# PMCN: Combining PDF-modified Similarity and Complex Network in Multi-document Summarization

Yi-Ning Tu*, Wei-Tse Hsu**

ARTICLE INFO

*Article history:*
Received 14 October 2018
Revised 28 August 2019
Accepted 01 September 2019

*Keywords:*
TF*PDF,
Multi-document
Summarization,
Complex Network

ABSTRACT

This study combines the concept of degree centrality in complex network with the Term Frequency * Proportional Document Frequency (TF*PDF) algorithm; the combined method, called PMCN (PDF-Modified similarity and Complex Network), constructs relationship networks among sentences for writing news summaries. The PMCN method is a multi-document summarization extension of the ideas of Bun and Ishizuka (2002), who first published the TF*PDF algorithm for detecting *hot* topics. In their TF*PDF algorithm, Bun and Ishizuka defined the publisher of a news item as its *channel*. If the PDF weight of a term is higher than the weights of other terms, then the term is hotter than the other terms. However, this study attempts to develop summaries for news items. Because the TF*PDF algorithm summarizes daily news, PMCN replaces the concept of “channel” with “the date of the news event”, and uses the resulting chronicle ordering for a multi-document summarization algorithm, of which the F-measure scores were 0.042 and 0.051 higher than LexRank for the famous d30001t and d30003t tasks, respectively.

## 1. Introduction

A constantly growing amount of daily news is available online. Readers must quickly find the trajectories of relevant news stories. Multi-document summarization can automatically help readers to filter some redundant pieces of information and provide a coherent and logical story. To generate accurate and complete (but not redundant) multi-document summaries is difficult. Each article is written from a different point of view and usually involves several minor events related to a large topic. Furthermore, the minor events may overlap with each other, or the story may be so specific that the story is focused on itself. These challenges complicate the task of judging whether the information of summaries is useful.

* Associate Professor, School of Fu Jen Catholic University, Department of Statistics and Information Science, (R.O.C.) Taiwan (eniddu@gmail.com) (082435@mail.fju.edu.tw) (First & Corresponding Author)
** Master's student, National Taipei University, Taiwan (victor.playtion@gmail.com) (Co-Author)
International Journal of Knowledge Content Development & Technology, 9(3): 23-41, 2019.
http://dx.doi.org/10.5865/IJKCT.2019.9.3.023

This study combines the Term Frequency * Proportional Document Frequency (TF*PDF) algorithm which first introduced by Bun and Ishizuka (2002), and with the concept of degree centrality from complex network (CN) to produce PMCN (PDF-Modified similarity and Complex Network), which constructs relationship networks between sentences to summarize news. The proposed PMCN multi-document summarization algorithm extends the work of Bun and Ishizuka (2002), who first published the TF*PDF algorithm for detecting *hot* topics. In the TF*PDF algorithm, Bun and Ishizuka defined the publisher of a news story as that story's *channel*. When the PDF weight of a term is higher than the weights of other terms, then the term is hotter than the other terms.

## 2. Literature Review

Multi-document summarization was introduced in the work of Topic Detection and Tracking (TDT) by Allan et al. (2003) and related works such as Yang, Pierce, and Carbonell (1998), Carbonell et al. (1999), and Yang et al. (1999). One of the three major tasks of TDT is to take a small number of sample news stories about an event and to find all relevant stories that follow in the stream data; this is also called event tracking. Some notable works have discussed this objective but have still treated each article as one single event. The original TDT research ended in 2004 and the trend of Automatic Content Extraction (ACE) replaced it. ACE aims at detecting and recognizing events in text. Walker et al. (2006) published some fundamental steps and definitions for later researchers.

Some studies have tried to exploit event-based multi-document summarization. Daniel, Radev, and Allison (2003) built six different methods to summarize a given corpus and found that a manual method may have preferable results. Because a manual method may bias the results, later works focused on automatic multi-document summarization (Marujo et al., 2014).

Automatic document summarization is a paramount issue. The ROUGE tool was proposed by Lin (2004); it was the first index to measure the quality of a summary by comparing it to ideal summaries created by humans. ROUGE counts the number of overlapping units between computer-generated summaries and ideal summaries. Subsequently, Mihalcea and Tarau (2004) suggested a tool, namely TextRank, that adapts the algorithm of PageRank to text extraction; it establishes the summary of a text through the PageRank of each sentence. TextRank can also extract the key words of a document.

In 2009, Antiqueira et al. (2009) proposed complex networks employing network concepts, such as node degree, length of shortest paths, d-rings, and k-cores for document summarization; they created 14 summarizers. In 2011, Ouyang et al. (2011) applied regression models to query-focused multi-document summarization. Their work used support vector regression to estimate the importance of a sentence for establishing a multi-document summarization.

In the same time, the opinion extraction and summarization from the large amount of evaluative text are also flourishing. In this stage, the multi-documentation summarization can be divided into two directions. One focus on the facts, the other is focus on the evaluative text summarization. For example, Hu and Liu (2004) mined and summarized customer reviews. Wilson, Wiebe, and

Hoffmann (2005) recognized contextual polarity in phrase-level sentiment analysis. Popescu and Etzoni (2007) extracted the product features and opinions from reviews.

Different from the factual text, which focus on selecting most important events, preventing the duplication, and present the contents in reasonable sequences and make readers understand the story. However, this study attempts to develop summaries for news events. The TF*PDF algorithm summarizes all news published on a particular day. In this study, the concept of channel is replaced by *the date of the news event* in the PMCN algorithm to construct chronicle ordering information for multiday news summarization.

## 3. Theoretical Background

This study attempts to establish a PMCN (PDF-Modified similarity and Complex Network) multi-document summarization algorithm. Section 3.1 explains the original TF*PDF multi-document summarization algorithm. Section 3.2 demonstrates the process of the PMCN multi-document summarization algorithm. Section 3.3 presents the methods of data preprocessing and construction of a term-sentence matrix. Section 3.4 defines the PDF-modified cosine similarity and the sentence similarity matrix. Section 3.5 illustrates the transformation of a sentence similarity matrix into a sentence relationship matrix. Section 3.6 explains the construction of a text summary.

### *3.1 Concept of original TF*PDF Multi-document Summarization*

Bun and Ishizuka (2002) first published the concept of TF*PDF algorithm for detecting hot topics. In TF*PDF algorithm, they defined the publisher of a news item as a channel. If the PDF weight of a term is higher than the weights of other terms, then the term is hotter than the other terms.

Fig. 1 illustrates the features of the TF*PDF algorithm. Each document in Fig. 1 has 50 terms, and the term “cat” can be considered as an example of a word with the feature of PDF weight. In Fig. 1(a), the term “cat” appears in several documents, and the PDF weight is 1.087. The highest PDF weight in all cases indicates that the term is relatively a hot term. The term “cat” shows up twice in both Fig. 1(b) and Fig. 1(c). Note that the PDF weight of “cat” is 0.544 in Fig. 1(b), but the PDF weight is 0.330 in Fig. 1(c). The PDF weight is higher when instances of the term are gathered in a channel (as in Fig. 1(b)) than it is when the instances are scattered over multiple channels (as in Fig. 1(c)). The PDF weight of “cat” in Fig. 1(d) is 0.330. This means that if a term only appears in a single document, its PDF weight is lower than separate instances of the term appear in multiple documents.

Bun and Ishizuka applied the concept of TF*PDF for the automatic identification of hot topics; however, in this study, an algorithm automatically applies the concept of TF*PDF to produce summaries of news events. In this study, the day of the news event is a more noteworthy factor than the publisher of a document; therefore, in this study, “day” (and not “news publisher”) is used as the channel.

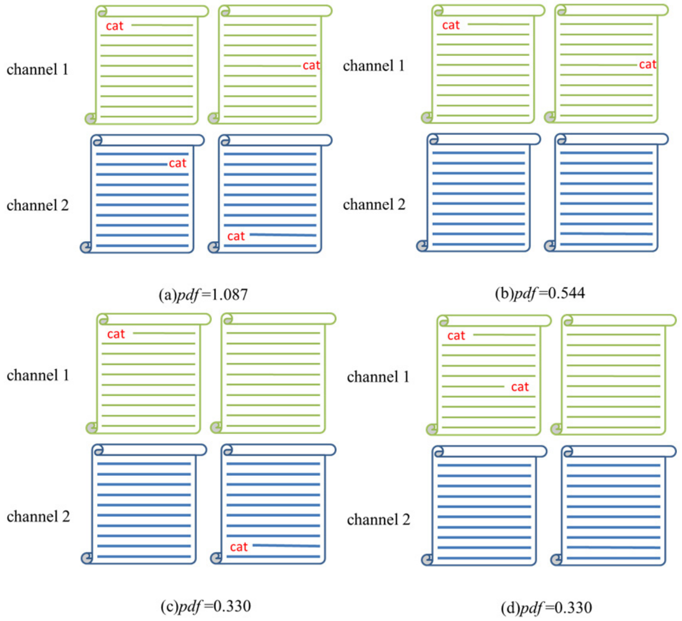

**Fig. 1.** Features of the TF*PDF algorithm

After calculating the PDF weight of each term, the proposed algorithm constructs the semantic relationship between each pair of sentences. If the PDF-modified cosine similarity between two sentences is higher than a threshold, which is selected by users, the algorithm considers these two sentences to be related. The relationships between sentences compose a semantic network, as illustrated in Fig. 2. This work considers the priority of a sentence to be proportional to the number of relationships in which it is involved. In Fig. 2, d1-s4 has the highest number of relationships with other sentences. As a result, this work considers d1-s4 as the paramount sentence in the summary. On the contrary, this work considers d1-s2 and d3-s1 as less vital, because they have no relationships with most of the other sentences.

For inspecting and quantizing the importance of a sentence, this work adopts the concept of degree centrality from complex network.

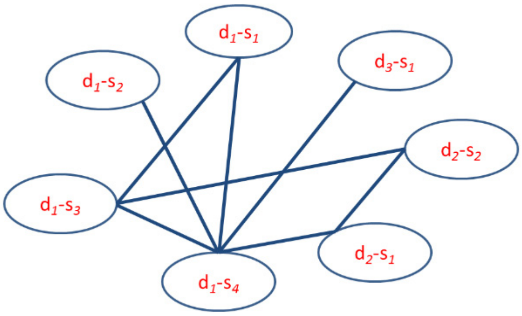

**Fig. 2.** Relationship network among sentences

### *3.2 The Process of PMCN Multi-document Summarization*

Fig. 3 illustrates the process of PMCN multi-document summarization algorithms. This study divides the algorithm into four parts: preprocess data, establish a similarity matrix, establish a relationship matrix, and output the summary. Each stage of the algorithm has a distinct target.

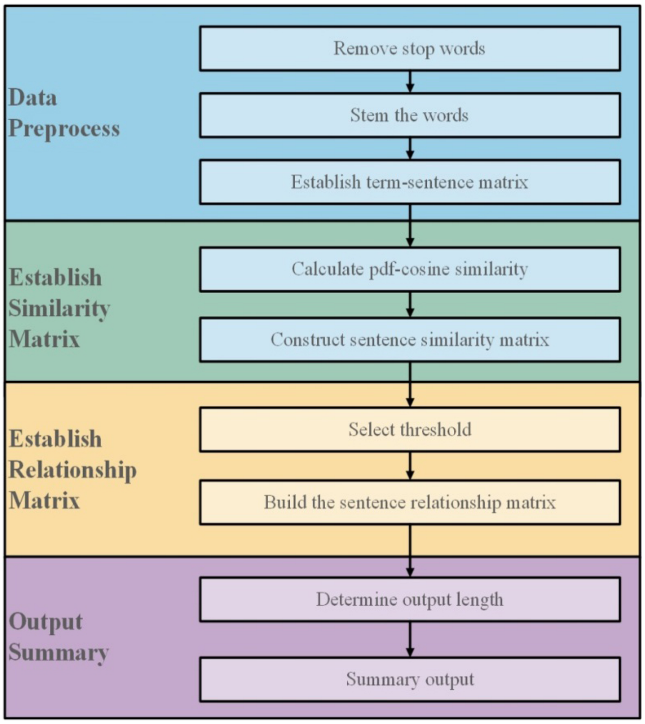
Data Preprocess
Remove stop words
Stem the words
Establish term-sentence matrix
Establish Similarity Matrix
Calculate pdf-cosine similarity
Construct sentence similarity matrix
Establish Relationship Matrix
Select threshold
Build the sentence relationship matrix
Output Summary
Determine output length
Summary output

**Fig. 3.** Phases of PMCN multi-document summarization algorithm

#### *3.2.1 Stage 1. Preprocess data*

Extract knowledge from unstructured text data and transfer the data into vector or matrix form to facilitate analysis in later phases. This phase has three steps.

Data preprocessing is defined as a series of processes on raw text data for multi-document summarizing. The preprocessed data becomes more structured and more analyzable. In this study, data preprocessing steps include removing stop words, stemming the words, and constructing a term-sentence matrix.

- Step 1. Remove stop words: Luhn (1960) first indicated that stop words are commonly used words that have little semantic value. For example, the English words “the”, “and”, “do”, and “since” are typical stop words. Stop words are removed from the sentences to prevent the algorithm from regarding them as crucial terms.

- Step 2. Stem the words: Lovins (1968) demonstrated that stemming is a process of transforming a word to its word stem, its base word form. For example, stemming transforms “wait”, “waits”, and “waiting” to the same base word form, “wait.” In this step, all the letters are put into lowercase forms. Because words sharing the same stem often have the same semantic meaning, this study uses these words.
- Step 3. Establish a term-sentence matrix: The term-sentence matrix shows the frequency with which each term appears in a sentence. This ensures the data is structured and facilities the adoption of degree centrality.

### *3.2.2 Stage 2. Establish a similarity matrix*

For investigation of the similarity between sentences, the measurement of similarity is crucial. The similarity is derived for every pair of sentences.

- Step 1. Calculate PDF-modified cosine similarity. Cosine similarity is a widely used measurement of similarity and is shown as follows.

$$\text{cosine similarity} = \frac{\vec{a}\cdot\vec{b}}{\|\vec{a}\|\|\vec{b}\|} \tag{1}$$

where,

- $\vec{a}$ denotes the term vector of sentence *a*
- $\vec{b}$ denotes the term vector of the sentence *b* for comparison
- $\|\vec{a}\|$ stands for the norm of vector $\vec{a}$
- $\|\vec{b}\|$ stands for the norm of vector $\vec{b}$

Erkan and Radev (2004) combined the concepts of term frequency–inverse document frequency (TF–IDF) and cosine similarity and used the IDF score to weight the terms for measuring the similarities among sentences.

$$\text{IDF-modified-cos}(s_i,s_j) = \frac{\sum_{w\in x,y} tf_{w,x}\times tf_{w,y}\times(\mathrm{i}df_w)^2}{\sqrt{\sum_{x_i\in x}(tf_{x_i,x}\times \mathrm{i}df_{x_i})^2}\times\sqrt{\sum_{y_i\in y}(tf_{y_i y}\times \mathrm{i}df_{y_i})^2}} \tag{2}$$

where,

- $tf_{w,x}$ stands for the occurrence of term *w* in sentence *x*
- $tf_{w,y}$ stands for the occurrence of term *w* in sentence *y*
- $idf_w$ means the IDF weight of term *w*
- $x_i$ indicates the *i*th word in sentence *x*
- $y_i$ indicates the *i*th word in sentence *y*
- *x* denotes the *x*th sentence
- *y* denotes the *y*th sentence

The TF-IDF tends to give frequently appearing terms relatively little weight. However, in this study, the more frequently a term occurs, the more vital that term is; the concept of TF*PDF is adopted from Bun and Ishizuka (2002).

Bun and Ishizuka (2002) indicated that the weight of a term from a channel is linearly proportional to the term's within-channel frequency and exponentially proportional to the ratio of documents containing the term in the channel. Bun and Ishizuka defined the PDF weight of a term by Equations (3) and (4).

$$W_j = \sum_{c=1}^{c=D} \left|F_{jc}\right| exp\left(\frac{n_{jc}}{N_c}\right) \tag{3}$$

$$\left|F_{jc}\right| = \frac{F_{jc}}{\sqrt{\sum_{k=1}^{k=K} F_{kc}^2}} \tag{4}$$

where,

- $W_j$ denotes the weight of term $j$
- $F_{jc}$ is the frequency of term $j$ in channel $c$
- $n_{jc}$ stands for the number of documents in channel $c$ where term $j$ occurs
- $N_c$ denotes the total number of documents in channel $c$
- $K$ represents the total number of terms in a channel
- $D$ is the number of channels

This study alters *idf*-modified cosine similarity defined by Erkan and Radev's work (2004) into the *pdf*-modified cosine similarity as fallow:

$$pdf\text{-modified-cos}(s_i,s_j) = \frac{\sum_{w \in x,y} tf_{w,x} \times tf_{w,y} \times (pdf_w)^2}{\sqrt{\sum_{x_i \in x}(tf_{x_i,x} \times pdf_{x_i})^2} \times \sqrt{\sum_{y_i \in y}(tf_{y_i y} \times pdf_{y_i})^2}} \tag{5}$$

Where,

- $tf_{w,x}$ stands for the numbers of occurrence of term $w$ in sentence $x$,
- $tf_{w,y}$ stands for the numbers of occurrence of term $w$ in sentence $y$,
- $pdf_w$ means the PDF weight of term $w$.
- $x_i$ indicates the $i$th word in sentence $x$
- $y_i$ indicates the $i$th word in sentence $y$
- $x$ means the $x$th sentence
- $y$ means the $y$th sentence

- Step 2. Construct a sentence similarity matrix by calculating the PDF-modified cosine similarity of every pair of sentences. This can depict the similarity network for a news event.

### 3.2.3 Stage 3. Establish a relationship matrix

Calculate the relationship between each pair of sentences and construct the sentence network.

- Step 1. Select the threshold above which the values in the similarity matrix must be considered.
- Step 2. Build the sentence relationship matrix: Values from the sentence similarity matrix that are higher than threshold $h$ must be transferred to the sentence relationship matrix. The default value of the threshold $h$ is the median of the sentence similarity matrix, but the value can be customized by the user. If the similarity between two sentences is higher than the threshold $h$, this study defines these two sentences as *related*. Otherwise, if the similarity between two sentences is lower than $h$, his study defines these sentences as *unrelated*. The method of transformation between sentence relationship matrix and sentence similarity matrix is shown as follows:

$$\mathrm{R}_{(i,j)} = \begin{cases} 1\,, S_{(i,j)} > h \\ 0\,, \text{otherwise} \end{cases} \tag{6}$$

where,

- $R$ is the sentence relationship matrix
- $R_{(i,j)}$ denotes the relationship between the $i$th and the $j$th sentences in matrix $R$
- $S$ is the sentence similarity matrix
- $S_{(i,j)}$ denotes the similarity between the $i$th and the $j$th sentences in matrix $S$
- $h$ is the threshold

### 3.2.4 Stage 4. Output the summary

Output the summary by sorting the sentences in their order of importance.

- Step 1. Determine output length: Users can determine the length of the output summary by inputting either the number of sentences or the percentage of the original un-summarized documents.
- Step 2. Output the summary: The algorithm sorts the sentences in chronicle order and outputs the filtered sentences as a summary.

## 3.3 Data Preprocessing

Fig. 4 displays the relationships among *channel*, *document*, and *sentence*. In this study, a document is a news report; a channel is defined as the specific date of a news story. The channel consists of documents. Each document comprises several *sentences*. Each sentence has a unique name. The code defines $\mathrm{d}_x$-$\mathrm{s}_y$ to be the unique name for the $y$th sentence of the $x$th day.

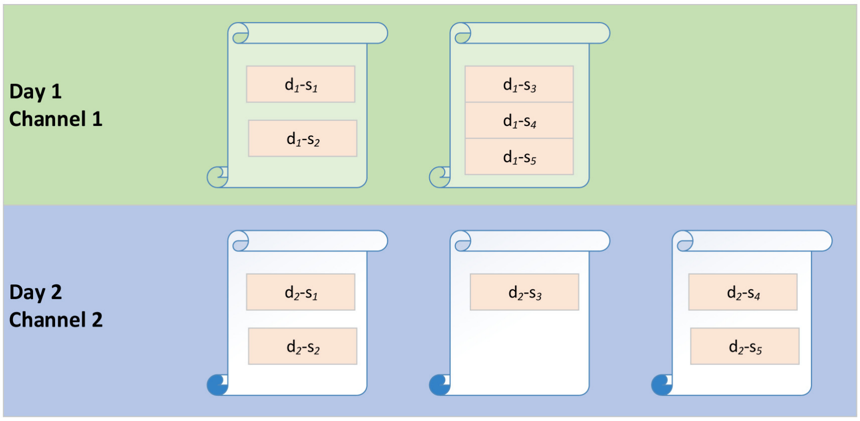

**Fig. 4.** Relationships among channel, document, and sentence

This study uses the MySQL Stopwords list[1] as a reference. The MySQL Stopwords list contains 543 frequently used words in English. This study displays a set of raw data as an example in Table 1(a). Stop words such as "are" in $d_1$-$s_1$ and "is" and "a" in $d_1$-$s_2$ are removed according to the stop word list. Table 1(b) shows the result of removing stop words. In the process of stemming, "cats" and "pets" in $d_1$-$s_1$ return to their stems "cat" and "pet." "Ate" in $d_2$-$s_1$ is transformed to its verb root "eat."

After the removal of stop words and stemming, the words remaining in the sentences are defined as terms. For example, in Table 1(b) "cat" and "pet" are terms in $d_1$-$s_1$. After that, the occurrences of each term in all sentences are calculated, and the term-sentence matrix is thus constructed.

The information in Table 1(c) serves to establish Table 1(d). In $d_1$-$s_1$, only the terms "cat" and "pet" appear, so the values of "cat" and "pet" are 1 and the rest of the terms are 0 in row $d_1$-$s_1$ of Table 1(d).

**Table 1(a).** Display raw data

| Date | Code | Content |
|---|---|---|
| day 1 | $d_1$-$s_1$ | Cats are pets. |
| | $d_1$-$s_2$ | Fish is a pet. |
| day 2 | $d_2$-$s_1$ | Cat ate the fish. |
| | $d_2$-$s_2$ | Fish was dead. |

**Table 1(b).** Remove stop words

| Date | Code | Content |
|---|---|---|
| Day 1 | $d_1$-$s_1$ | Cats pets |
| | $d_1$-$s_2$ | Fish pet |
| Day 2 | $d_2$-$s_1$ | Cat ate fish. |
| | $d_2$-$s_2$ | Fish dead. |

**Table 1(c).** Stem the word

| Date | Code | Content |
|---|---|---|
| day 1 | $d_1$-$s_1$ | cat pet |
| | $d_1$-$s_2$ | fish pet |
| day 2 | $d_2$-$s_1$ | cat eat fish. |
| | $d_2$-$s_2$ | fish die |

**Table 1(d).** Construct the term-sentence matrix

| | cat | pet | fish | eat | die |
|---|---|---|---|---|---|
| $d_1$-$s_1$ | 1 | 1 | 0 | 0 | 0 |
| $d_1$-$s_2$ | 0 | 1 | 1 | 0 | 0 |
| $D_2$-$s_1$ | 1 | 0 | 1 | 1 | 0 |
| $d_2$-$s_2$ | 0 | 0 | 1 | 0 | 1 |

[1] https://dev.mysql.com/doc/refman/5.5/en/fulltext-stopwords.html

### 3.4 PDF-Modified Cosine Similarity and the Sentence Similarity Matrix

Table 2 uses the information in Table 1(d) to illustrate the method of deriving the PDF weight. The row labels $d_1$ and $d_2$ represent the term frequencies in day 1 and day 2, respectively. For example the term "pet" appears 2 times in day 1, but appears 0 times in day 2. The PDF weight of each term must be calculated. As an example, the PDF weight of "cat" can be calculated as follows. In day 1, term "cat" appears in once in two documents, therefore the value is exp(1/2) divided by the term vector of day 1. Similarly, in day 2, term "cat" appears in once in two documents, therefore the value is exp(1/2) divided by the term vector of day 2.

As an example, the PDF weight of "fish" can be calculated as follows. In day 1, term "fish" appears in once in two documents, therefore the value is exp(1/2) divided by the term vector of day 1. Similarly, in day 2, term "fish" appears in both two documents, therefore the value is exp(1/2) divided by the term vector of day 2. The other PDF values are calculated similarly and are shown in Table 2.

$$\exp(1/2)*(1/\sqrt{1^2+1^2+2^2+0^2+0^2})+\exp(1/2)*(1/\sqrt{1^2+2^2+0^2+1^2+1^2})=1.296.$$

$$\exp(1/2)*(1/\sqrt{1^2+1^2+2^2+0^2+0^2})+\exp(2/2)*(1/\sqrt{1^2+2^2+0^2+1^2+1^2})=2.728.$$

**Table 2.** PDF weights of all terms

| | Cat | fish | pet | eat | die |
|---|---|---|---|---|---|
| $d_1$ | 1 | 1 | 2 | 0 | 0 |
| $d_2$ | 1 | 2 | 0 | 1 | 1 |
| *pdf* | 1.296 | 2.728 | 2.219 | 0.623 | 0.623 |

The sentence similarity matrix is established by calculation of the PDF-modified cosine similarity between every pair of sentences. Table 3 shows the sentence similarity matrix with the weights derived from Table 2 Consider, for example, the similarity between $d_1$-$s_1$ and $d_1$-$s_2$; the PDF-modified cosine similarity is calculated as follows:

$$\frac{1\times0\times1.296^2+1\times1\times2.728^2+0\times1\times2.219^2+0\times0\times0.623^2+0\times0\times0.623^2}{\sqrt{1\times1.296^2+1\times2.728^2+0\times2.219^2+0\times0.623^2+0\times0.623^2}\times\sqrt{0\times1.296^2+1\times2.728^2+1\times2.219^2+0\times0.623^2+0\times0.623^2}}=0.549$$

Table 3 indicates that the PDF-modified similarity between $d_1$-$s_2$ and $d_2$-$s_2$ is 0.756; however, for another example, the PDF-modified similarity between $d_1$-$s_1$ and $d_2$-$s_2$ is calculated as follows:

$$\frac{1\times0\times1.296^2+1\times0\times2.728^2+0\times1\times2.219^2+0\times0\times0.623^2+0\times1\times0.623^2}{\sqrt{1\times1.296^2+1\times2.728^2+0\times2.219^2+0\times0.623^2+0\times0.623^2}\times\sqrt{0\times1.296^2+0\times2.728^2+1\times2.219^2+0\times0.623^2+1\times0.623^2}}=0$$

This means $d_2$-$s_2$ is more similar to $d_1$-$s_2$ than to $d_1$-$s_1$.

**Table 3.** Sentence similarity matrix

| | $d_1$-$s_1$ | $d_1$-$s_2$ | $d_2$-$s_1$ | $d_2$-$s_2$ |
|---|---|---|---|---|
| $d_1$-$s_1$ | 1 | 0.549 | 0.212 | 0 |
| $d_1$-$s_2$ | 0.549 | 1 | 0.686 | 0.756 |
| $d_2$-$s_1$ | 0.212 | 0.686 | 1 | 0.862 |
| $d_2$-$s_2$ | 0 | 0.756 | 0.862 | 1 |

### *3.5 Sentence Relationship Matrix*

The sentence similarity matrix in Table 3 shows the similarities between certain sentences. Complex network includes Boolean matrix technique; the sentence similarity matrix can be transform into a Boolean matrix, termed the sentence relationship matrix.

Table 4 shows the sentence relationship matrix derived from Table 3 For the convenience of further demonstration, a threshold $h$ = 0.7 can serve as an example. Table 3 indicates that the similarity between $d_1$-$s_1$ and $d_1$-$s_2$ is 0.549, which is lower than the threshold 0.7. Consequently, the relationship between $d_1$-$s_1$ and $d_1$-$s_2$ is 0. In the same way, the similarity between $d_2$-$s_2$ and $d_1$-$s_2$ is 0.756, which is higher than the threshold 0.7. Hence, the relationship between $d_2$-$s_2$ and $d_1$-$s_2$ is 1. The sentence relationship matrix is mathematically equivalent to its graph form. Each sentence is a vertex and each cell entry in the matrix is an edge in the graph. The matrix in Table 4 is equivalent to Fig. 5.

Fig. 5 shows the relationship between each pair of sentences. A line between two sentences indicates that the two sentences are related. Thus, the relationships among sentences weave a network that can undergo degree centrality. Fig. 5 shows that $d_2$-$s_2$ has the largest number of relationships with other sentences. This means $d_2$-$s_2$ is one of the most crucial sentences in the semantic network. However, if the number of sentences were much higher, centrality measures from complex network could be used to extract pivotal sentences.

**Table 4.** Sentence relationship matrix

| $h$=0.7 | $d_1$-$s_1$ | $d_1$-$s_2$ | $d_2$-$s_1$ | $d_2$-$s_2$ |
|---|---|---|---|---|
| $d_1$-$s_1$ | 1 | 0 | 0 | 0 |
| $d_1$-$s_2$ | 0 | 1 | 0 | 1 |
| $d_2$-$s_1$ | 0 | 0 | 1 | 1 |
| $d_2$-$s_2$ | 0 | 1 | 1 | 1 |

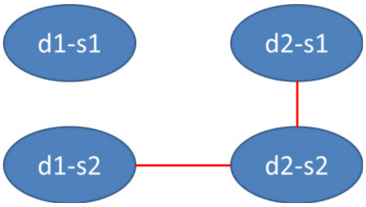

**Fig. 5.** Graph form of Table 4

### 3.6 Result of TF*PDF Multi-document Summarization

Freeman (1978) first depicted degree centrality for a measurement of centrality in graph theory and complex network. Freeman (1978) defined degree centrality as the number of edges that connected a vertex to other vertices. In Table 4 (or Fig. 2), which is equivalent to Table 4), $d_1$-$s_1$ is not related to any other sentence. As a result, the degree centrality of $d_1$-$s_1$ is 0. On the contrary, $d_2$-$s_2$ is related to $d_1$-$s_2$ and $d_2$-$s_1$, so the degree centrality of $d_2$-$s_2$ is 2. The calculation of degree centrality is displayed as Equation (7). The degree centrality of a sentence is calculated by adding up all the relationships that share the same column as the sentence, and then subtracting 1. The 1 must be subtracted to remove the diagonal element because the relationship of the sentence with itself is always 1.

$$\text{Degree centrality}(s) = \text{colSum(s)} - 1 \tag{7}$$

where,

- Degree centrality($s$) stands for the degree centrality of sentence $s$
- colSum($s$) is the summation of all elements of the same column as $s$

In this study, the sentences are ranked by degree centrality. The user defines the parameter $k$, and then the algorithm outputs the $k$ sentences with the highest degree centrality values as the result summary. If the degree centrality is the same, the algorithm compares the highest similarity values in the sentence similarity matrix. For the sample data in Table 5, the output summary of the raw data in Table 1 is “Fish was dead.” and “Cat ate the fish.”

**Table 5.** Result of summarization when $k$ = 2

| Rank | Degree | Highest Similarity | Code | Content |
|---|---|---|---|---|
| 1 | 2 | 0.862 | $d_2$-$s_2$ | Fish was dead. |
| 2 | 1 | 0.862 | $d_2$-$s_1$ | Cat ate the fish. |
| 3 | 1 | 0.759 | $d_1$-$s_2$ | Fish is a pet. |
| 4 | 0 | 0.549 | $d_1$-$s_1$ | Cats are pets. |

Algorithm 1 specifies the main processes of TF*PDF multi-document summarization, excluding the data preprocessing.

**Algorithm 1. PMCN multi-document summarization algorithm**

```
Step 1. Remove the stop words.
Step 2. Stem the words.
Step 3. Calculate the frequency of each term in every sentence to construct the term-sentence
        matrix.
Step 4. Calculate the PDF-modified cosine similarity of every pair of sentences to establish the
        sentence similarity matrix.
Step 5. Define parameter $h$ as a threshold of similarity for the sentence relationship matrix.
```

Step 6. Derive the degree centrality of each sentence.
Step 7. Rank the sentences by degree centrality.
Step 8. Select the top *k* sentences.
Step 9. Sort the selected sentences in chronicle order.

## 4. Validation of PMCN Multi-document Summarization Algorithms

Task d30001t of the DUC2004 dataset was used for validation. Section 4.1 shows the results of data preprocessing and constructing the term-sentence matrix. Section 4.2 demonstrates the PDF-modified cosine similarity and sentence similarity matrix. Section 4.3 illustrates the transformation of the sentence similarity matrix into the sentence relationship matrix. Section 4.4 displays the construction of the summary.

### *4.1 Data Preprocessing and Constructing Term-Sentence Matrix*

Table 6 shows a brief overview of tasks d30001t, d30002t, and d30003t. The tasks include 10 documents with similar numbers of sentences, but task d30002t has a notably larger standard deviation for the number of sentences in the documents.

**Table 6.** Overview of tasks d30001t, d30002t, and d30003t

| | Nds documents | sentences | terms | Sentences per doc. | Sentences sd |
|---|---|---|---|---|---|
| d30001t | 10 | 194 | 782 | 19.4 | 5.95 |
| d30002t | 10 | 172 | 932 | 17.2 | 16.38 |
| d30003t | 10 | 187 | 886 | 18.7 | 8.15 |

Table 7 shows a sample of the sentences in the d30001t dataset. After dataset preprocessing, including the removal of stop words and stemming, the terms can be identified, and several terms are shown as the column headings of Table 8.

**Table 7.** Content of task d30001t

| | |
|---|---|
| $d_1s_1$ | Cambodian leader Hun Sen on Friday rejected opposition parties' demands for talks outside the country, accusing them of trying to “internationalize” the political crisis. |
| $d_1s_2$ | Government and opposition parties have asked King Norodom Sihanouk to host a summit meeting after a series of post-election negotiations between the two opposition groups and Hun Sen's party to form a new government failed. |
| $d_1s_3$ | Opposition leaders Prince Norodom Ranariddh and Sam Rainsy, citing Hun Sen's threats to arrest opposition figures after two alleged attempts on his life, said they could not negotiate freely in Cambodia and called for talks at Sihanouk's residence in Beijing. |
| $d_1s_4$ | Hun Sen, however, rejected that. |
| $d_1s_5$ | “I would like to make it clear that all meetings related to Cambodian affairs must be conducted in the Kingdom of Cambodia”, Hun Sen told reporters after a Cabinet meeting on Friday. |
| … | … |

After terms are identified, the term-sentence matrix can be constructed. Table 8 shows a part of the term-sentence matrix derived from the data in Table 7.

**Table 8.** Result of term-sentence matrix (partial) of task d30001t

| | Hun | Sen | Opposit | Combodia | Meet | Sihanouk | Two | Reject | Leader | Negoti | ⋯ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $d_1s_1$ | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | ⋯ |
| $d_1s_2$ | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | ⋯ |
| $d_1s_3$ | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | ⋯ |
| $d_1s_4$ | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | ⋯ |
| $d_1s_5$ | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | ⋯ |

### 4.2 PDF-Modified Cosine Similarity and Sentence Similarity Matrix

By using the information in Table 8, the PDF weight can be calculated for each term. Table 9 lists the 15 terms with the highest PDF weights. The PDF weight indicates the importance of a term.

**Table 9.** Fifteen terms from task d30001t with the highest PDF weights

| | Hun | Sen | Parti | govern | ranariddh |
|---|---|---|---|---|---|
| *pdf* | 5.886 | 5.886 | 3.938 | 2.100 | 2.064 |
| | opposit | said | Rainsi | sam | cambodia |
| *pdf* | 2.054 | 2.008 | 1.493 | 1.436 | 1.210 |
| | two | nation | Will | elect | assembl |
| *pdf* | 1.126 | 1.115 | 1.114 | 1.087 | 1.062 |

Using the PDF weights shown in Table 9, the PDF-modified cosine similarity between every pair of sentences can be derived, and the sentence similarity matrix can be constructed; part of that matrix is shown in Table 10.

**Table 10.** Sentence similarity matrix (partial) of task d30001t

| | $d_1s_1$ | $d_1s_2$ | $d_1s_3$ | $d_1s_4$ | $d_1s_5$ | $d_1s_6$ | ⋯ |
|---|---|---|---|---|---|---|---|
| $d_1s_1$ | 1.000 | 0.863 | 0.797 | 0.869 | 0.858 | 0.091 | ⋯ |
| $d_1s_2$ | 0.863 | 1.000 | 0.662 | 0.633 | 0.621 | 0.000 | ⋯ |
| $d_1s_3$ | 0.797 | 0.662 | 1.000 | 0.811 | 0.807 | 0.000 | ⋯ |
| $d_1s_4$ | 0.869 | 0.633 | 0.811 | 1.000 | 0.974 | 0.000 | ⋯ |
| $d_1s_5$ | 0.858 | 0.621 | 0.807 | 0.974 | 1.000 | 0.103 | ⋯ |
| $d_1s_6$ | 0.091 | 0.000 | 0.000 | 0.000 | 0.103 | 1.000 | ⋯ |
| ⋯ | ⋯ | ⋯ | ⋯ | ⋯ | ⋯ | ⋯ | |

### 4.3 Transformation of Sentence Relationship Matrix

A threshold $h$ = 0.5 was selected for the sentence relationship matrix. A sample of the sentence relationship matrix is presented in Table 11. Table 11 reveals that the pairs of sentences from $d_1s_1$ to $d_1s_5$ are related to each other; however, $d_1s_6$ does not relate to any sentences from $d_1s_1$ to $d_1s_5$.

**Table 11.** Sentence relationship matrix (partial) of task d30001t

| | $d_1s_1$ | $d_1s_2$ | $d_1s_3$ | $d_1s_4$ | $d_1s_5$ | $d_1s_6$ | … |
|---|---|---|---|---|---|---|---|
| $d_1s_1$ | 1 | 1 | 1 | 1 | 1 | 0 | … |
| $d_1s_2$ | 1 | 1 | 1 | 1 | 1 | 0 | … |
| $d_1s_3$ | 1 | 1 | 1 | 1 | 1 | 0 | … |
| $d_1s_4$ | 1 | 1 | 1 | 1 | 1 | 0 | … |
| $d_1s_5$ | 1 | 1 | 1 | 1 | 1 | 0 | … |
| $d_1s_6$ | 0 | 0 | 0 | 0 | 0 | 1 | … |
| … | … | … | … | … | … | … | |

The content of Table 11 is equivalent to a subgraph of the relationship graph shown in Fig. 6. The results of this research indicate that for this task, two groups of sentences are highly related to the other sentences in each group. Five critical sentences are related to both of those two groups. A well-designed algorithm should output those critical sentences in its summary.

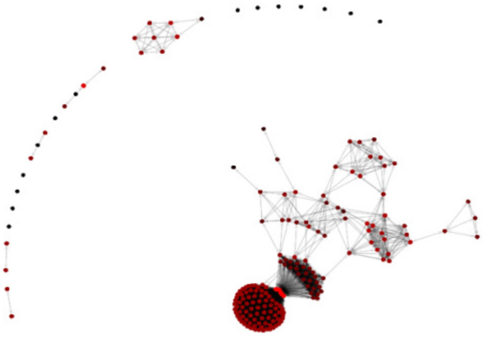

**Fig. 6.** Relationship graph of task d30001t

### 4.4 Construction of the News Summary

The degree centrality of each sentence can be calculated from the results listed in Table 11. The degree centrality values of sentences from task d30001t are shown in Table 12; the degree centrality value of $d_1s_2$ is higher than those of other sentences. Thus, it should appear in the final summary. Conversely, because of its low degree centrality, $d_1s_6$ should not appear in the summary.

**Table 12.** Degree centrality (partial) of task d30001t

| | $d_1s_1$ | $d_1s_2$ | $d_1s_3$ | $d_1s_4$ | $d_1s_5$ | $d_1s_6$ |
|---|---|---|---|---|---|---|
| Degree Centrality | 91 | 118 | 90 | 90 | 90 | 2 |

Finally, the algorithm selects the top k = 5 sentences for the summary and sorts the selected sentences in chronicle order; the result is shown in Table 13.

**Table 13.** Summary of results of task d30001t for $k$ = 5

| Encode | Content |
|---|---|
| $d_1$-$s_2$ | Government and opposition parties have asked King Norodom Sihanouk to host a summit meeting after a series of post-election negotiations between the two opposition groups and Hun Sen's party to form a new government failed. |
| $d_2$-$s_2$ | Cambodian leader Hun Sen's ruling party and the two-party opposition had called on the monarch to lead top-level talks but disagreed on its location. |
| $d_6$-$s_3$ | Hun Sen's Cambodian People's Party dropped insistence on a joint assembly chairmanship shared by Ranariddh and party boss Chea Sim, the current speaker. |
| $d_8$-$s_8$ | He said it contained indirect language and loopholes that suggest he and his Sam Rainsy Party members are still under threat of arrest from Hun Sen's ruling party. |
| $d_{10}$-$s_{11}$ | Government and opposition parties have asked King Norodom Sihanouk to host a summit meeting after a series of post-election negotiations between the two opposition groups and Hun Sen's party to form a new government failed. |

## *4.5 Evaluation of TF*PDF-Based Multi-document Summarization Algorithms*

In this study, 23 tasks from the DUC2004 dataset were used for evaluation. The DUC2004 dataset has a manual summary that was regarded as the “correct answer.” ROUGE, a *n*-gram based method, was used as a validation tool; precision, recall, and F-score were compared for the PMCN (PDF-Modified similarity and Complex Network) multi-document summarization algorithm and LexRank. Both PMCN and LexRank process a 4-sentence summary, which is the same as the reference hand-made summary. The results of comparison of ROUGE-1 scores are shown in Tables 14. Though in most of the cases, the proposed algorithm performs lower F-score than LexRank does, the proposed method performs well in recall. This means proposed algorithm capable to obtain important messages than LexRank does.

**Table 14.** Result of validation of dataset DUC2004

| Algorithm | PMCN | | | LexRank | | |
|---|---|---|---|---|---|---|
| Task | Recall | Precision | F-score | Recall | Precision | F-score |
| d30001t | .432 | .326 | **.372** | .427 | .268 | .330 |
| d30002t | .236 | .252 | .244 | .280 | .331 | **.295** |
| d30003t | .487 | .308 | **.377** | .417 | .267 | .326 |
| d30005t | .408 | .244 | **.306** | .183 | .328 | .235 |
| d30006t | .410 | .254 | .314 | .380 | .322 | **.349** |

| Algorithm | PMCN | | | LexRank | | |
|---|---|---|---|---|---|---|
| Task | Recall | Precision | F-score | Recall | Precision | F-score |
| d30007t | .353 | .243 | .288 | .324 | .289 | **.305** |
| d30008t | .396 | .266 | **.318** | .313 | .278 | .294 |
| d30010t | .359 | .333 | .346 | .373 | .326 | **.348** |
| d30011t | .390 | .218 | .280 | .272 | .458 | **.341** |
| d30015t | .381 | .188 | .252 | .377 | .301 | **.334** |
| d30017t | .485 | .233 | .315 | .425 | .281 | **.338** |
| d30020t | .483 | .213 | .296 | .313 | .319 | **.316** |
| d30022t | .312 | .277 | .294 | .407 | .274 | **.327** |
| d30024t | .518 | .239 | .327 | .326 | .336 | **.331** |
| d30026t | .536 | .172 | .260 | .292 | .341 | **.315** |
| d30027t | .130 | .158 | .143 | .310 | .250 | **.277** |
| d30028t | .404 | .344 | **.372** | .317 | .371 | .342 |
| d30029t | .393 | .300 | .340 | .413 | .335 | **.370** |
| d30031t | .408 | .362 | .384 | .439 | .384 | **.410** |
| d30033t | .408 | .311 | **.353** | .203 | .398 | .269 |
| d30034t | .356 | .309 | **.331** | .206 | .518 | .295 |
| d30036t | .177 | .213 | .193 | .248 | .405 | **.308** |
| d30037t | .445 | .307 | **.363** | .231 | .245 | .238 |

## 5. Conclusion

The contributions of this study are listed as follows. First, this study reports that PMCN produced a multi-document summarization algorithm, of which the F-measure scores were 0.042 and 0.051 higher than LexRank for tasks d30001t and d30003t, respectively. Second, the TF*PDF algorithm can summarize daily news; the concept of channel was replaced with the date of the news event. This produced chronicle ordering for the multiday news summarization algorithm. Third, in this study, the complex network concept of degree centrality is combined with the TF*PDF algorithm; the combined method, called PMCN, constructs relationship networks among sentences for writing news summaries.

## References

Freeman, L. C. (1978). Centrality in social networks conceptual clarification. *Social Networks, 1*(3), 215-239.

Luhn, H. P. (1960). Key word-in-context index for technical literature (kwic index). *American Documentation, 11*(4), 288-295.

Allan, J., Carbonell, J. G., Doddington, G., Yamron, J., & Yang, Y. (2003). Topic detection and tracking pilot study final report. Retrieved from
https://kilthub.cmu.edu/articles/Topic_Detection_and_Tracking_Pilot_Study_Final_Report/66

10943

Antiqueira, L., Oliveira Jr, O. N., da Fontoura Costa, L., & Nunes, M. D. G. V. (2009). A complex network approach to text summarization. *Information Sciences, 179*(5), 584-599.

Bun, K. K., & Ishizuka, M. (2002, December). Topic extraction from news archive using TF* PDF algorithm. In *Proceedings of the Third International Conference on Web Information Systems Engineering, 2002. WISE 2002*. (pp. 73-82). IEEE.

Carbonell, J. G., Yang, Y., Lafferty, J., Brown, R. D., Pierce, T., & Liu, X. (1999). CMU Approach to TDT-2: Segmentation, Detection, and Tracking. Retrieved from https://kilthub.cmu.edu/articles/CMU_Approach_to_TDT-2_Segmentation_Detection_and_Tracking/6621371/files/12117779.pdf

Daniel, N., Radev, D., & Allison, T. (2003, May). Sub-event based multi-document summarization. In *Proceedings of the HLT-NAACL 03 on Text summarization workshop-Volume 5* (pp. 9-16). Association for Computational Linguistics.

Erkan, G., & Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research, 22*, 457-479.

Hu, M., & Liu, B. (2004, August). Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 168-177). ACM.

Lin, C. Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out* (pp. 74-81).

Lovins, J. B. (1968). Development of a stemming algorithm. *Mech. Translat. & Comp. Linguistics, 11*(1-2), 22-31.

Marujo, L., Ling W., Ribeiro, R., Gershman, A., Carbonell, J., Matos, D. M., & Neto, H. P. (2016). Exploring events and distributed representations of text in multi-document summarization. *Knowledge-Based Systems, 94*, 33-42.

Mihalcea, R., & Tarau, P. (2004). Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing* (pp. 404-411).

Ouyang, Y., Li, W., Li, S., & Lu, Q. (2011). Applying regression models to query-focused multi-document summarization. *Information Processing & Management, 47*(2), 227-237.

Popescu, A. M., & Etzioni, O. (2007). Extracting product features and opinions from reviews. In *Natural language processing and text mining* (pp. 9-28). Springer, London.

Walker, C., Strassel, S., Medero, J., & Maeda, K. (2006). ACE 2005 Multilingual Training Corpus. In *Linguistic Data Consortium, Philadelphia, 57.*

Wilson, T., Wiebe, J., & Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing.*

Yang, Y., Carbonell, J. G., Brown, R. D., Pierce, T., Archibald, B. T., & Liu, X. (1999). Learning approaches for detecting and tracking news events. *IEEE Intelligent System, 14*(4), 32–43.

Yang, Y., Pierce, T., & Carbonell, J. (1998). A study of retrospective and on-line event detection. In *Proceedings of the 21$^{st}$ Annual International ACMSIGIR Conference on Research and Development in Information Retrieval*, 28–36.

## [ About the authors ]

**Yi-Ning Tu** currently is an associate professor in the Department of Statistics and Information Science, College of Management, Fu Jen Catholic University, (R.O.C.) Taiwan. She also served as the reviewer of SSCI/EI journals. Her research interests include text mining, data mining, artificial intelligence, decision support systems and the application in big data. She also served as the adjunct associate professor at National Chengchi University, (R.O.C.) Taiwan. Besides, she is also invited as the visiting scholar at Chonnam National University (CNU) International Summer Session 2018 (Korea) and providing the course as "Introduction to Data Analysis." Her academic personal website is https://sites.google.com/view/yiningtu. She may be contacted as E-mail: eniddu@gmail.com or 082435@mail.fju.edu.tw

**Wet-Tse Hsu** is graduated from Fu Jen Catholic University, and is currently taking Master's degree at National Taipei University. His works focus specifically on text mining, dimensionality reduction, and class imbalanced problems. Email: victor.playtion@gmail.com or 402422569@mail.fju.efu.tw