A New Hybrid Model for Automatic Text Classification

Mahmoud Elmarhoumy, Fuji Ren

Department of Information Science and Intelligent Systems, Faculty of Engineering, University of Tokushima, Japan.

Abstract—This paper introduces three novel pointes the first one is creating the proposed \(tfsc, dfsc\) algorithm to sort the terms in N-gram model which effected goodly on the classification performance. The second one is proposing a new distance similarity method for N-gram model where the new method solves the problem of the difference in representation lengths among classes and documents. The third one is establishing a hybrid Center Profile Vector (CPV) classification model based on the modified N-gram and centroid classifier models. The hybrid (CPV) classification model gain a higher classification accuracy better than N-gram and centroid models as the paper will show in the evaluation result.

Keywords—Hybrid CPV model, Text Classification, Centroid Classifier, N-gram, \((tfsc, dfsc)\), VSM.

1. INTRODUCTION

Text classification is defined as the task of classifying documents into a fixed number of predefined categories. Recently, there have been many documents processed automatically by computer. With the growth of the Internet, the number of the online documents has become huge. Therefore, development of automated text classification has become a serious and important issue in organizing these documents. A number of techniques for text classification have been viewed as supervised learning, ranging from those situations where the target has predefined class labels to test documents where organization is based on similarity deduction from the training set of labeled documents.

Considerable research has investigated text classification, using many approaches, such as nearest neighbor method and Rocchio classifiers model, which classifies full-text news stories and reaches high recall with moderate precision, without requiring manual definitions of the various topics. Other algorithms, such as decision-trees models, focused on text categorization and developing predictive capabilities, e.g., Bayesian probabilistic algorithms. This approach examines inductive learning to classify natural language texts into predecessor content categories, and support vector machines method. It consists of a set of related supervised learning methods that analyze data and arrange patterns for classification. However, all these approaches involve large computations in both learning and testing processing; this is very expensive as most of these algorithms use complicated processes [1-6] [8, 9, 11, 12].

Among these models, a variant of linear models called a centroid-based method is attractive since it has relatively less computation than other methods in both the learning and classification stages. The traditional centroid-based method can be viewed as a specialization of so-called Rocchio method and used in several works on text categorization. Based on the vector space model, a centroid-based method computes beforehand, for each category, an explicit profile (or class prototype), which is a centroid vector for all positive training documents of that category. The classification task is to find the most similar class to the vector of the document we would like to classify. This type of classifiers is easy to implement and effective in computation [10-12].

II. BASELINE METHODS

This section describes the traditional classification models which are centroid classifier model and N-gram model based on word. Moreover, the end of this section will show the applying weighting method.

Centroid Classifier Model

Let’s take a set of classes \(K = \{k_1, k_2, ..., k_m\}\) and the training data set of documents \(D = \{d_1, d_2, ..., d_l\}\) where the training document \(d_i\) should be assigned to one class, by using this given information classifier can find one suitable class for a new document. Vector space model represents documents and classes by vectors based on the \(tfidf\) weight of each term in the document or class. In centroid classifier documents and classes are represented by using (VSM) Vector space model which consider each document and class as a vector in the term-space. However, centroid has three applications to apply it with text classification. The first one is to create the centroid of the class by summing all the terms values in the documents vectors which related to one class based on formula 1, which calls traditional centroid. The second one is to create the centroid of the class by taking the average of summing all terms values in the documents vectors of the class divided by number of the documents in the class based on formula 2, which calls Average centroid. The third one is to create the centroid of the class by taking the normalized value of summing all terms values in the documents vectors of the class based on formula 3, which calls normalized centroid. The three types of centroid classifier are established by using the following equations:

Reference Number: W13-C-0010

132
In N-gram model the similarity will be computed by measuring the distance between the representations of query document profile \( d \) and each class profile \( k \). For each word in a query document profile, we will find its counterpart in the class profile and compute the number of places its location differs. Based on Figure 1 the distance similarity between the query document and the class will be as following (in query profile the word [New] sorted as 1, the same word in the class profile sorted as 6, according to formula 5 the distance will be \( |1-6| = 5 \), but for the word [Bank] in the query profile has not find it’s similar in the class profile because of this the distance will be \( 7 \) the maximum distance \( \{ \text{the total number of the terms inside the class profile} \} \), similarly, word [Book] = \( [3-7] = 4 \), word [Add] = \( 7 \), word [Go] = \( [5-1] = 4 \), distance \( (d,c) = 5+7+4+7+4 = 27 \). Therefore, the query document will be classified for the class profile which has a less difference with the query profile, according to (Manhattan distance) as the following formula:

\[
c_i = \arg \min_{c \in C} \text{dis}(d,c) = \arg \min_{c \in C} |d - c|
\] 

(5)

**Weighting Method**

The most famous weighting algorithm *tfidf* has been applied in our experimentation to establish the term weighting. The *tfidf* will be calculated according to the following formula:

\[
tf_{i,m} = \frac{t_{i,m}}{\sum_t t_{i,m}}
\]

(6)

Where \( t_{i,m} \) is time of occurrences of term \( t_i \) in document \( d_m \), and the denominator is the sum of number of the occurrences of all terms in document \( d_m \).

\[
idf_i = \log \left( \frac{|D|}{|d_i|} \right)
\]

(7)

Where \(|D|\) is the total number of documents in the corpus and \(|d_i|\) is the numbers of documents that contain term \( t_i \).

\[
tfidf_{i,m} = tf_{i,m} \times idf_i
\]

(8)

### III. PROPOSED METHODS

There is useful information for the classification task in the training data set. This information can decide the importance of the term for the classes. To use the powerful of this information you need to determine the information as following:

1. Term frequency in the class.
2. Frequency of documents which contain the term in the class.
3. Term distribution in the class.
4. Distribution of the documents which contain the term in the class.
5. Term frequency in other classes.
6. Frequency of documents which contain the term in other classes.
7. Term distribution in other classes.
8. Distribution of the documents which contain the term in other classes.
9. The number of the classes which contain the term
10. Relationship between (term frequency in a certain class, documents frequency in a certain class) and (term frequency in other classes, documents frequency in other classes).
11. Relationship between (term distribution in the class, documents distribution in the class) and (term distribution in other classes, documents distribution in other classes).

The proposed eight formulas have been established to extract all the needed information from the training data set. The eight formulas calculate all the previous information as follows:

1- Term frequency in the class

\[ TF_{i,k} = \frac{f_{i,k}}{\sum_{m} t_{m,k}} \]  

(9)

Where \( TF_{i,k} \) is frequency of term \( t_i \) in class \( c_k \), \( f_{i,k} \) is the time of occurrence of term \( t_i \) in class \( c_k \) and \( \sum_{m} t_{m,k} \) is summation of occurrence of all terms in class \( c_k \).

The result of the formula 9 has a direct relationship with the importance of the term in the class. This means the term \( t_i \) will be importance for the class \( c_k \) when the term \( t_i \) has a high rank of frequency in the class \( c_k \).

2- Document frequency in the class

\[ DF_{i,k} = \frac{d_{i,k}}{d_k} \]  

(10)

Where \( DF_{i,k} \) is frequency of the documents which contain term \( t_i \) in class \( c_k \), \( d_{i,k} \) is the number of the documents which contain term \( t_i \) in class \( c_k \) and \( d_k \) is the total number of documents in class \( c_k \).

The importance of the term in the class has a direct relationship with the formula 10, because the importance of the term for the class will increase when the frequency of documents which contain the term in the class increased.

3- Term distribution in the class

\[ TS_{i,k} = \frac{f_{i,k}}{\sum_{j} f_{j,c}} \]  

(11)

Where \( TS_{i,k} \) is distribution of term \( t_i \) in class \( c_k \), and \( \sum_{j} f_{j,c} \) is summation of occurrence of term \( t_i \) in all classes \( c \).

The term which has a high distribution value will be an important term for the class. Based on this, the formula 11 has a direct relationship with the importance of the term for the class.

4- Document distribution in the class

\[ DS_{i,k} = \frac{d_{i,k}}{d_{i,c}} \]  

(12)

Where \( DS_{i,k} \) is distribution of the documents which contain term \( t_i \) in class \( c_k \), and \( d_{i,c} \) is total number of the documents in all classes \( c \) which contain term \( t_i \).

Achieving a high distribution number for the documents which contain term \( t_i \) in the class \( c_k \) makes term \( t_i \) an important term for class \( c_k \). Therefore formula 12 has a direct relationship with the importance of the term for the class.

5- Term frequency in other classes

\[ TF_{i,c*} = \frac{\sum_{i} f_{i,c*}}{\sum_{i} \sum_{c*} f_{i,c*}} (nc) \]  

(13)

Where \( TF_{i,c*} \) is frequency of term \( t_i \) in all classes \( c* \) except class \( c_k \), \( f_{i,c*} \) is sum of occurrence of term \( t_i \) in all classes \( c* \) except class \( c_k \), \( \sum_{i} \sum_{c*} f_{i,c*} \) is sum of occurrence of all terms in all classes \( c* \) which contain term \( t_i \) except class \( c_k \) and \( nc \) is number of classes which contain term \( t_i \) except class \( c_k \).

If the frequencies of the term in other classes have a high value, the importance of the term for the class will be decreased. Then formula 13 has an inverse relationship with the importance of the term for the class.

6- Document frequency in other classes

\[ DF_{i,c*} = \frac{d_{i,c*}}{d_{c*}} (nc) \]  

(14)

Where \( DF_{i,c*} \) is frequency of documents which contain term \( t_i \) in all classes \( c* \) except class \( c_k \), \( d_{i,c*} \) is the number of documents which contain term \( t_i \) in all classes \( c* \) except class \( c_k \), and \( d_{c*} \) is total number of the documents in all classes \( c* \) except class \( c_k \).

The importance of the term for the class will be decreased if the frequencies of the documents which contain the term in other classes have a high value. According to this, the formula 14 has an inverse relationship with the importance of the term for the class.

7- Term distribution in other classes.

\[ TS_{i,c*} = \frac{\sum_{i} f_{i,c*}}{\sum_{i} \sum_{c*} f_{i,c*}} (nc) \]  

(15)

Where \( TS_{i,c*} \) is distribution of term \( t_i \) in all classes \( c* \) except class \( c_k \), \( f_{i,c*} \) is sum of occurrence of term \( t_i \) in all classes \( c* \) except class \( c_k \), and \( \sum_{i} \sum_{c*} f_{i,c*} \) is sum of occurrence of all terms in all classes \( c* \) except class \( c_k \).
8- Document distribution in other classes.

\[ DS_{i,c} = \frac{d_{i,c}}{d_{i,c}(nc)} \quad (16) \]

Where \( DS_{i,c} \) is distribution of documents which contain term \( t_i \) in all classes \( c \) except class \( c_c \), and \( d_{i,c} \) is total number of the documents which contain term \( t_i \) in all classes \( c \). as same as the formula 14, the formula 16 has an inverse relationship with the importance of the term for the class.

The previous values decide the important terms for each class, first let’s find the relationship between the term priority (important) and each value, it will be clear that the first four values have positive relationship (direct correlation) with the term priority (important). The importance of the term for a certain class will increase when (the frequency of the term in the class increase, the number of the documents which contain the term in the class increase, the distribution of the term in a certain class more than other classes and the distribution of the documents in a certain class more than other classes) and vice versa.

As for the last four values, it has inverse relationship with the term priority (important). The importance of the term for a certain class will increase when (the frequency of the term in other classes decrease, the number of the documents which contain the term in other classes decrease, the distribution of the term in other classes less than the term distribution in a certain class and the distribution of documents in other classes less than the documents distribution in a certain class) and vice versa.

Based on the previous relation, we will merge all the previous formulas in one equation to get only one value, the proposed merging equation will be as follows:

\[ \text{tf}sc,\text{df}sc_{i,k} = \frac{TF_{i,k} + DF_{i,k} + TS_{i,k} + DS_{i,k}}{TF_{i,c} + DF_{i,c} + TS_{i,c} + DS_{i,c}} \quad (17) \]

Where \( \text{tf}sc,\text{df}sc_{i,k} \) is frequency and distribution of term \( t_i \) in the class \( c_c \), frequency and distribution of the documents which contain term \( t_i \) in the class \( c_c \).

Therefore using the previous value will be affected to sort the terms in the classes based on its importance for the class. The experimentation will show the effectiveness of the new value in the classification performance.

Enhancing N-gram model based word

Modified N-gram model uses the result of the proposed merging equation 17 to sort the words or terms inside the class profile. Similarly the query documents will be sorted according to the highest \( \text{tf}sc,\text{df}sc \) score.

In the proposed modified model, each class has a different \( \text{tf}sc,\text{df}sc \) value. Therefore the test documents or query documents can not represent using only one value from the \( \text{tf}sc,\text{df}sc \) classes values, because if the test documents have been represented using only one \( \text{tf}sc,\text{df}sc \) class value, so the class label will be decided for the test documents before starting the similarity measure. Therefore, the test documents should be represented using all the \( \text{tf}sc,\text{df}sc \) classes values.

As mentioned before the distance similarity measure between each word in the query document profile and its counterpart in the class profile is calculated based on the Manhattan distance in N-gram model.

A new distance similarity measure is a novel algorithm has been applied for the modified N-gram model to calculate the distance similarity. However, the new distance similarity measure will calculate the distance between the query document profile and the class profile, as follows:

\[ \text{distance}(d,c) = \sum_i \log (c/d) \quad (18) \]

Where \( d \) is the sort number of the term \( t_i \) in the query document profile \( d_i \), and \( c_i \) is the sort number of the term \( t_i \) in the class profile \( c_i \).

The new distance measure algorithm solves the problem of the difference in profiles lengths among the classes and between the document and the class. The new distance algorithm helps the model to classify the documents correctly, because in the experimentation shows the longest class profile contain 4451 words, and the shortest class contain 2714 words, that means there are high difference among classes. In this case the Manhattan distance will calculate the high length difference as (4451-2714=1737), which has a bad effects on the classification task. But in the new distance algorithm the high length difference will be (log 4451-log 2714 = 0.215). Therefore, the new distance measure effect goodly on the performance of the modified N-gram. Taking the modulus of \( \log (c/d) \) make the distance value started from 0 to 3.7, which solves the problem of the distinction between the query document profile length and the class profile length.

\[ c_i = \arg \min_{c \in c} \text{distance}(d,c) \quad (19) \]

The Hybrid Model Based Center Profile Vector

The new proposed Center Profile Vector (CPV) model will represent the query documents and classes as a profile vector. However, the modified N-gram and Centroid classifier models have been used as baseline method to establish the proposed CPV model. Any classification model has a training step and testing step. The training step and testing step of the CPV model will be as follows:

The CPV Model Training Step

Firstly, the center vector of the class will be calculated as same as the class prototype vector in the centroid model using formula (1, 2 or 3) based on the \( \text{tf}df \). After that the CPV model will present each class by the profile vector. This profile vector contains three values. The first value is the center vector of the class using formula (1, 2 or 3) based on the \( \text{tf}df \). The second value is the word it self. The third value is the sorting of the words ordered by the proposed \( \text{tf}sc,\text{df}sc \) value according to formula 17. Now the center profile vector will present the class by the previous three values, (the word
The CPV Model Testing Step

As same as the class the test document will be represented by a profile contains three vector values, word it self, tfidf of word and number of term sorting using tfsc,dfsc. In the new model the similarity measure will be calculated as follows:

\[ c_{a} = \arg \max_{c_{a} \in \mathcal{C}} \sum_{t \in \mathcal{C}} \frac{d_{c_{a}}}{d_{c_{a}}} \cdot \log \left( \frac{d_{c_{a}}}{d_{c_{a}}} \right) + 0.0001 \cdot \| \mathcal{N} \| \cdot \left( \frac{d_{c_{a}}}{d_{c_{a}}} \right) \] (20)

Where \( d_{c_{a}} \) is the tfidf of the term \( t \) in the test document \( d_{c_{a}} \), \( c_{a} \) is the tfidf of the term \( t \) in class \( c_{a} \), \( d_{c_{a}} \) is the order sorting of the term \( t \) in class \( c_{a} \), and \( d_{c_{a}} \) is the number of terms in the test document \( d_{c_{a}} \) which have matching with the class \( c_{a} \), and \( d_{c_{a}} \) is the total number of terms in the test document \( d_{c_{a}} \).

By using formula 20 the propose CPV model will gain the benefits of the centroid model and the modified N-gram model to achieve a higher classification accuracy as we will show in the result.

IV. EVALUATION AND EXPERIMENTATIONS

This paper uses the 20 Newsgroups collection as a data set for the experiments. This corpus contains 18828 documents in 20 different categories which is available in the next home page http://qwone.com/~jason/20Newsgroups/.

The half of this corpus has been used for training and other half for testing. The removing stop words and stemming had been done for all the documents in the corpus. In this paper the F measure has been calculated using the precision and recall according to the following formula:

\[ F - measure = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \] (21)

<table>
<thead>
<tr>
<th>N-gram</th>
<th>DN-g</th>
<th>TC</th>
<th>AC</th>
<th>NC</th>
</tr>
</thead>
<tbody>
<tr>
<td>07.93%</td>
<td>06.13%</td>
<td>75%</td>
<td>76.3%</td>
<td>76.8%</td>
</tr>
<tr>
<td>SN-g</td>
<td>MN-g</td>
<td>CPV&amp;T</td>
<td>CPV&amp;A</td>
<td>CPV&amp;N</td>
</tr>
<tr>
<td>45%</td>
<td>71.88%</td>
<td>80.95%</td>
<td>81.02%</td>
<td>81.3%</td>
</tr>
</tbody>
</table>

Table 1 shows us the classification accuracy using (N-gram) = traditional N-gram model with tfidf and Manhattan distance, (DN-g) = N-gram model with tfidf and new distance measure, (SN-g) = N-gram model with tfsc,dfsc and Manhattan distance, (MN-g) = modified N-gram with tfsc,dfsc and new distance measure, (TC) = traditional centroid classifier, (AC) = average centroid classifier, (NC) = normalized centroid classifier, (CPV&T) = CPV model using class prototype vector of the traditional centroid classifier, (CPV&A) = CPV model using class prototype vector of the average centroid classifier, and (CPV&N) = CPV model using class prototype vector of the normalized centroid classifier. Clear from the result of the N-gram model in table 1 and table 2 that the N-gram get 7.93% where all query documents have been classified only to five (3,4,5,6 and 7) classes which are the smallest class profiles in the length. In another side DN-g gain 6.13 % where the query documents have been classified to fourteen classes. This means the applying of a new distance measure solves the problem of the difference lengths among classes. The SN-g collects 46% this accuracy explains to us that, the applying of a new sorting method tfsc,dfsc enhances the performance of the N-gram model. The MN-g uses the word as a term without slicing and gain a good classification accuracy which enhance the total classification accuracy of the N-gram by 63.95%, DN-g by 65.75%, and SN-g by 25.88%, so using MN-g makes N-gram model able to enter the computations with other models.

From table 1 and table 2 we find that, Using CPV&T enhance the total classification accuracy over the TC by 5.95%. This means the CPV&T is working powerful with the classification task, because it enhances the F-measure for each class and the total accuracy. Using CPV&A improve the total classification accuracy over the AC by 4.72%. This means the CPV&A has a goodly effect on the classification task, because it improves the F-measure accuracy for each class except class 10 and the total accuracy has been improved. The CPV&N gains higher classification accuracy over all models, for NC it is higher by 4.5%, and for SN-g it is higher by 9.42%. This model achieves high classification accuracy because it gains the advantage of the two classifier models which are the centroid classifier model and the modified N-gram model.

V. CONCLUSION AND FUTURE WORK

This paper has investigated different approaches for automatic text classification. Firstly, we have exploited the centroid classifiers and N-gram classifier as a baseline models. Secondly, we proposed the modified N-gram model using the proposed tfsc,dfsc value to sort the profiles in N-gram and the proposed similarity distance measure. Finally we established the hybrid CPV model according to centroid classifiers and the modified N-gram to improve the classification system performance. As the result has shown, the hybrid CPV classification model enhanced all the classes except class 10. This means the hybrid classification model has a good effectiveness on the performance of the classification task. The proposed tfsc,dfsc value and the proposed similarity distance measure have a good effectiveness in the classification performance of the N-gram model.

In the future work, we will apply the CPV classification model on a large English corpus and other languages such as Arabic, French. Furthermore, we could focus on exploiting other classification models like (Support Vector Machine, Decision Trees, and N Bayes,) to compare it with the propose CPV model, to improve the automatic text classification.
### Table 2: Presentation of F-measure for Each Class Using the Traditional and Proposed Models

<table>
<thead>
<tr>
<th>N-gram</th>
<th>DN-g</th>
<th>SN-g</th>
<th>MN-g</th>
<th>TC</th>
<th>CPV&amp;T</th>
<th>AC</th>
<th>CPV&amp;A</th>
<th>NC</th>
<th>CPV&amp;N</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>0%</td>
<td>0.99%</td>
<td>65.15%</td>
<td>75.97%</td>
<td>60.95%</td>
<td>75.74%</td>
<td>66.02%</td>
<td>76.17%</td>
<td>66.24%</td>
</tr>
<tr>
<td>C2</td>
<td>0%</td>
<td>6.99%</td>
<td>33.89%</td>
<td>58.87%</td>
<td>68.76%</td>
<td>74.92%</td>
<td>69.11%</td>
<td>75%</td>
<td>70.57%</td>
</tr>
<tr>
<td>C3</td>
<td>11.72%</td>
<td>9.34%</td>
<td>25.99%</td>
<td>58.03%</td>
<td>60.74%</td>
<td>66.25%</td>
<td>60.79%</td>
<td>66.25%</td>
<td>65.31%</td>
</tr>
<tr>
<td>C4</td>
<td>14.23%</td>
<td>9.75%</td>
<td>47.16%</td>
<td>55.93%</td>
<td>60.91%</td>
<td>66.60%</td>
<td>60.91%</td>
<td>65.79%</td>
<td>63.02%</td>
</tr>
<tr>
<td>C5</td>
<td>17.75%</td>
<td>8.82%</td>
<td>29.36%</td>
<td>50.29%</td>
<td>71.14%</td>
<td>77.90%</td>
<td>71.00%</td>
<td>77.16%</td>
<td>71.71%</td>
</tr>
<tr>
<td>C6</td>
<td>3.35%</td>
<td>6.96%</td>
<td>62.05%</td>
<td>68.90%</td>
<td>62.32%</td>
<td>72.94%</td>
<td>62.23%</td>
<td>72.86%</td>
<td>68.84%</td>
</tr>
<tr>
<td>C7</td>
<td>0.40%</td>
<td>3.47%</td>
<td>79.63%</td>
<td>88.11%</td>
<td>87.48%</td>
<td>92.01%</td>
<td>88.08%</td>
<td>91.98%</td>
<td>88.73%</td>
</tr>
<tr>
<td>C8</td>
<td>0%</td>
<td>0%</td>
<td>73.51%</td>
<td>85.57%</td>
<td>92.85%</td>
<td>93.73%</td>
<td>93.18%</td>
<td>94.10%</td>
<td>93.16%</td>
</tr>
<tr>
<td>C9</td>
<td>0%</td>
<td>2.08%</td>
<td>63.66%</td>
<td>82.64%</td>
<td>85.48%</td>
<td>89.34%</td>
<td>85.51%</td>
<td>89.81%</td>
<td>85.56%</td>
</tr>
<tr>
<td>C10</td>
<td>0%</td>
<td>17.79%</td>
<td>49.46%</td>
<td>61.19%</td>
<td>72.93%</td>
<td>73.53%</td>
<td>73.24%</td>
<td>72.93%</td>
<td>74.71%</td>
</tr>
<tr>
<td>C11</td>
<td>0%</td>
<td>1.75%</td>
<td>49.79%</td>
<td>56.91%</td>
<td>20.84%</td>
<td>51.5%</td>
<td>50.90%</td>
<td>59.31%</td>
<td>51.91%</td>
</tr>
<tr>
<td>C12</td>
<td>0%</td>
<td>0%</td>
<td>19.33%</td>
<td>71.51%</td>
<td>65.18%</td>
<td>72.5%</td>
<td>68.75%</td>
<td>72.19%</td>
<td>60.23%</td>
</tr>
<tr>
<td>C13</td>
<td>0%</td>
<td>0%</td>
<td>47.55%</td>
<td>91.28%</td>
<td>90.28%</td>
<td>92.43%</td>
<td>89.27%</td>
<td>91.16%</td>
<td>86.74%</td>
</tr>
<tr>
<td>C14</td>
<td>0%</td>
<td>0%</td>
<td>61.28%</td>
<td>82.82%</td>
<td>78.10%</td>
<td>82.51%</td>
<td>79.11%</td>
<td>82.97%</td>
<td>79.10%</td>
</tr>
<tr>
<td>C15</td>
<td>0%</td>
<td>0%</td>
<td>52.00%</td>
<td>81.54%</td>
<td>69.89%</td>
<td>82.23%</td>
<td>78.70%</td>
<td>83.65%</td>
<td>78.62%</td>
</tr>
<tr>
<td>C16</td>
<td>0%</td>
<td>0%</td>
<td>26.80%</td>
<td>83.3%</td>
<td>86.82%</td>
<td>88.97%</td>
<td>87.00%</td>
<td>89.14%</td>
<td>87.02%</td>
</tr>
<tr>
<td>C17</td>
<td>0%</td>
<td>0.4%</td>
<td>35.17%</td>
<td>77.59%</td>
<td>85.80%</td>
<td>86.57%</td>
<td>86.67%</td>
<td>87.26%</td>
<td>86.89%</td>
</tr>
<tr>
<td>C18</td>
<td>0%</td>
<td>5.15%</td>
<td>43.92%</td>
<td>49.5%</td>
<td>65.24%</td>
<td>73.81%</td>
<td>64.93%</td>
<td>73.31%</td>
<td>67.91%</td>
</tr>
<tr>
<td>C19</td>
<td>0%</td>
<td>0.39%</td>
<td>40.69%</td>
<td>85.05%</td>
<td>86.33%</td>
<td>90.64%</td>
<td>87.08%</td>
<td>90.96%</td>
<td>86.33%</td>
</tr>
<tr>
<td>C20</td>
<td>0%</td>
<td>2.47%</td>
<td>81.03%</td>
<td>90.30%</td>
<td>88.78%</td>
<td>94.32%</td>
<td>89.20%</td>
<td>94.69%</td>
<td>90.49%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>0.79%</strong></td>
<td><strong>0.61%</strong></td>
<td><strong>46%</strong></td>
<td><strong>71.88%</strong></td>
<td><strong>75%</strong></td>
<td><strong>80.95%</strong></td>
<td><strong>76.3%</strong></td>
<td><strong>81.02%</strong></td>
<td><strong>76.8%</strong></td>
</tr>
</tbody>
</table>

### REFERENCES


Reference Number: W13-C-0010