University of Technology, Kenya

## ABSTRACT

*Data available from web based sources has grown tremendously with growth of the internet. Users interested in information from such sources often use a search engine to obtain the data which they edit for presentation to their audience. This process can be tedious especially when it involves the generation of a summary. One way to ease the process is by automation of the summary generation process. Efforts by researchers towards automatic summarization have yielded several approaches among them machine learning. Thus, recommendations have been made on combining the algorithms with different strengths, also called hybridization, in order to enhance their performance. Therefore, this research sought to establish the impact of hybridization of Deep Belief Network (DBN) with Support Vector Machine (SVM) on precision, recall, accuracy and F-measure when used in the case of query oriented multi-document summarization. The experiments were carried out using data from National Institute of Standards and Technology (NIST), Document Understanding Conference (DUC) 2006. The data was split into training and test data and used appropriately in DBN, SVM, SVM-DBN hybrid and DBN-SVM hybrid. Results indicated that the hybridized algorithm has better precision, accuracy and F-measure as compared to DBN. Pre-classification hybridization of DBN with SVM (SVM-DBN) gives the best results. This research implies that use of DBN and SVM hybrid algorithms would enhance query oriented multi-document summarization.*

## KEYWORDS

*Automatic summarization, Deep Belief Network, Hybridization, Restricted Boltzmann Machine, Support Vector Machine.*

## 1. INTRODUCTION

The amount of electronic data available has grown exponentially with the growth of the World Wide Web [1]. Hence, exploring this content for potential mining of information is possible but tedious when done manually [2]. Therefore there is a need to automate to save time required to summarize the information available or accessible from the web and other sources.

Research in Natural Language Processing (NLP) and automatic summarization of content have given rise to various approaches to summarization and the application of various algorithms [3]. Hybridization of algorithms is one of the techniques sought after for improvement of performance in the summarization task.

Reference [4] define automatic summarization as the technique of reducing a text document with a system to create a summary that retains the essential information of the original document. Another research done by [5] extends definition of automatic summarization to include the length aspect.

Automatic summarization approaches can be classified as a generic or query-based, extractive, or abstractive, single document or multi-document summarization [6]. Reference [7] noted a more significant focus among the research community on extractive summarization due to the complexity associated with abstractive summarization. Reference [8] point out that multi-document summarization is most likely used where multiple documents occur, such as in web search. In Query-based summarization, the relevance of sentences for generating an extractive summary is based on the presence of terms in the query within a sentence. Hence, sentences with more terms have a higher score than those with fewer terms [9].

Several algorithms have been used in the automatic summary generation, including machine learning algorithms [10]. Reference [11] considers deep learning an advancement of machine learning that provides nonlinear processing in multiple layers of shallow architectures to perform a task. Hence, the use of Deep learning in content extraction and summarization has yielded positive performance reports, especially in text summarization and image processing in computer vision [12].

Similarly, research was done by [12] and [14] propose hybridization as a way of improving the performance of document summarization. Therefore, this research delves into combining DBN with SVM to improve the performance of algorithms in multi-document summarization. The rest of the paper is organized as follows. Section 2 describes the research objectives. Section 3 describes the research questions. Section 4 describes some background and related works. Section 5 describes the source of data and the methodology. Section 6 describes the models of hybridization. Section 7 provides details about the experiment. Section 8 discusses the results and section 9 concludes and gives future direction.

## 2. RESEARCH OBJECTIVE

To improve the performance of algorithms used in automatic summarization, researchers have proposed hybridization, which aims to combine algorithms that solve the same problem either by choosing between one based on data or switching from one to another during execution in order to exploit the desired features of each. This research, therefore, aims to:

a) Establish the impact of hybridization of DBN with SVM on Precision, Recall, Accuracy and F-measure.
b) Determine if hybridization would result in improved performance of automatic multi-document summarization.
c) Determine the better approach to hybridization of DBN with SVM.

## 3. RESEARCH QUESTIONS

To achieve the above-stated objectives, the following research questions will be addressed:

a) How can DBN and SVM algorithms be hybridized?
b) What is the impact of hybridization of DBN and SVM on Precision, Recall, Accuracy, and F-Measure?
c) What is the result of applying hybridization algorithms in multi-document summarization?

# 4. LITERATURE REVIEW

## 4.1. Deep Learning and Multi Document Summarization

Summarization of multiple documents is useful in arrears such as teaching, news media, and preparation of legal documents among others. Several methods can be used in multi document summarization among them machine learning [3].

Deep learning is an advanced form of machine learning which has been used with appreciable performance in image processing and document summarization [15]. It is also the best choice for discovering complex architecture in high-dimensional data by employing back propagation algorithm [11].

There are several approaches to summarization among them query oriented extractive summarization which is particularly suitable for summarization of multiple document arising from web search. Previous research by [16] involving six deep neural network techniques and using DUC 2005-2007 datasets showed that Bi-directional Long Short Term Memory with Maximum Pulling (Bi-LSTM-Max) having the best performance. The performance was determined statistically using Tukey HSD (Honestly Significant Difference) test based on two-way ANOVA (without replication). The remaining five algorithms the were tested are Stacked LSTM (Long Short-Term Memory), CNN(Convolutional Neural Network), Hierarchical CNN(HieConv), Combination of CNN and LSTM (CNN+LSTM), Ensemble neural network (stacked LSTM, Bi-LSTM-Max and Hierarchical (CNN).

Research conducted by [7] showed that in generic summarization, ranking –based MMR performs better than Bi-LSTM. For query oriented multi-document summarization, research by [13] indicated that Query Oriented Deep Extraction (QODE) which is based on Deep Belief Network (DBN) has slightly better performance than Bi-LSTM.

## 4.2. Hybridization of Algorithms

Efforts towards improving the performance of algorithms result to either a modified algorithm, a hybrid algorithm or a novel algorithm [17]. Hybridization of algorithms is done to enhance their performance by combining two or more algorithms with an intention of exploiting the advantages of each. Existing research shows performance improvement after hybridization, such as is the case for hybridization of sine cosine algorithm [18]. Reference [13] proposed integrating a classification algorithm such as Support Vector Machine (SVM) into Query Oriented Deep Extraction (QODE). Similarly, [14] propose combining deep learning and artificial neural network. Additionally, [19] show that for single document summarization, hybridization of algorithms results in better performance than either of the composite algorithms. This research therefore focused on deep learning multi-document summarization techniques, specifically, DBN and the impact of its hybridization with SVM classifier on performance in line with the recommendations.

## 4.3. Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised classification algorithm designed to handle linear and non-linear data [2]. Given a set of data, SVM effectively separates it into two, along a region termed as an isolating hyper plane. According to [20], the content to be summarized was first organized into sentences whose features then form a features matrix. Hence, two steps were

followed to obtain the SVM summary sentences. First, the labeled data was generated and used to train the SVM then the unlabeled data was subjected to the trained SVM for classification.[21].

## 4.4. Deep Belief Network

Deep Belief Network (DBN) is a deep structure made of layers of Restricted Boltzmann Machine (RBM) [13]. Ideally, RBM is made up of two layers, a visible layer and a hidden layer. Hence, all input is in the visible layer and for this reason, in this DBN, the first layer is a visible layer. Consequently, all other hidden layers double up as visible layers when their output becomes the input of the next layer save for the last layer in the sequence [22]. Moreover, RBM does not require labelled data [23]. Therefore, DBN can be unsupervised or semi supervised.

A Deep Belief Network (DBN) such as the one used in this experiment consists of the several layers of RBM and can be represented using algorithm 1.0 as shown below.

---

Algorithm 1.0: Deep Belief Network

---

Input: $TV[\,N_{tv}\,]$   // Test Vector array, each vector is a row of features
Input: $b_i$  // bias, b
Input: $q_i$ // weights qi, i=1..l
Output:  $F$   // Decision function output
Output:  $S$   // summary sentences
//Testing process
**for** $i \leftarrow 1$ **to**  $N_{tv}$ **by** 1 **do**
      $A_i = 0$   //initialize activation function
      $S_i = 0$                //initialize results variable
      $Y_i = F(N_{tvi})$       //predicted value 1 or -1
      $N_E \leftarrow n_e$ // initialize with number of epochs
      **for** $k \leftarrow 1$ **to** $N_e$ **by** 1 **do**
            $A_i = N_{tvi} * q_i + b_i$ //generate the activation function
            $S_i \leftarrow A_i(N_{tvi})$  //determine summary sentences
      **end**
      return $S_i$
**end**

---

## 5. RESEARCH METHOD

In this research, experiments were carried out using web content data from National Institute of Standards and Technology (NIST).   Ideally, the data was preprocessed into text data and organized into sentences. Hence, sentence features were generated from the data. Both training and test data were prepared. Therefore, the detailed approach towards testing and evaluation of results is as outlined in the following sections.

## 5.1. Sources and Organization of Data

Data used in this research was obtained from the United States of America, National Institute of Standards and Technology (NIST), Document Understanding Conference (DUC) (https://duc.nist.gov/data.html). The specific data used is from 2006 document understanding conference.
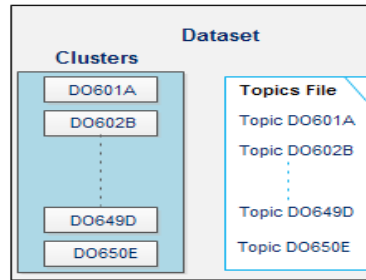
Figure 1: DUC dataset organized into clusters

The selection of this data was based on the fact that it was extracted from reliable websites, based on queries called topics from various sources. In other words, several websites are already in a format that is easy to preprocess for natural language processing and have been used by other researchers for similar or related research [24]. For instance, DUC 2006 dataset is an extract of specific news topics from New York Times (NYT), Associated Press (APW) and Xinhua News (XIN) and is well organized into clusters (Figure 1) with each cluster having files with related data, which is, based on a common topic (Figure 2).
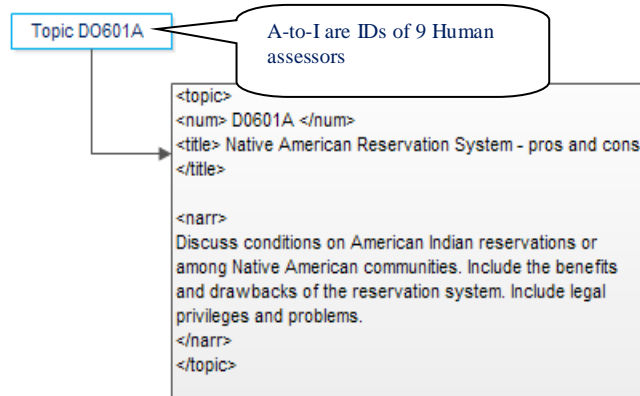


Figure 2: Cluster Topics- Cluster DO601A Topic

The structure of the data across the various clusters allows for application of natural language processing albeit with preprocessing. The data consisted of sgml files with each file in a cluster specifying its source and date the content was prepared in the file name (Figure 3). The topics of the cluster are specified in a separate sgml file.
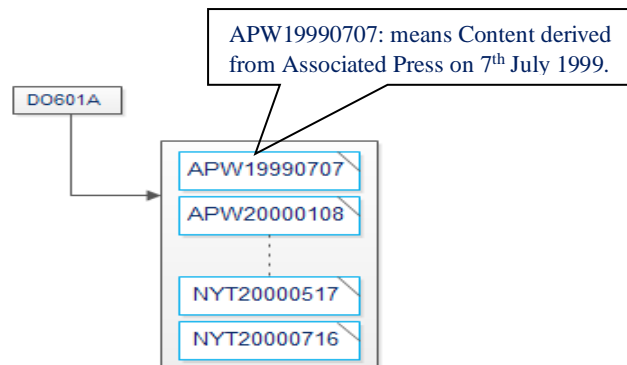


Figure 3: Cluster Files-Cluster DO601A illustrating the file naming scheme

Each cluster in DUC 2006 has about 25 files. Most files have more than 20 sentences. Therefore, a sample was selected form DUC 2006 and split into training and test data.

## 5.2. Sample Size

Reference [25], recommends 500 documents for an appropriate sample size for use in Natural Language processing. Hence, based on this fact, the sample size for this research was set at 525 documents from DUC 2006. The documents were organized into 21 clusters or folders. Documents in each folder had related content and were extracted from the web based on same search query.

The 21 folders sample was then split using the 2/3 rule according to [26]. 14 folders were used for training and 7 folders for testing.

## 5.3. Preprocessing

All data was subject to various stages of preprocessing to lighten the content by applying segmentation, tokenization, stop word filtering, normalization and stemming [27]. Conversion to lowercase was also done as well as removal of special characters and removal of html tags otherwise called content extraction [28].

## 5.4. Development of a user Summary

In order to minimize biasness, four human summarizers were separately assigned the task of generating a human summary. The human summary was then determined based on the average score to the four independent summarizers. Each user summary consisted of selected sentences per file. In each file, the evaluators flagged a sentence with a zero (0) or a one (1) before its serial number to indicate if it forms part of the summary or not based on the topic or query. The number of summarizers was selected to be consistent the practice at NIST DUC 2006.

The DUC 2006 data had 50 clusters and a matching number of topics. A separate topics files provided 50 topics. Each cluster had at most 25 files; in aggregate all files in the 50 clusters had a total of 33,525 sentences. The task for the evaluators was to flag the sentences as either summary or dropped sentences. In total, the evaluators picked 1,050 sentences which is 3% of the population and dropped 32,475 which is 97% of the population.

## 5.5. Feature Determination and Generation

The features matrix was generated based on four features. The four features include term frequency (tf), sentence similarity with the topic (SS), temporal dimension (td) and sentence length penalty (sp). Reference [15] observed that cosine similarity is the most often used method for measuring sentence similarity. Reference [24] supports the assignment of a value against content based on its age. This helps give preference to more recent content and hence avoid some of the challenges associated with multi-document summarization specifically temporal dimension [29]. Reference [13] adds length of summary and achieving good coverage as problems of multi-document summarization hence the importance of sentence length penalty (sp) feature. Additionally, [5] proposes use of more than one feature in sentence scoring which is consistent with research done by [3]. The latter states that a combination of statistical measures improves the quality of the results. The combination of features is linear. In the said research, temporal dimension was calculated by taking the reciprocal of the documents age (DA) [24] as shown in Equation (1) below.

$$TD = \frac{1}{DA} \hspace{3cm} \textbf{(1)}$$

$DA$ (Equation 2) is calculated as the difference in years between the latest possible year $LY$, that is year 2000 and the year the candidate document was published $CY$. The one is added to avoid division by zero error and effectively obtain the temporal dimension [24].

$$DA = (1 + LY) - CY \hspace{3cm} \textbf{(2)}$$

Sentence length penalty $SP$ shown in Equation 3 is a feature designed to reward sentences for having an appropriate size, that is, not too small and not too large hence sentences with a size between 5 and 20 words were rewarded [24].

$$SP = \begin{cases} 0, & 20 < SW < 5 \\ 0.05, & 5 \leq SW \leq 20 \end{cases} \hspace{2cm} \textbf{(3)}$$

Where $SW$ is the number of words in a sentence.

The features matrix (Figure 4) therefore assumed the form:

$$
\begin{array}{c}
\\
\overrightarrow{x_1} \\
\overrightarrow{x_2} \\
\overrightarrow{x_3} \\
. \\
. \\
. \\
\overrightarrow{x_n}
\end{array}
\left(
\begin{array}{cccc}
TF & SS & TD & SP \\
f_1 & f_3 & f_4 & f_5 \\
f_1 & f_3 & f_4 & f_5 \\
f_1 & f_3 & f_4 & f_5 \\
\ldots & & & \\
& & & \\
\ldots & & & \\
f_1 & f_3 & f_4 & f_5
\end{array}
\right)
$$

Figure 4: Features Matrix Depicting Four futures derived from every sentence

Where $\overrightarrow{x_i}$ denotes the $i^{th}$ sentence in a cluster of documents under the same query having a total of $n$ sentences.

The features matrix for the training set was matched with the manual user summary labels (figure 10) to form the training data while the features matrix for test data was used to run the test. The tools used for this research were developed using python and are available in github [30]

## 6. HYBRIDIZATION OF DBN WITH SVM

This research involved establishing the impact of hybridization of DBN with SVM on precision, recall, accuracy and f-measure. The values of the SVM and DBN were determined alongside those of the hybridized algorithms for comparison purposes. Hence, the research then proposed two models of hybridization based on the placement of the classification algorithm (SVM) hence pre-classification and Post –classification.

## 6.1. Pre-Classification Hybridization (SVM-DBN)

In the first model, classification using SVM precedes deep learning using DBN (Algorithm 2.1). DBN is used for fine tuning the summary class resulting from SVM classification. This way, the strength of SVM as a classifier is integrated into the summarization process.

---

Algorithm 2.1: Pre - classification hybridization

Input:     $N_{in}$     // Number of input vectors

Input:     $N_{sv}$     // Number of support vectors

Input:     $N_{ft}$     // Number of features in support a vector

Input:     $SV[\,N_{sv}]$     // support vector array

Input:     $IV[\,N_{in}]$     // input vector array

Input:     $TV[\,N_{tv}]$     // Test Vector array

Input: $b_i$   // bias, b

Input: $q_i$   // weights qi, i=1..l

Output:   $F$   // Decision function output

Output:   $Y$   // Classification results either 1 or -1

//Training process

**for** $i \leftarrow 1$ **to** $N_{in}$ **by** $1$ **do**

     $F = 0$   //Decision function

     **for** $j \leftarrow 1$ **to** $N_{sv}$ **by** $1$ **do**

         $dist = 0$ //distance from Sv

         **for** $k \leftarrow 1$ **to** $N_{ft}$ **by** $1$ **do**

             $dist\mathrel{+}= (SV[j].feature[k] - IV[i].feature[k])^2$

         **end**

         $k = \exp(-y * dist)$

         $F\mathrel{+}= SV[j].a^* \ x \ k$

     **end**

     $F = F + b^*$

**end**

//Testing process

**for** $i \leftarrow 1$ **to** $N_{tv}$ **by** $1$ **do**

     $Y_i = F(N_{tvi})$     //predicted value 1 or -1

     $A_i = 0$   //initialize activation function

     $S_i = 0$         //initialize results variable

     $N_E \leftarrow n_e$   // initialize with number of ephocs

     **if**   $Y_i = 1$ **then**

         **for** $k \leftarrow 1$ **to** $N_e$ **by** $1$ **do**

             $A_i \leftarrow$ //generate the activation function

             $S_i \leftarrow A_i(N_{tvi})$  //obtain results of summary sentences

         **end**

     **end**

return $S_i$

**end**

---

## 6.2. Post-Classification Hybridization (DBN-SVM)

In the second model, feature vectors are received as DBN input and undergo multiple layers refinement (Algorithm 2.2). The refined features vectors are input to SVM which classifies them into either dropped class or summary class. The summary class is the result of interest.

Algorithm 2.2: Post-Classification Hybridization
Input:     $N_{in}$     // Number of input vectors
Input:     $N_{sv}$     // Number of support vectors
Input:     $N_{ft}$     // Number of features in support a vector
Input:     $SV[\,N_{sv}]$     // support vector array
Input:     $IV[\,N_{in}]$     // input vector array
Input:     $TV[\,N_{tv}]$     // Test Vector array
Input: $b_i$ // bias, b
Input: $q_i$ // weights qi, i=1..l
Output:   $F$   // Decision function output
Output:   $S$   // summary sentences


//DBN Feature refinement process
**for** $i \leftarrow 1$ **to** $N_{tv}$ **by** 1 **do**
    $A_i = 0$   //initialize activation function
    $R_i = 0$                //initialize results variable
    $Y_i = F(N_{tvi})$       //predicted value 1 or -1
    $N_E \leftarrow n_e$ // initialize with number of ephocs
    **for** $k \leftarrow 1$ **to** $N_e$ **by** 1 **do**
        $A_i = N_{tvi} * q_i + b_i$ //generate the activation function
        $R_i \leftarrow A_i(N_{tvi})$
        $TV_i \leftarrow ([N_{tvi}][R_i])$ // determine feature matrix for input /test vector array
    **end**
//SVM Training process
**for** $i \leftarrow 1$ **to** $N_{in}$ **by** 1 **do**
    $F = 0$  //Decision function
    **for** $j \leftarrow 1$ **to** $N_{sv}$ **by** 1 **do**
        $dist = 0$//distance from Sv
        **for** $k \leftarrow 1$ **to** $N_{ft}$ **by** 1 **do**
            $dist\mathrel{+}= (SV[j].feature[k] - IV[i].feature[k])^2$
        **end**
        $k = \exp(-y * dist)$
        $F\mathrel{+}= SV[j].a^* \; x \; k$
    **end**
    $F = F + b^*$
**end**
//SVM Summary Generation process
**for** $i \leftarrow 1$ **to** $N_{tv}$ **by** 1 **do**
    **if** $TV_i = 1$ **then**
        $Y_i = F(N_{tvi})$     //predicted value 1 or -1
    **End**
    return   $Y_i$
**end**

## 7. EXPERIMENT AND RESULTS

The metrics used to measure performance of summarization algorithms in the multi-document summarization task are precision, recall, accuracy and f-measure. Reference [31] states that F-measure is the preferred metric to analyze this type of data. The other metric will however be important for various applications. The values used in computing the metrics are derived from the confusion matrix [32] also called matching matrix [33] or error matrix and which is a visually represented as shown in Table 1 below:

Table 1: Confusion Matrix

| | | Predicted | | |
|---|---|---|---|---|
| | | Dropped sentences | Summary sentences | All Sentences |
| Actual | Dropped sentences | Correctly Dropped Sentences | Erroneous Picked Summary sentences | Actual Summary total |
| | Summary sentences | Erroneously Dropped Sentences | Correctly picked summary Sentences | Actual Dropped total |
| | All Sentences | Predicted Dropped total | Predicted Summary total | Overall Number of sentences |

Precision (Equation 4) indicates how often the algorithm is correct when it predicts that a sentence should be included in the summary [34].

$$Precision = \frac{correctly\ picked\ summary\ sentences}{predicted\ summary\ total} \quad (4)$$

Recall (Equation 5) also called positive rate, indicates the probability that a sentence that should be in the summary is predicted to be in included in the summary [33]

$$Recall = \frac{correctly\ picked\ summary\ sentences}{Actual\ summary\ total} \quad (5)$$

Accuracy (Equation 6) shows how often the algorithm is correct, that is, how often does it predict that a sentence is in the summary when it is actually in the summary or that a sentence is dropped and its actually dropped?

$$Accuracy = \frac{correctly\ picked\ summary\ sentences + Coorectly\ dropped\ sentences}{overall\ number\ of\ sentences} \quad (6)$$

F-measure (Equation 7) also called F-score or dice similarity coefficient is a weighted average of recall and precision [33].

$$F - measure = 2 \times \left(\frac{Precision * Recall}{Precision + Recall}\right) \quad (7)$$

## 7.1. Results as Computed from the Confusion Matrix

Table key: TN-True Negative, FP-False Positive, FN-False Negative, TP-True Positive; AY-Actual Yes, PY-Predicted Yes

Table 2: Results derived from the confusion matrix per folder (D0615 to D0621)

| | Confusion Matrix | | | | | | | Measures | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | TN | FP | FN | TP | AY | PY | All | Precision | Recall | Accuracy | F-measure |
| **D0615** | | | | | | | | | | | |
| **SVM** | 1007 | 15 | 75 | 99 | 174 | 114 | 1196 | 0.868 | 0.569 | 0.925 | 0.687 |
| **DBN** | 511 | 511 | 91 | 83 | 174 | 594 | 1196 | 0.14 | 0.477 | 0.497 | 0.216 |
| **SVM-DBN** | 1011 | 11 | 100 | 74 | 174 | 85 | 1196 | 0.871 | 0.425 | 0.907 | 0.571 |
| **DBN-SVM** | 1014 | 8 | 125 | 49 | 174 | 57 | 1196 | 0.86 | 0.282 | 0.889 | 0.425 |
| **D0616** | | | | | | | | | | | |
| **SVM** | 325 | 0 | 6 | 2 | 8 | 2 | 333 | 1 | 0.25 | 0.982 | 0.4 |
| **DBN** | 152 | 173 | 3 | 5 | 8 | 178 | 333 | 0.028 | 0.625 | 0.471 | 0.054 |
| **SVM-DBN** | 325 | 0 | 6 | 2 | 8 | 3 | 333 | 0.667 | 0.25 | 0.982 | 0.364 |
| **DBN-SVM** | 325 | 0 | 7 | 1 | 8 | 1 | 333 | 1 | 0.125 | 0.979 | 0.222 |
| **D0617** | | | | | | | | | | | |
| **SVM** | 443 | 4 | 33 | 22 | 55 | 26 | 502 | 0.846 | 0.4 | 0.926 | 0.543 |
| **DBN** | 214 | 233 | 25 | 30 | 55 | 263 | 502 | 0.114 | 0.545 | 0.486 | 0.189 |
| **SVM-DBN** | 443 | 4 | 38 | 17 | 55 | 21 | 502 | 0.81 | 0.309 | 0.916 | 0.447 |
| **DBN-SVM** | 444 | 3 | 40 | 15 | 55 | 18 | 502 | 0.833 | 0.273 | 0.914 | 0.411 |
| **D0618** | | | | | | | | | | | |
| **SVM** | 298 | 37 | 27 | 54 | 81 | 91 | 416 | 0.593 | 0.667 | 0.846 | 0.628 |
| **DBN** | 155 | 180 | 40 | 41 | 81 | 221 | 416 | 0.186 | 0.506 | 0.471 | 0.272 |
| **SVM-DBN** | 304 | 31 | 38 | 43 | 81 | 74 | 416 | 0.581 | 0.531 | 0.834 | 0.555 |
| **DBN-SVM** | 313 | 22 | 57 | 24 | 81 | 46 | 416 | 0.522 | 0.296 | 0.81 | 0.378 |
| **D0619** | | | | | | | | | | | |
| **SVM** | 557 | 46 | 71 | 82 | 153 | 128 | 756 | 0.641 | 0.536 | 0.845 | 0.584 |
| **DBN** | 291 | 312 | 73 | 80 | 153 | 392 | 756 | 0.204 | 0.523 | 0.491 | 0.294 |
| **SVM-DBN** | 566 | 37 | 85 | 68 | 153 | 105 | 756 | 0.648 | 0.444 | 0.839 | 0.527 |
| **DBN-SVM** | 575 | 28 | 105 | 48 | 153 | 76 | 756 | 0.632 | 0.314 | 0.824 | 0.42 |
| **D0620** | | | | | | | | | | | |
| **SVM** | 498 | 62 | 98 | 155 | 253 | 217 | 813 | 0.714 | 0.613 | 0.803 | 0.66 |
| **DBN** | 283 | 277 | 138 | 115 | 253 | 392 | 813 | 0.293 | 0.455 | 0.49 | 0.356 |
| **SVM-DBN** | 509 | 51 | 136 | 117 | 253 | 168 | 813 | 0.696 | 0.462 | 0.77 | 0.555 |
| **DBN-SVM** | 528 | 32 | 183 | 70 | 253 | 102 | 813 | 0.686 | 0.277 | 0.736 | 0.395 |
| **D0621** | | | | | | | | | | | |
| **SVM** | 213 | 16 | 56 | 72 | 128 | 88 | 357 | 0.818 | 0.563 | 0.798 | 0.667 |
| **DBN** | 120 | 109 | 59 | 69 | 128 | 178 | 357 | 0.388 | 0.539 | 0.529 | 0.451 |
| **SVM-DBN** | 213 | 16 | 72 | 56 | 128 | 72 | 357 | 0.778 | 0.438 | 0.754 | 0.56 |
| **DBN-SVM** | 219 | 10 | 87 | 41 | 128 | 51 | 357 | 0.804 | 0.32 | 0.728 | 0.458 |

For ease of analysis, the average across the dataset were computed and represented per algorithm as shown in the sub sections below.

**7.2 Results per algorithm**

**Support Vector Machine (SVM)**

SVM is subjected to labelled data for training purposes. The training data was used in preparing the labeled data. The trained SVM was then assigned unclassified data from the test data. The results obtained tabulated as follows:

Table 3: SVM-Precision, Recall, Accuracy, F-measure values

| Algorithm | Precision | Recall | Accuracy | F-Measure |
| --- | --- | --- | --- | --- |
| SVM | 0.783 | 0.514 | 0.875 | 0.596 |

**Deep belief Network (DBN)**

DBN is subjected to the test data and the results obtained are as follows:

Table 4: DBN - Precision, Recall, Accuracy, F-measure values

| Algorithm | Precision | Recall | Accuracy | F-Measure |
| --- | --- | --- | --- | --- |
| DBN | 0.193 | 0.524 | 0.491 | 0.262 |

**SVM-DBN hybridization**

The outcome of the SVM consists of data classified into summary and dropped sentences. This data is subjected to the DBN and the results are as shown in table 3 below:

Table 5: SVM-DBN - Precision, Recall, Accuracy, F-measure values

| Algorithm | Precision | Recall | Accuracy | F-Measure |
| --- | --- | --- | --- | --- |
| SVM-DBN | 0.722 | 0.408 | 0.857 | 0.511 |

**DBN-SVM hybridization**

The results obtained from DBN above are subjected to SVM and the results obtained are as shown below:

Table 6: DBN-SVM-Precision, Recall, Accuracy, F-measure values

| Algorithm | Precision | Recall | Accuracy | F-Measure |
| --- | --- | --- | --- | --- |
| DBN-SVM | 0.762 | 0.27 | 0.84 | 0.387 |

## 8. DISCUSSION OF RESULTS

The results shown in the previous section can be combined into one table (table 6) for effective comparison among the various multi- document summarization algorithms.

Table 7: Comparative Analysis

| Algorithm | Measures of performance | | | |
|---|---|---|---|---|
| | Precision | Recall | Accuracy | F-Measure |
| SVM | 0.783 | 0.514 | 0.875 | 0.596 |
| DBN | 0.193 | 0.524 | 0.491 | 0.262 |
| SVM-DBN | 0.722 | 0.408 | 0.857 | 0.511 |
| DBN-SVM | 0.762 | 0.27 | 0.84 | 0.387 |

## 8.1. Precision

The results above indicate that SVM has the highest precision whereas DBN has the lowest precision. Hence, hybridization significantly improves precision with Post-Classification hybridization having better results than pre-classification hybridization.

## 8.2. Recall

DBN has the highest Recall followed closely by SVM. SVM-DBN and DBN-SVM both have significantly lower recall values. Thus, the results indicate that hybridization of DBN and SVM in any form, negatively affects Recall. This research also shows that between pre –classification and post- classification hybridization of DBN, the earlier has better recall values.

## 8.3. Accuracy

SVM has the highest accuracy and DBN has the lowest accuracy. Therefore, hybridization of any from significantly boost the accuracy of DBN with pre-classification hybridization (SVM-DBN) having better results than post-classification hybridization (DBN-SVM).

## 8.4. F-measure

SVM has the highest F-measure and DBN has the lowest F-measure as shown in Table 6. Hence, both Pre and Post classification hybridization improves the f-measure value. Pre –classification hybridization of DBN grants the highest improvement in terms of performance as determined by F-measure.

## 9. CONCLUSION AND FUTURE DIRECTION

The purpose of this research was to test the approaches to hybridization of DBN with SVM and establish the impact of the hybridization on precision, recall, accuracy and f-measure based on a multi-document summarization task. The results revealed that DBN and SVM can be hybridized using two approaches; pre-classification in which SVM precedes DBN and post-classification in which DBN precedes SVM. Result indicate that pre-classification is the better approach to the hybridization of DBN with SVM. Hybridization results to improvement in Precision, Accuracy and F-Measure. Recall is negatively affected. Therefore, the results of this research can be used to enhance multi-document summarization. The practical application of this research would be in summarization of documents obtained from the World Wide Web, the search for appropriate precedence in preparation of defense by lawyers and judgment by judges, social media profiling where this is necessary as part of background checks conducted by employment and security agencies. The results can also be used for automatic curation or aggregation of content from the World Wide Web especially for news where legal issues regarding news aggregation have been dealt with. Further research can be conducted to find out the optimal epochs based on the sample

size in the case of Deep Belief Network and related to the network error. Finally, the DBN used in this research was semi supervised. Research can be done to assess the hybridization of unsupervised DBN.

## ACKNOWLEDGEMENTS

## REFERENCES

[1] Bharti, S. K., Babu, K. S., & Jena, S. K. (2017). Automatic Keyword Extraction for Text Summarization: A Survey.

[2] Suthaharan, S. (2016). Support vector machine. In Machine learning models and algorithms for big data classification (pp. 207-235). Springer, Boston, MA.

[3] Nazari, N., & Mahdavi, M. A. (2019). A survey on Automatic Text Summarization. Journal of AI and Data Mining, 7(1), 121-135

[4] Mosa, M. A., Anwar, A. S., & Hamouda, A. (2019). A survey of multiple types of text summarization with their satellite contents based on swarm intelligence optimization algorithms. Knowledge-Based Systems, 163, 518-532.

[5] Naserasadi, A., Khosravi, H., & Sadeghi, F. (2019). Extractive multi-document summarization based on textual entailment and sentence compression via knapsack problem. Natural Language Engineering, 25(1), 121-146.

[6] Das, D., & Martins, A. F. (2007). A survey on automatic text summarization. Literature Survey for the Language and Statistics II course at CMU, 4(192-195), 57

[7] Gambhir, M., & Gupta, V. (2017). Recent automatic text summarization techniques: a survey. Artificial Intelligence Review, 47(1), 1-66.

[8] Saggion, H., & Poibeau, T. (2013). Automatic text summarization: Past, present and future. In Multi-source, multilingual information extraction and summarization (pp. 3-21). Springer, Berlin, Heidelberg

[9] Abdelaleem, N. M., Kader, H. A., & Salem, R. (2019). A Brief Survey on Text Summarization Techniques. IJ of Electronics and Information Engineering, 10(2), 103-116.

[10] Dhankhar, Y., Bala, I., Singh, S., & Dalal, S. (2018). A Comprehensive Study of Text Summarization Algorithms.

[11] Dargan, S., Kumar, M., Ayyagari, M. R., & Kumar, G. (2019). A Survey of Deep Learning and Its Applications: A New Paradigm to Machine Learning. Archives of Computational Methods in Engineering, 1-22.

[12] Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., ... & Iyengar, S. S. (2018). A survey on deep learning: Algorithms, techniques, and applications. ACM Computing Surveys (CSUR), 51(5), 1-36.

[13] Zhong, S. H., Liu, Y., L i, B., & Long, J. (2015, June). Query-oriented unsupervised multi-document summarization via deep learning model. Expert Systems with Applications, 42(21), 8146-8155.

[14] Kirmani, M., Hakak, N. M., Mohd, M., & Mohd, M. (2019). Hybrid Text Summarization: A Survey. In Soft Computing: Theories and Applications (pp. 63-73). Springer, Singapore

[15] Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., & Kochut, K. (2017). A brief survey of text mining: Classification, clustering and extraction techniques. arXiv preprint arXiv:1707.02919.

[16] Sakai, T. (2019, March). A Comparative Study of Deep Learning Approaches for Query-Focused Extractive Multi-Document Summarization. In 2019 IEEE 2nd International Conference on Information and Computer Technologies (ICICT) (pp. 153-157). IEEE.

[17] Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M., & Gandomi, A. H. (2021). The arithmetic optimization algorithm. Computer methods in applied mechanics and engineering, 376, 113609.

[18] Abualigah, L., & Diabat, A. (2021). Advances in sine cosine algorithm: a comprehensive survey. Artificial Intelligence Review, 1-42.

[19] Mao, X., Yang, H., Huang, S., Liu, Y., & Li, R. (2019). Extractive summarization using supervised and unsupervised learning. Expert Systems with Applications, 133, 173-181

[20] Shen, X., Niu, L., Qi, Z., & Tian, Y. (2017). Support vector machine classifier with truncated pinball loss. Pattern Recognition, 68, 199-210.

[21] Zou, X., Cao, J., Guo, Q., & Wen, T. (2018). A novel network security algorithm based on improved support vector machine from smart city perspective. Computers & Electrical Engineering, 65, 67-7

[22] Alrawashdeh, K., & Purdy, C. (2016, December). Toward an online anomaly intrusion detection system based on deep learning. In 2016 15th IEEE international conference on machine learning and applications (ICMLA) (pp. 195-200). IEEE.

[23] Károly, A. I., Fullér, R., & Galambos, P. (2018). Unsupervised clustering for deep learning: A tutorial survey. Acta Polytechnica Hungarica, 15(8), 29-53.

[24] Qumsiyeh, R., & Ng, Y. K. (2016). Enhancing web search by using query-based clusters and multi-document summaries. Knowledge and Information Systems, 47(2), 355-380.

[25] Juckett, D. (2012). A method for determining the number of documents needed for a gold standard corpus. Journal of biomedical informatics, 45(3), 460-470.

[26] Dobbin, K. K., & Simon, R. M. (2011). Optimally splitting cases for training and testing high dimensional classifiers. BMC medical genomics, 4(1), 1-8.

[27] Patel, D. B., Shah, S., & Chhinkaniwala, H. R. (2019). Fuzzy logic based multi Document Summarization with improved sentence scoring and redundancy removal technique. Expert Systems with Applications

[28] Savitha, R., Arulmurugan, A., & Rajaraman, K. (2020). Online RBM: Growing Restricted Boltzmann Machine on the fly for unsupervised representation. Applied Soft Computing, 106278.

[29] Campos, R., Dias, G., Jorge, A. M., & Jatowt, A. (2015). Survey of temporal information retrieval and related applications. ACM Computing Surveys (CSUR), 47(2), 15.

[30] Karari, E. K., (2020, August). QODEC. Retrieved from https://github.com/Karari-EK/QODEC

[31] Weninger, T., Palacios, R., Crescenzi, V., Gottron, T., & Merialdo, P. (2016). Web content extraction: a metaanalysis of its past and thoughts on its future. ACM SIGKDD Explorations Newsletter, 17(2), 17-23.

[32] Boudia, M. A., Hamou, R. M., Amine, A., & Lokbani, A. C. (2020). An adaptation of a F-measure for automatic text summarization by extraction. Cluster Computing, 1-10.

[33] Vikas, B. O., & Mungara, J. (2016). An Enhanced Extraction and Summarization Technique with User Review Data for Product Recommendation to Customers.

[34] Pattanaik, A., Mishra, S. S., & Das, M. (2020). A Comparative Study of Classifiers for Extractive Text Summarization. In Machine Learning and Information Processing (pp. 173-181). Springer, Singapore.

## AUTHORS

**Karari Kinyanjui** graduated with an MSc. In Information System from the University of Nairobi in 2010. Since then, he has worked in several organizations including Dedan Kimathi University of Technology and County Government of Nyeri. He is currently a PhD student at Dedan Kimathi University, department of Computer Science. His research interests include big data, deep learning and cyber security\

**Malanga Ndenga :** graduated with a PhD from the University of Paris.. He is s lecturer at Dedan Kimathi University of Technology and is currently the Chairman, Information Technology Department. His research interests include big data, machine learning and software quality.

**H.O Nyongesa :** He is a Professor of Computer Science and Systems Engineering and an Academic and Technology Consultant at Dedan Kimathi University of Technology, School of Computer Science and Information Technology. His research interests include big data, machine learning, intelligent systems and Data Analytics.