Kaunia, Vol.XII, Num.1, April 2016, pp. 9~16
ISSN 1829-5266 (paper), ISSN 2301-8550 (online)
Available online at http://ejournal.uin-suka.ac.id/saintek/kaunia

Summarizing Online Media News Content Using TFIDF Algorithm

M. Didik R. Wahyudi

Department of Informatics Engineering, Faculty of Science and Technology, UIN Sunan Kalijaga,
Jl. Marsda Adicucipto Yogyakarta 55281, Indonesia
Correspondance; Email: m.didik@uin-suka.ac.id

Abstract

Online news media has an important role as a source of news and information for the public. In reading the news, the public wants to read the article as short as possible, but can obtain news core. Desire to obtain information quickly can be done by reading the summary of a story. One algorithm that can be used to summarize the content of the news in the online news media are TFIDF algorithm. The contents of the selected message will be read by using the Document Object Model library in the PHP programming language. The results of this research is a summary of the selected online news media are then read by the DOM library to take the news content. Furthermore, the news content will be summarized by using TFIDF algorithms.

Keywords: news, summarization, DOM, TFIDF.

Introduction

Developments in information technology are improved very fast. The need for information is increasing along with the demand to know information from any places in the world. Technology allows one to obtain and transmit information in a variety of digital formats through the Internet. The information in the various fields can be easily obtained. One medium that provides a variety of information is online news media.

The online news media develop very rapidly. Almost all print newspapers have a digital version that can be accessed freely by the public at any time. The number of online news media is not necessarily able to meet the needs of information quickly and accurately. Usually, however, an online news media page contains very long content, so it takes a long time to read and understand the information. The number of news sources and news footage presented in one page will cause problems for people who would like to obtain information quickly and accurately. For employees or an office employee, the time available for reading news are very limited. To obtain information quickly and accurately is to read a news summary from online news media. By reading this news summary, the information can still be obtained in a short time, but still has a high degree of accuracy.

Summarizing a document makes it easy to understand the contents of the document. Document summarization is the process of getting important information from each sub-section of the entire document. It can be done manually and automatically. When document summarization is done manually, it takes a long time compared with the automatically document summarization (Aristotle et al. 2012). Some of the techniques that can be done to document summarization including the extraction technique and abstraction techniques (Ježek and Steiberger 2008). Extract are summaries completely consisting of word sequences copied from the original document. As the word sequences can be used phrases, sentences or paragraphs. As expected, extracts suffer from inconsistencies, lack of balance, and lack of cohesion. Sentences may be extracted out of context, anaphoric reference may be broken. Abstract are summaries containing word sequences not present in the original document. Up to now it is too hard task for computer research to solve it successfully (Ježek and Steiberger 2008).

One document extraction technique is TFIDF. TFIDF algorithm initiated from 1972. Karen Jones Sp¨arck publish in the Journal of Documentation a paper entitled A statistical interpretation of the term specificity and its application in retrieval, a measurement of the specificity of a term that became known as the inverse document frequency (IDF) (Jones, KS, 1972). The measurement is based on a calculation of the number of times a term in the document. In this study, done with the online news peringkasan TFIDF method.

Research Methods

This research was conducted in several steps, namely:

a. Specifies the web address of the online news media to be summarized
b. Parsing the web page to retrieve the news link address by using the library Document Object Model (DOM)
c. Read a web page that is chosen to take the headlines and news content by using the DOM library

News content summarization process by using TFIDF algorithms.

Results and Discussion

Determination of online news media web address

There are many online news media web pages. In principle, all the online news media web pages can be choosed to be summarized news content. Online news media are quite famous today as detik.com, republika.co.id, cnnindonesia.com, etc. In this study, will be used http://www.cnnindonesia.com online news media. However, if desired can be replaced by others.

Parsing web page online news media

Once the web page online news media, http://www.cnnindonesia.com is to do the process parsing the retrieved address the main page to link existing news. The making of the link address parsing news using the Document Object Model (DOM) library. Document Object Model (DOM) is an application programming interface (API) for valid HTML and XML document format. DOM defines the logical structure of a document and how documents are accessed and manipulated. In the DOM specification, the term "document" is used in a broad sense. XML is used as a way to represent various types of information that can be stored in diverse systems and much of this would traditionally be seen as data rather than as documents. Nevertheless, the present data as XML documents, and the DOM can be used to manage that data.

With DOM, programmers can build documents, navigate their structure, add, modify, or delete elements and content. Anything found in an HTML or XML document can be accessed, modified, deleted, or added using the Document Object Model. With a few exceptions in particular, the DOM interfaces for the XML internal and external subsets have not been determined.

As a W3C specification, one important objective for the DOM is to provide a standard programming interface that can be used in a variety of environments and applications. DOM is designed to be used with any programming language. In order to provide a precise specification, language-independent from DOM interface, has adapted to the specification of the Object Management Group (OMG) IDL [OMG IDL], as defined in 2.3.1 CORBA CORBA specification. In addition to the OMG IDL specification, DOM is also available in languages for Java and ECMAScript, which is an industrial standard scripting language based on JavaScript and JScript. As mentioned above, the DOM is a programming API for documents. It is based on the structure of objects similar to the structure of the document model. For example XHTML document following example:

```
<table>
  <tbody>
    <tr>
      <td>Shady Grove</td>
      <td>Aeolian</td>
    </tr>
    <tr>
      <td>Over the River, Charlie</td>
      <td>Dorian</td>
    </tr>
  </tbody>
</table>
```

Figure 1 XHTML document example.

DOM representation on the XHTML document above is:

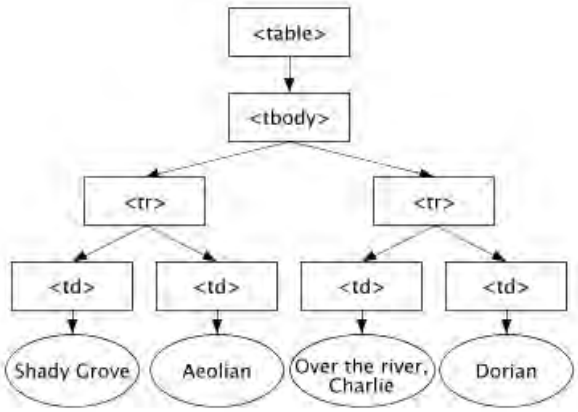

Figure 2 DOM Representation on XHTML Documents.

In the DOM, documents have a logical structure which is very much like a tree, which can contain more than one type of tree. Each document contains zero or one node type of document, the document element node and zero or more comments or processing instructions. The document element serves as the root element tree for the document. However DOM does not require that a document should be shaped like a tree structured or unstructured, nor does it require

specifically how the relationships among objects be implemented. DOM is a logical model that can be implemented in an easy manner. In this specification, the use of the term to describe a structured model is like a tree representation of the document. Use of the term "tree" when referring to the arrangement of the items of information can be achieved by using the method "tree-foot". One important property of DOM structure is structural isomorphism, i.e., if there are two DOM implementations are used to create a representation of the same document, it will create the same structure model, according to the XML Information Set.

One example of the implementation of DOM functions in PHP is a PHP library that was written by S. C. Chen inspired by the idea of Jose Solorzano to create html parsing on php4. This library is named simple_html_dom.php contains a summary of the command to read a file that having various an html and xml formats. Readings were performed by the library that is intended to create a structure of an html or xml document that will be captured from the structure can be taken a certain pattern.

In the process of online news reading, Document Object Model (DOM) library will be called in the php file index.php to read the news. Here's the script library calls:

```
include_once('simple_html_dom.php');
```

With the script library, each URL's news in http://www.cnnindonesia.com taken. Here is a script to take the news URL address:

```
include_once('simple_html_dom.php');
 $html = new simple_html_dom();
 $alamat=http://www.cnnindonesia.com;
 $html->load_file($alamat);
foreach($html->find('a') as $link) {
 $urlberita = $link->href;
 Echo ("<a
href=\"bcurl.php?urlnya=$urlberita\">
".$jdl."</a><br>"); }
```

$html variable is defined to make creating a new DOM object. The DOM object contains the index page of the contents of the variable `$address` are loaded with orders `$html->load_file ($address)`. Variables `$urlberita` contains all URL addresses news in http://www.cnnindonesia.com taken with command `foreach ($html -> find ('a') as $link)` and `$urlberita = $link -> href`. List of news generated url address to be delivered to bcurl.php making process titles and news content with the command: `echo ("<a href=\"bcurl.php?urlnya=$urlberita\">". $jdl."</a> Reviews");`. Here are the results of execution url index.php which displays the message for processing.

Figure 3 Execution result of index.php.

Capturing the title and content of the selected message

News url address, delivered to bcurl.php to grab headlines and news content. Making headlines and news content is also using the DOM library. Here is a script to capture the title and content of the news.

```
$url = $_GET['urlnya'];
$pisah=explode("/",$url);
$jdl = str_replace("-","
",$pisah[5]);
```

The command `$url=$_GET['url']` would take the address selected news to read his message attributes. The headline can be taken from one part of the URL address. The command `$split = explode ("/", $url)` will break into pieces url address in the variable array `$pisah`. The headline which is in the index number 5 are drawn with the command `$jdl= str_replace ("-","", $separation [5])`. Variables `$jdl` contain headlines.

The reading of the contents of the news by reading the news pages are selected by using the DOM library. This process will read the contents of the page shown in the URL address. Therefore, it is needed to know the structure of a web page to read the news. Based on observation, online news page at CNN Indonesia has a news format that contains the contents of the news as follows:

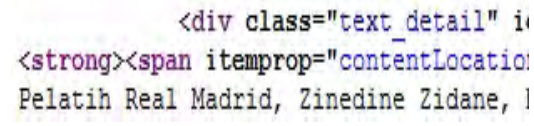

Figure 4 PageSource of news URL.

Detail news content lies in the html tag `<div class = "text_detail">`. HTML tags will be read

by the DOM library simple_html_dom.php to take the news content is needed. Here is a php script to take an html tag that contains the news content in question.

```
$htmlisi = new simple_html_dom();
$htmlisi->load_file($url);
foreach($htmlisi->find('div') as
$linkisi)
if ($linkisi->class=="text_detail")
$beritanya=str_replace("'","-
",(strip_tags($linkisi)));
```

The reading of the news web pages is defined by the new DOM object `simple_html_dom();` and put in the variable `$htmlisi`. Command `$htmlisi-> load_file ($url)` function to retrieve the contents of selected pages in the URL address. After the news pages are read and placed in the variable `$htmlisi` then the next step is to find all the html tag "div". This search process is done by using the script `foreach ($htmlisi-> find ('div') as $linkisi)`. All positions html tag "div" that is found will be placed in the variable `$linkisi`. Of all the html tag "div" is found, the next step is to find where the location of the property `class == "text_detail"`. It ordered a script is done by piece `if ($ linkisi -> class == "text_detail")`. Furthermore, the content of the news that has been found will be placed in the variable `$message`. Variable `$news` is filled with eliminating the html tag that accompanies and dispose of single quotes if found within the contents of the news. The process shown in the script `$news = str_replace (" '", "-", (strip_tags ($linkisi)))`.

Summary of News Contents with TFIDF Algorithm

The contents of the news that has been taken in the above process, will now be summarized with TFIDF algorithm. Algorithm Term Frequency-Inverse Document Frequency (TF-IDF) is a way of assigning weights to the relationship of a word (term) of the document. For a single document, each sentence is considered as a document. This algorithm combines two concepts for weight calculation, namely Term frequency (TF) which is the frequency of words (t) of the sentence (d). Document frequency (DF) is the sentence number where a word (t) appears. Frequency of words in a given document shows how important word in the document. The frequency of documents containing the word indicates how common the word. The weight of even greater if the word appears frequently in a document and getting smaller when it appears in many documents (Robertson, 2005). In this method, the weighting of words in a document is done by multiplying the value of TF and IDF. Weighting is obtained based on the number of occurrences term in the sentence (TF) and the number of occurrences of the term in the entire sentence in the document (IDF). The weight of a term greater if those terms appear frequently in a document and the smaller if the term appears in many documents (Grossman, 1998). IDF value of a term calculated using the following equation:

$$W_{td} = tf_{td} * idf \quad (1)$$

$$W_{td} = tf_{td} * log\left(\frac{N}{df_t}\right) \quad (2)$$

Information:

WTD	=	weighted word / token ss tt against documents
tftd	=	number of occurrences of the word / token in the document ss tt
N	=	the number of all the documents in the database
dft	=	number of documents that contain the word / token tt

Based on the formula (2), regardless of the value tftd, if N = dft where a word / token appear in any document, then we will get the result 0 (zero) for the calculation of idf, so the weight calculation shall be amended as follows:

$$W_{td} = tf_{td} * \left(log\left(\frac{N}{df_t}\right) + 1\right) \quad (3)$$

News summarization process is done by calling the TFIDF functions contained in the tfidf.php file. TFIDF file called from bcurl.php with the command:

```
include "tfidf.php";
```

TFIDF function called from curl.php file with the command:

```
$rksisiberita = tfidf($beritanya);
```

Variables `$rksisiberita` contains a summary of the news content that has been processed by the TFIDF algorithm in tfidf.php file. Here are the steps in the documents summarization process with TFIDF function in the tfidf.php file:

1. Take the news content and insert in TFIDF function. News content as a parameter of the TFIDF function for further processing. Retrieval command is: `function tfidf ($iberita).`
2. The process of cleaning special character of news content. This process is done by using the command `$sentence = preg_split ("/[.]+/", $iberita);`

 ISSN 1829-5266 (paper), ISSN 2301-8550 (online)

3. Breaking news content per sentence and then store it in the array. Solving the news content is done to capture the word in every sentence, and later said that will be analyzed by the TFIDF algorithm. Commands for breaking news content is added to the array with the command: `$sentence = array_slice ($sentence, 0, sizeof ($sentence)-1);`. An Array of news content is stored in the `$sentence` variable.
4. Cleaning of news content that has been broken each word of stop word. Stop word is a word that often appear in a sentence. Usually in the form of common words like “itu”, “ini”, “yang”, ”adalah” or ”yaitu”, etc. Here is a php script that works to eliminate the stop word.

```
$stopwords = file_get_contents
("stopwords.txt");
$stopwords = preg_split("/[\s]+/",
$stopwords);
foreach ($sentence as $key =>
$value) { $token =
preg_split("/[\d\W\s]+/",
strtolower($value));
$token = array_diff($token,
$stopwords);
```

The words that fall into the category of stop word stopwords.txt collected in a text file. Words stop word is extracted and stored in the variable `$stopwords` with command `$stopwords = preg_split ("/ [\ s] + /", $ stopwords)`. Furthermore, the words that exist in news content already stored in the array `$sentence` cleared of stop word with command `$tokens = array_diff ($ token, $ stopwords)`. Words of news content that has been cleared of the stop word is stored in the variable $token.

5. Collection and calculation of words that appear in news content. Every word that is clean in the variable `$token` will be calculated frequency of occurrence for the next IDF its calculated value. Here is a script that is used to calculate the frequency of occurrence of the word:

```
$token=array_count_values($token);
foreach ($token as $key => $value)
{
 if (array_key_exists($key,
$df_token))  $df_token[$key]++;
 else if (!isset($df_token[$key]))
$df_token[$key] = 1;}
```

If a word appears repeatedly, it will be added frequency of occurrence with command `if (array_key_exists ($key, $df_token)) $df_token [$key] ++`. If a word has never appeared, then the word will be added in a new word to be counted its frequency of occurrence with command `if (!Isset ($df_token [$key])) $df_token [$key] = 1`.

6. Calculate the value of IDF. Once the frequency of occurrence of each word counted, then counted the value of IDF. Here is a script to calculate the value of IDF:

```
foreach ($df_token as $key =>
$value) {
$df   = $value;
$inv_index[$key] = array();
$inv_index[$key]['idf'] =
log10($N/$df); }
```

Idf value is calculated based on the formula 2 above. In the script above, IDF value calculated by the command `$inv_index [$key] ['idf'] = log10 ($N / $df)`. The variable `$N` shows the number of sentences in news content. While the variable `$df` contains the number of a word that appears in a sentence.

7. The calculation of the TFIDF. TFIDF value is calculated by multiplying the value TFIDF value. Standard formula using the formula number 1 above. TFIDF calculation is done using the script:

```
foreach ($freq_word as $token =>
$tf)  $tf_idf += $tf *
$inv_index[$token]['idf'];
array_push($sentence_weight,
$tf_idf);
```

Tfidf value stored in the array with the command `array_push ($sentence_weight, $tf_idf)`. Variables `$sentence_weight` contain TFIDF value.

8. Summary of News Contents. Summary of News Contents process is based on the value of TFIDF. Here is a php script to summary of news contents:

```
$cmpr=floor($N*0.5);
arsort($sentence_weight);
$sorted =
array_slice($sentence_weight, 0,
$cmpr, true);
ksort($sorted);
foreach ($sorted as $key => $value)
$summary =
$summary.$sentence[$key].". ";
```

TFIDF value that has been stored in an array in descending order by using the command `arsort($sentence_weight)`. The next step is to determine many sentences to be taken based on

the TFIDF highest value. The number of sentences to be retrieved, stored in the variable $cmpr which is calculated by the formula `$cmpr = floor($N * 0.5)`. The composition of news content that has been summarized will then revert to the original sequence of news content with command `ksort($sorted)`. The next sentence summary of the results will be compiled and included in the variable `$summary in command $summary = $ summary, $ sentence [$key].'.'`.

Here are the results of summary of news contents with TFIDF algorithm.

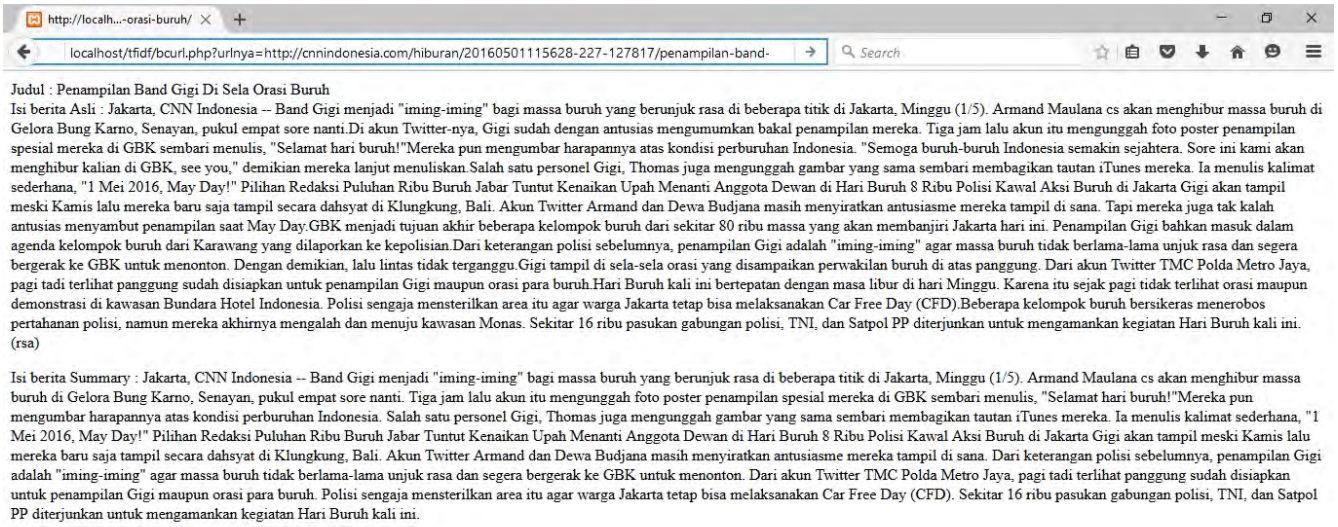

Figure 5 The Result of News Sumarization.

The first paragraph contains the original news content. The second paragraph contains the contents of the news that has been summarized by TFIDF algorithms. News summed up by 50%.

Conclusion

After the research process, news content summarization to acquire part of the contents of the most important news in the online news media successfully. The process of determining the body is the most important news, is done by using algorithms TFIDF. TFIDF algorithm based on frequent summarizing news frequency of words in a sentence. TFIDF algorithm implemented in the PHP programming language to summarize the content of the news has been successfully carried out. Intake of selected news content using DOM library in PHP.

References

Aristoteles, 2013, *Penerapan Algoritma Genetika pada Peringkasan Teks Dokumen Bahasa Indonesia*, Prosiding Semirata, vol. 1, no. 1.

Bakken, S., S., Aulbach, A., etc., 1997-2016, *PHP Manual*, PHP Documentation Group.

Bambang Kurniawan, dkk., 2012, *Klasifikasi Konten Berita Dengan Metode Text Mining*, Jurnal Dunia Teknologi Informasi, Vol. 1, Universitas Sumatera Utara, Medan

Grossman, D., Ophir, F. 1998. *Information Retrieval: Algorithm and Heuristics.* Kluwer Academic Publisher.

Jezek, K., Steinberger, J., 2008, *Automatic Text Summarization*, Proceedings of Znalosti, Bratislava, Slovakia. page:1-12, ISBN 978-80-227-2827-0.

Jones, K.S., 1972, “A statistical interpretation of term specificity and its application in retrieval”, Journal of documentation, Volume 28, Number 1.

Karmila,N.K.M., Kesiman, M.W.A., Darmawiguna,I.G.M, 2013, *Aplikasi Automated Text Summarization (ATS) Pembuat Lead (Teras) Berita dengan Text to Speech (TTS) Menggunakan Algoritma TF-IDF dan Vector Space Model*. (KARMAPATI) Volume 2, Nomor 6

Raharjo, S. & Winarko, E., 2014, *Klasterisasi, Klasifikasi Dan Peringkasan Teks Berbahasa Indonesia*. Prosiding Seminar Ilmiah Nasional Komputer dan Sistem Intelijen Vol. 8 Universitas Gunadarma, Depok, ISSN: 2302-3740

Robertson, S., 2004, *Understanding Inverse Document Frequency: On Theoritical* Arguments *for IDF*. Journal of Documentation; 60, 5; ABI/INFORM Global.

S. C. Chen, 2004, *PHP Simple HTML DOM* Parser.https://sourceforge.net/projects/simplehtmldom, accessed: 20/01/2016 jam 13:05

W3C, 2004, *Document Object Model (DOM) Level 3 Core Specification.* https://www.w3.org/TR/DOM-Level-3-Core/Overview.html, accessed: 09/02/2016 jam 21:00

Appendix

Index.php file content:

```
<?php
include_once('simple_html_dom.php');
$html = new simple_html_dom();
$alamat="http://www.cnnindonesia.com/
";
$html->load_file($alamat);
foreach($html->find('a') as $link){
if (substr("'".$link-
>href."'",1,24)==
"http://cnnindonesia.com/") {
$pecahurl = $link->href;
$pisah=explode("/",$pecahurl);
$jdl = ucwords(str_replace("-","
",$pisah[5]));
Echo ("<a
href=\"bcurl.php?urlnya=$pecahurl\">
".$jdl."</a><br>");}}
?>
```

Bcurl.php file content:

```
<?php
include_once('simple_html_dom.php');
include "tfidf.php";
$url = $_GET['urlnya'];
$pisah=explode("/",$url);
$jdl = str_replace("-","
",$pisah[5]);
$htmlisi = new simple_html_dom();
$htmlisi->load_file($url);
foreach($htmlisi->find('div') as
$linkisi)
if ($linkisi->class=="text_detail")
$beritanya=str_replace("'","-
",(strip_tags($linkisi)));
echo "Judul : ".ucwords($jdl)."<br>";
$output = tfidf($beritanya);
echo "Isi berita Asli :
".$output['original']."<p>";
echo "Isi berita Summary :
".$output['summary']."<p>";
?>
```

Tfidf.php file content:

```
<?php
function tfidf($iberita){
$tf_idf = 0;
$sentence  = preg_split("/[.]+/", $iberita);
$sentence=array_slice($sentence,0,sizeof($sentence)-1);
$N=sizeof($sentence);
$cmpr=floor($N*0.5);
$df_token   = array();
$inv_index  = array();
$sentence_weight = array();
$stopwords = file_get_contents("stopwords.txt");
$stopwords = preg_split("/[\s]+/", $stopwords);
foreach ($sentence as $key => $value) {
  $token = preg_split("/[\d\W\s]+/",strtolower($value));
  $token = array_diff($token, $stopwords);
  $token = array_values($token);
  $token = array_count_values($token);
  foreach ($token as $key => $value) {
    if (array_key_exists($key, $df_token))
       $df_token[$key]++;
    else if (!isset($df_token[$key]))
      $df_token[$key] = 1; }}
foreach ($df_token as $key => $value) {
$df = $value;
$inv_index[$key] = array();
$inv_index[$key]['idf'] = log10($N/$df); }

foreach ($sentence as $key => $value) {
$word = preg_split("/[\d\W\s]+/",strtolower($value));
$word = array_diff($word, $stopwords);
$word = array_values($word);
$tf_idf     = 0;
```

```
$freq_word = array_count_values($word);
foreach ($freq_word as $token => $tf)
$tf_idf += $tf * $inv_index[$token]['idf'];
array_push($sentence_weight, $tf_idf);}
arsort($sentence_weight);
$sorted = array_slice($sentence_weight, 0, $cmpr, true);
ksort($sorted);
$summary = "";
foreach ($sorted as $key => $value)
$summary = $summary.$sentence[$key].". ";
$output = array();
$output['original'] = $iberita;
$output['summary']  = $summary;
return $output;}
?>
```

 ISSN 1829-5266 (paper), ISSN 2301-8550 (online)