ISSN 2007-9737

Lexical Patterns Based on Maximal Frequent Secuences for Automatic Keyphrase Extraction

Yanet Hernández Casimiro, Yulia Ledeneva, René Arnulfo García Hernández,
Marco Antonio Ramos Corchado

Universidad Autónoma del Estado de México,
Mexico

{yhernandezcs, marco.corchado}@gmail.com, yledeneva@yahoo.com,
renearnulfo@hotmail.com

Abstract. This paper presents a method for the automatic keyphrase extraction task using lexical patterns. First, the patterns are obtained from a set of data and converted into regular expression search patterns, allowing to consider sequences of characters that define a phrase without depending on its syntactic or semantic characteristics and thus obtain a list of possible candidates. Besides, to select the best, only those that obtained a high weight will be considered, in the following four weights: Boolean (B), Precision (P), Recall (R), and F-Measure (F); which corresponds to the result obtained from each evaluated pattern, therefore a list is generating of the best 5,10 and 15 keyphrases for each document. The evaluation of the method was realized by length (L) and combination (C), where the combination takes the best candidates for each length (1 to 4). The method was tested in corpus of scientific articles using the SemEval-2010 data set for task 5.

Keywords. Lexical patterns, key phrases, automatic key phrase extraction, maximal frequent sequences.

1 Introduction

At present, the number of digital documents has exponentially grown due to the constant use of the computer, resulting in the need for information, which has become a primary activity for any aspect of life. However, not all the information is important, so it is necessary to identify it. This activity is difficult for a user to perform, since in each search for information an endless number of documents appear that must be reviewed, hence the importance of having a summarized form of the document.

One way to solve this problem is through keyphrases, words, or terms [4, 6, 19, 31], which provide a brief way to describe the main content of the document and thus help the user to decide if that information is adequate for him [2, 3, 15, 24]. In this paper, we will refer to them as keyphrases.

Most of the contents of the web do not have phrases that represent the web. However, it is common to see them in scientific articles and news pages, where authors are requested to provide a list of phrases for representation [36]. The low use of these is because it is a time-consuming activity; in some cases, it is necessary to have experts with knowledge in a specific domain. Therefore, efforts have been made to find keyphrases that represent the main content of a document [6, 29, 36]. This activity is called keyphrase extraction, which is a process related to human cognition, so that the field of computational sciences has seen the need to carry out investigations that can model this process, through techniques thus creating the task of automatic extraction of keyphrases. This task generates phrases from a source document, instead of using thesauri as a resource (keyphrase assignment) [22, 29, 32].

Usually, the task is divided into two phases: identification of phrases that can serve as candidates and selection, where the importance of the phrase is determined by scoring [22, 40].

The extracted phrases are not only used to represent the content of a document [10]. Furthermore, they are useful for other tasks such as indexing [21], grouping [4], classification [20], abstract generation [33], and opinion mining [7].

doi: 10.13053/CyS-25-1-3868

One of the characteristics that years ago was considered relevant for the keyphrase extraction task was frequency [29]. However, this criterion is not the main one, since, in a text, there can be very repetitive phrases, but it does not mean that they are keys [37].

In [11], a method based on the generation of lexical patterns is described, making use of frequent maximal sequences [14] that allow obtaining keyphrases from a text, regardless of their frequency, resulting in a list of candidate phrases, which will later be selected through a ranking, considering the ones with the highest weight as the best.

2 Related Works

Most of the papers that study the extraction of keyphrases use data sets, mostly in English [3, 4, 22, 40], also in French [3, 9], Croatian [2] and Chinese [41]. There are corpus for different domains such as scientific [16, 18, 22, 40], emails [17], social networks [1, 40], among others.

However, they are used to test and evaluate supervised and unsupervised automatic keyphrase extraction methods. One of the most important supervised works within the keyphrase extraction study is that carried out by Turney [35], who implements a machine learning system consisting of two sections; a genetic algorithm called Genex and the extractor, which consists of twelve parameters, which are adjusted by the genetic algorithm, to later generate a list of keyphrases.

In addition, other methods use external resources that are dependent on the language, such as the system called SEERLAB [34] that integrates three components for the identification of candidate phrases; parser, extractor, and candidate classifier, where it trains a Random Forestar classifier for keyphrase selection.

Phrases can be identified using sequential patterns, which calculate the semantic relationship between the words in a document [39].

On the other hand, there are unsupervised methods; one of the most basic is TF [28], which is used as a baseline to determine the acceptable value that a system can have the task [16].

It is considered a characteristic in some works, where the task is considered as classification problem [5, 17, 30].

Although there are unsupervised methods such as TF, there are also more robust ones, such as SemCluster [4]. First, n-grams and named entities are obtained to use knowledge bases and identify semantic relationships with other terms. Additionally, it uses a grouping model to systematically identify and filter unimportant groups of phrases, allowing only those representing the document to be scored as candidate keyphrases.

In [25], two techniques are used to extract keyphrases: maximal frequent sequences and PageRank. These are used to obtain a limited number of text fragments called n-grams (uni-grams, bi-grams, and tri-grams). Subsequently, a weight is assigned to the sequences using PageRank, and the ones with the highest value are chosen as candidates.

Maximal Frequent Sequences (MFS) are essential in pattern mining, and these are not a subsequence of any other frequent sequence but rather a compact representation of the entire set of frequent sequences [14]. They have been applied to different works as in [11], where a method for detecting text fragments as a candidate for hyperlinking is presented. First, a set of lexical patterns is obtained, called that way because they work at the lexical level without considering syntactic or semantic aspects. The text should be normalized from examples of human-generated candidate fragments (seeds). These are tagged as <*Link*>. They must also contain right and left context, limited to 20 words that help to delimit the beginning and the end of a text fragment that contains a hyperlink.

The text then serves as input to the MFS algorithm that derives the lexical patterns. Sequences are essential since the sequential order of the word is essential to obtain a good pattern. These patterns become search patterns that are applied to a set of plain text to obtain those candidates.

The patterns have been useful in this task, and in [13], he applies this same technique and process to answer definition questions.

Table 1. Examples of special characters and normalized elements

Character	Name	Normalization
(	Parenthesis that opens	@PaA
:	Two points	@DP
{	Braces that open	@LLA
<	greater-than	@MAY
\	Right diagonal	@DIAD
´	Apostrophe	@APOS
1	One	@UNO
2	Two	@DOS

However, in [26], he also uses this technique for the information extraction task. These three works have been tested with the Spanish language.

In this paper, we propose an unsupervised method for automatic keyphrase extraction, which consists of two essential sections; the first is the discovery of lexical patterns, and the second is the selection of keyphrases through a series of weighing derived from the evaluation of each pattern.

Our method is compared to the following novel unsupervised approaches: Yake [12], Topic Rank [9], Single Rank [38], and Text Rank [23], which are implemented using the PKE library [8].

Yake is a method based on the extraction of statistical characteristics from the text, such as term co-occurrence; an important aspect is that not based on the frequency of terms, which means that conditions are not established for the minimum of frequency that a keyword must have to be a candidate, as is the case with our proposed method [12].

TopicRank [9] is a graph-based method based on the grouping of sentences as topics, calculating weights as a function of the distances between positions. This method relies on identifying noun phrases, while in our method this is not a determiner. SingleRank [38] is graph-based, which builds a local graph for each unique document. Also, it calculates based on the co-occurrence relationship between two words that express cohesion relationships.

For our method, the co-occurrence in the MFS and the generation of lexical patterns, is important. Finally, TextRank [23] is a graph-based model, where the units to be classified are sequences of one or more lexical units extracted from the text, and these represent the vertices added to the graph. In this way, we can say that both this method and the proposed one take words as lexical units. The rest of the document is organized as follows: Section 3 describes the proposed method for obtaining the keyphrases. Section 4 is dedicated to experimentation and the description of obtained results. Finally, in section 5, we give the conclusion of the work.

3 Proposed Method

In general, the proposed method is based on six important phases for the Automatic Keyphrase Extraction task, where MFSs are used for the lexical pattern discovery process, which will assist in generating a candidate keyphrase list.

3.1 Pre-Processing

The first phase of this method is cleaning, restructuring, and coding of the data. This step is carried out because the MFS module does not accept special characters. Therefore, all symbols other than punctuation marks and numbers are eliminated since they do not provide relevant information.

Afterward, the accepted symbols are normalized, transforming them into labels using regular expressions, which allow their identification; URLs, emails, and dates were labeled. Finally, the sentences' lemmatization is applied using Porter's stemming algorithm [27] for each data set document.

Table 1 shows some examples of the normalization of special characters.

3.2 Data Construction and Preparation

In this phase, from a set of data $\boldsymbol{D} = \{\boldsymbol{d_1}, \boldsymbol{d_2}, \boldsymbol{d_3} \ldots \boldsymbol{d_j}\}$, three more are created. The first set $\boldsymbol{D_{format}}$ is obtained from having applied phase 3.1 in D and will be used in 3.3.

Table 2. Example of normalization of keyphrases by length (1-3)

Long l	**Keyphrase**	**Normalization**
l^1	*…on* ***uddi*** *author…*	*…on<KP> author…*
l^2	*…the* ***scalabl issu*** *of…*	*…the<KP><KP>of…*
l^3	*…Ani* ***grid servic discoveri*** *mechan…*	*…Ani<KP><KP><KP> mechan…*

Table 3. Examples of lexical patterns according to their length

$\boldsymbol{l}$	**Lexical Patterns**
l^1	$Of < KP >.$
l^2	$Of < KP >< KP > ,$
l^3	$The\ < KP >< KP >< KP >.$
l^4	$A < KP >< KP >< KP >< KP >,$

Table 4. Example of converting lexical patterns to search patterns by length

Long l	**Lexical patterns**	**Search pattern**
l^1	Of **<KP>** .	\s+Of\s+(\w+)\s+\.\s+
l^2	The **<KP> <KP>** of the	\s+The\s+(\w+)\s+(\w+)\s+of\s+the\s+
l^3	A **<KP> <KP> <KP>** (	\s+A\s+(\w+)\s+(\w+)\s+(\w+)\s+\(\s+

$\boldsymbol{D_{search}}\ y\ \boldsymbol{D_{context}}$ begin from D and a set of keyphrases generated by an expert $\boldsymbol{KP}_{expert}$**.** These are divided by length $\boldsymbol{l}$**,** in such a way that:

$$\boldsymbol{KP_{expert}{}^{l}_{d_j}}.$$

$KP_{\boldsymbol{expert}}{}^{l}_{d_j}$ must be transformed into search keyphrases $\boldsymbol{KP_{search}{}^{l}_{d_j}}$, using regular expressions, which will later be identified in each D document.

$\boldsymbol{D_{search}}$s obtained by generating a list of keyphrases extracted with $\boldsymbol{KP_{search}{}^{l}_{d_j}}$.

Whereas for $\boldsymbol{D_{context}}$, the keyphrases identified with $\boldsymbol{KP_{search}{}^{l}_{d_j}}$ are normalized $\boldsymbol{KP}_{seed}{}^{\boldsymbol{l}}_{\boldsymbol{k\ d_j}}$, according to a tag "<KP>", which is assigned to each word that constitutes a keyphrase.

Furthermore, for each $\boldsymbol{kp}_{seed}{}^{\boldsymbol{l}}_{\boldsymbol{k\ d_j}}$ sequence of characters must be considered, which we name right and left "context" consisting of 20 words each; these will be placed in a list according to the sentence's length [11, 13, 25].

This set will be required for the lexical pattern discovery phase $\boldsymbol{P}_{\boldsymbol{l}\mathrm{ex}}$.

Table 2 shows some examples of the normalization of $\boldsymbol{KP_{seed}{}^{l}_{d_j}}$, which have already been pre-processed according to phase 3.1.

3.3 Lexical Pattern Discovery

In this phase, the extraction of MFS [14] is applied in $D_{context}$, to obtain $\boldsymbol{P}_{\boldsymbol{l}\mathrm{ex}}{}^{l}$, of which only those that meet the following structure are chosen:

$$context_{izq}/ < KP >/context_{der}.$$

From this phase, a series of patterns is obtained, whose discovery depends on a threshold β, where the percentage of this depends on the amount of $\boldsymbol{KP_{seed}{}^{l}_{d_j}}$ that have been identified.

For example, if the amount of $\boldsymbol{KP_{seed}{}^{l}_{d_j}}$, β at 1% would be 13.5≈14, this threshold represents the number of occurrences that an MFS must present to consider it as a lexical pattern; this parameter is assigned to the MFS algorithm. The following table shows some examples of lexical patterns found by length.

3.4 Identification of Candidate Keyphrases

The $P_{lex}{}^{l}$ is converted to $P_{search}{}^{l}$ by regular expressions; then, we can say that $P_{lex}{}^{l} = P_{search}{}^{l}$, but the latter are transformed and applied to the D_{format} to identify keyphrases $\boldsymbol{KP_{ident_{w\,d_j}^{l}}}$.

It is important to note that $p_{search}{}^{l}_{w}$, can identify $\boldsymbol{kp_{ident_{w\,d_j}^{l}}}$ more than once because the lexical information can be repeated. For example, the lexical pattern ***“\s+Of\s+(\w+)\s+\.\s+”*** can find the word ***"DHT"*** more than once in the text since the lexical structure that identifies it is the form ***“OF {DHT}.”*** and the word can be presented that way more than once. However, the method is not based on the frequency of the word, it is simply considered as an identified phrase once $p_{search}{}^{l}_{w}$ finds it. Table 4 shows examples of the transformation from a lexical pattern to a search pattern.

3.5 Lexical Pattern Evaluation

To evaluate the performance of $P_{search}{}^{l}$, it is necessary $\boldsymbol{D_{search}}$, which is considered a gold standard set, and $\boldsymbol{KP_{ident_{w\,d_j}^{l}}}$, which is the set of phrases identified by the system. The evaluation is carried out for each $p_{search}{}^{l}_{w}$ using Precision measures (P), Recall (R), and F-Measure (F-M) [11].

3.6 Selection and Evaluation of Keyphrases

The selection is made from $KP_{ident_{w\,d_j}^{l}}$, in which $kp_{ident_{w\,d_j}^{l}}$ is assigned as weight the value of Precision, Recuerdo, and F-Measure of the $p_{search}{}^{l}_{m}$ by which it was extracted, this value is taken from phase 3.5. The formulas used to obtain the value of each $p_{search}{}^{l}_{w}$ are shown below:

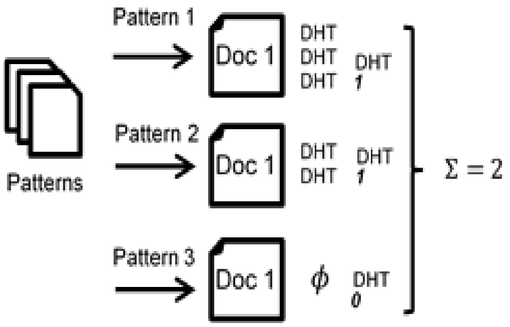

Fig. 1. Boolean weighting assignment process

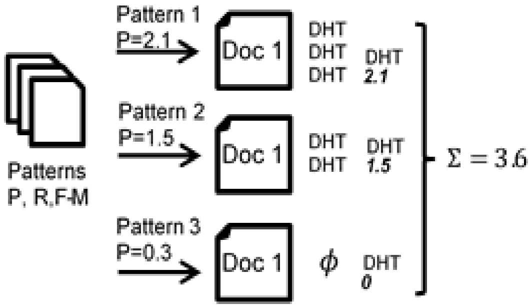

Fig. 2. Precision weighing assignment process

Table 5. Combinations of n-grams

Combination (C)	n-gramas
Combination 1 (C1)	$\mathrm{l}^{1}_{[1,5]}, \mathrm{l}^{2}_{[1,10]}$
Combination 2 (C2)	$\mathrm{l}^{1}_{[1,4]}, \mathrm{l}^{2}_{[1,9]}, \mathrm{l}^{3}_{[1,2]}$
Combination 3 (C3)	$\mathrm{l}^{1}_{[1,3]}, \mathrm{l}^{2}_{[1,9]}, \mathrm{l}^{3}_{[1,3]}$
Combination 4 (C4)	$\mathrm{l}^{1}_{[1,4]}, \mathrm{l}^{2}_{[1,9]}, \mathrm{l}^{3}_{[1,3]}, \mathrm{l}^{4}_{1}$
Combination 5 (C5)	$\mathrm{l}^{1}_{[1,5]}, \mathrm{l}^{2}_{[1,10]}, \mathrm{l}^{3}_{[1,2]}$
Combination 6 (C6)	$\mathrm{l}^{1}_{[1,4]}, \mathrm{l}^{2}_{[1,8]}, \mathrm{l}^{3}_{[1,3]}$
Combination 7 (C7)	$\mathrm{l}^{1}_{[1,2]}, \mathrm{l}^{2}_{[1,11]}, \mathrm{l}^{3}_{[1,3]}$

$$P\left(p_{search}{}^{l}_{m}, d_j\right) = \frac{KP_{ident_{w\,d_j}^{l}} \cap KP_{expert_{w\,d_j}^{l}} \in D}{/KP_{ident_{w\,d_j}^{l}} \in D/}, \qquad (1)$$

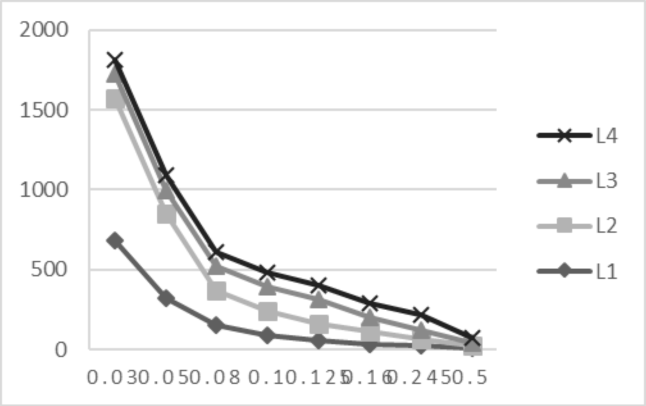

Fig. 2. Number of patterns extracted per threshold

Table 6. Performance of the proposed method for different thresholds, combinations, and weights

Proposed Method	**Top 15**								
Threshold	**Author**			**Reader**			**Combined**		
	P	**R**	**F-M**	**P**	**R**	**F-M**	**P**	**R**	**F-M**
$\beta = 0.1$ C2 B	6.00%	23.26%	9.54%	14.67%	18.27%	16.27%	17.27%	17.67%	17.47%
$\beta = 0.08$ C2 B	6.07%	23.51%	9.65%	14.00%	17.44%	15.53%	16.73%	17.12%	16.92%
$\beta = 0.03$ C2 P	6.07%	23.51%	9.65%	13.80%	17.19%	15.31%	16.60%	16.98%	16.79%
$\beta = 0.05$ C2 P	6.00%	23.26%	9.54%	13.67%	17.03%	15.17%	16.47%	16.85%	16.66%
$\beta = 0.125$ C2 P	5.80%	22.48%	9.22%	13.33%	16.61%	14.79%	16.00%	16.37%	16.18%
$\beta = 0.16$ C2 P	5.47%	21.19%	8.70%	12.07%	15.03%	13.39%	14.60%	14.94%	14.77%
$\beta = 0.254$ C6 P	5.67%	21.96%	9.01%	11.80%	14.70%	13.09%	14.53%	14.87%	14.70%
$\beta = 0.5$ C4 P	4.67%	18.09%	7.42%	11.20%	13.95%	12.42%	13.27%	13.57%	13.42%

$$R\left(p_{search}{}_{m}^{l}, d_j\right) = \frac{KP_{ident}{}_{w\,d_j}^{l} \cap KP_{expert}{}_{w\,d_j}^{l} \in D}{/KP_{ident}{}_{w\,d_j}^{l}/}\ , \tag{2}$$

$$F - M\left(p_{search}{}_{m}^{l}, d_j\right) = \frac{2 \cdot P \cdot R}{P + R}\ . \tag{3}$$

In addition to the previous weights, a Boolean weight was added, where the value of 1, always corresponds to $kp_{ident}{}_{w\,d_j}^{l}$ extracts $p_{search}{}_{m}^{l}$ otherwise the value will ϕ, such that if $kp_{ident}{}_{w\,d_j}^{l}$ is identified by itself $p_{search}{}_{m}^{l}$ more than once, it will be considered as an occurrence. Figure 1 shows the process for assigning the Boolean weighing:

$$B\left(p_{search}{}_{m}^{l}, d_j\right) = \sum\nolimits_{k}^{n} F\left(p_{search}{}_{m}^{l}, d_j\ \right), \tag{4}$$

where:

$$F\left(p_{search}{}_{m}^{l}, d_j\ \right) = \begin{cases} 0 & p_{search}{}_{m}^{l}(d_j) = \phi, \\ 1 & p_{search}{}_{m}^{l}(d_j) \neq\ \phi. \end{cases}$$

Table 7. Patterns that appear in different β for the identification of keyphrases of l^1

Patron	**0.1**	**0.08**	**0.03**	**0.254**	**0.05**	**0.08**	**0.16**	**0.125**
Of $< \mathrm{KP} >$.	X			X		X	X	X
The $< \mathrm{KP} >$ is	X			X		X	X	X
Of $< \mathrm{KP} >$,	X			X			X	X
The $< \mathrm{KP} >$ of the	X	X		X			X	X
Of $< \mathrm{KP} >$ and	X	X		X	X		X	X
($< \mathrm{KP} >$)	X		X	X	X	X	X	X
And $< \mathrm{KP} >$.	X	X		X			X	X
The $< \mathrm{KP} >$ of a	X	X		X	X	X	X	X
Of $< \mathrm{KP} >$ is	X	X	X	X	X	X	X	X
The $< \mathrm{KP} >$ to	X	X		X		X	X	X

Table 8. Performance of the proposed method, considering the 5 best keyphrases of the author, reader, and combined sets

Proposed method	**Top 5**								
Threshold	**Author**			**Reader**			**Combined**		
	P	**R**	**F-M**	**P**	**R**	**F-M**	**P**	**R**	**F-M**
$\beta = 0.1$ C2 B	8.00%	10.34%	9.02%	16.00%	6.64%	9.39%	19.60%	6.68%	9.96%
Yake [12]	9.40%	12.14%	10.60%	14.40%	5.98%	8.45%	19.00%	6.48%	9.66%
Topic Rank [9]	8.00%	10.34%	9.02%	5.60%	6.48%	9.16%	18.40%	6.28%	9.36%
Single Rank [38]	0.80%	1.03%	0.90%	2.00%	0.83%	1.17%	2.60%	0.89%	1.33%
Text Rank [23]	0.20%	0.26%	0.23%	0.80%	0.33%	0.47%	0.80%	0.27%	0.40%

After assigning the weights, if the $kp_{ident}{}^{l}_{w\,d_j}$ is extracted by $P_{search}{}^{l}$, then the sum of each of the weights must be performed, considering the quantity of $p_{search}{}^{l}_{m}$ by which it was identified $kp_{ident}{}^{l}_{w\,d_j}$.

$$Weight\left(kp_{ident}{}^{l}_{w\,d_j}\right) = \sum_{k=1}^{n} W\left(p_{search}{}^{l}_{m} | kp_{ident_w} \in p_{search}{}^{l}_{m}(d_j)\right). \quad (5)$$

Figure 2 shows the process for assigning precision weighing of $kp_{ident}{}^{l}_{w\,d_j}$ with respect to the value of each $p_{search}{}^{l}_{m}$.

As mentioned in the boolean weighting, if $kp_{ident}{}^{l}_{w\,d_j}$ is obtained by the same $p_{search}{}^{l}_{m}$ more than once, it will be considered as an occurrence.

In such a way that to identify the keyphrases, it does not depend on the frequency that

Table 9. Performance of the proposed method, considering the 10 best keyphrases of the author, reader, and combined sets

Proposed method	**Top 10**								
Threshold	**Author**			**Reader**			**Combined**		
	P	**R**	**F-M**	**P**	**R**	**F-M**	**P**	**R**	**F-M**
$\beta = 0.1$ *C2 B*	8.00%	20.67%	11.54%	17.60%	14.62%	15.97%	20.90%	14.26%	16.95%
Yake [12]	7.40%	19.12%	10.67%	13.50%	11.21%	12.25%	17.00%	11.60%	13.79%
Topic Rank [9]	5.90%	15.25%	8.51%	12.50%	10.38%	11.34%	14.70%	10.03%	11.92%
Single Rank [38]	0.80%	2.07%	1.15%	2.80%	2.33%	2.54%	0.20%	2.18%	2.59%
Text Rank [23]	0.70%	1.81%	1.01%	1.70%	1.41%	1.54%	2.00%	1.36%	1.62%

Table 10. Performance of the proposed method, considering the 15 best keyphrases of the author, reader, and combined sets

Proposed method	**Top 15**								
Threshold	**Author**			**Reader**			**Combined**		
	P	**R**	**F-M**	**P**	**R**	**F-M**	**P**	**R**	**F-M**
$\beta = 0.1$ *C2 B*	6.00%	23.26%	9.54%	14.67%	18.27%	16.27%	17.27%	17.67%	17.47%
Yake [12]	6.40%	24.81%	10.18%	11.47%	14.29%	12.73%	14.67%	15.01%	14.84%
Topic Rank [9]	5.20%	20.16%	8.27%	10.53%	13.12%	11.68%	12.80%	13.10%	12.95%
Single Rank [38]	1.07%	4.13%	1.70%	2.87%	3.57%	3.18%	3.47%	3.55%	3.51%
Text Rank [23]	0.73%	2.84%	1.16%	1.80%	2.24%	2.00%	2.20%	2.25%	2.22%

$kp_{ident}{}_{w\,d_j}^{l}$ has in the document, but on the number of patterns by which it is extracted.

From this phase, we obtain $KP_{candidate}{}_{w\,d_j}^{l}$, where the top 5, 10 and 15 with the best score for each document are considered. In addition, seven phrase combinations were created according to l^n (1-4 grams) as shown in table 5.

Finally, $KP_{candidate}{}_{w\,d_j}^{l}$ is evaluated using the measures of Precision (P), Recall (R) and F-Measure (FM).

4 Experimentation

4.1 Data

This article uses the SemEval-2010 dataset for task # 5. It has 284 scientific articles, of which 100 are for tests, 144 for training, and 40 for validation. It also contains a set of gold standard keyphrases, assigned by author, reader, and combined (assigned by author and reader) for each article. The reader phrases were generated by 50 student annotators from the University of Singapore, where the main objective was to obtain keyphrases from any section of the document. However, the indication was not fully considered, thus having 15% of keyphrases assigned by the reader and 19% assigned by author that do not appear in the document; due to this, the maximum memory that the participating systems could have reached it was 81% and 85% [16].

4.2 Results

Our method was tested with eight thresholds, of which n number of patterns were generated according to a β threshold. Figure 2 shows the number of patterns obtained per threshold.

As can be seen in the figure, the lower the threshold, the greater the number of patterns that can be obtained. However, these patterns are already beginning to depend on the context and domain of the documents, which is why it was considered that the best threshold is not the one with the greatest number of patterns, but the one with the greatest number. The best was $\beta = 0.1$ for the top 15. Table 6 shows the results obtained for the keyphrases by author, reader, and combined, referring to the best performance for each threshold and combination, as well as weighing.

The results show that the best keyphrases are obtained from weighing B with threshold $\beta = 0.1$ and $C2$. However, it can be observed that the results have a variation concerning β due to the amount and type of patterns discovered.

This is because the patterns are sequences of characters with a percentage of the frequency with respect to β, as shown in section 3.3. That is why in some β patterns can coincide, as shown in Table 7, where examples of patterns are given to identify the sentences of l^1. If a pattern is found in more than one β as is the case in the examples "(<KP>)" and "OF <KP> IS", we can consider these patterns to be relevant for the identification of keyphrases.

Table 8 shows the results obtained from our method for the top 5 of the sets, author, reader, and combined, and we compare them with four unsupervised methods.

It can be seen that the proposed method has an F-M value of 9.96% for the top 5 in the set of combined phrases, which places us in the first position of this ranking.

The methods with which we compare were implemented from the PKE toolkit [8], which is developed in Python for the automatic extraction of keyphrases, each of the methods consists of different parameters. For Yake, stoplist = 'english', selection n = 3, window = 2, threshold = 0.8 and extraction n = 15 were used. Topic Rank; pos = {'NOUN', 'PROPN', 'ADJ'}, stoplist = 'english', method = 'average', threshold = 0.7 and extraction n = 15. Single Rank; pos = {'NOUN', 'PROPN', 'ADJ'}, window = 10, normalization = "stemming" and extraction n = 15. Lastly, TextRank; pos = {'NOUN', 'PROPN', 'ADJ'}, window = 2, normalization = "stemming", top_percent = 0.33 and extraction n = 15.

Tables 9 and 10 show the top 10 and 15, where our method continues to retain the first position.

5 Conclusions

This paper presents a method for extracting keyphrases using lexical patterns. The results prove that this proposed method and the weights used to identify keyphrases are useful for the task. We can also see that the best weighting was boolean, which considers a positive or negative value when the phrase is or is not detected by a certain search pattern.

However, the importance of the threshold with which the patterns are extracted must be considered, since the lower β the patterns are content dependent, while, being smaller, they are more general and only consist of punctuation marks and stopwords.

In addition, in the same way, it can be considered that the keyphrases identified by our proposed method do not depend on the frequency of the word, but on the search pattern that identifies it, therefore we can conclude that a keyphrase should not only be a frequent word in the text, but it can also appear only once and be considered highly relevant.

Finally, through our method, we discover that a pattern can appear in more than one threshold, which we consider important since we can mention that the pattern's lexical structure is constant, and it is more likely that a keyword can be contained in these contexts lexicons.

As a future work, we test lexical functions [42] and content-based characteristics [43].

References

1. **Abilhoa, W.D., de Castro, L.N. (2014).** A keyword extraction method from twitter messages represented as graphs. Applied Mathematics and Computation, Vol. 240, pp. 308–325. DOI: 10.1016/ j.amc.2014.04.090.
2. **Ahel, R., Dalbelo, B., Šnajder, J. (2009).** Automatic Keyphrase Extraction from Croatian Newspaper Articles. The Future of Information Sciences, Digital Resources and Knowledge Sharing, pp. 207–218.
3. **Ali, C.B., Wang, R., Haddad, H. (2015).** A Two-Level Keyphrase Extraction Approach. **Gelbukh, A. (ed.)** Computational Linguistics and Intelligent Text Processing, pp. 390–401.
4. **Alrehamy, H., Walker, C. (2018).** Exploiting extensible background knowledge for clustering-based automatic keyphrase extraction. Soft Computing, Vol. 22, No. 21, pp. 7041–7057. DOI: 10.1007/s00500-018-3414-4.
5. **Awoyelu, I.O., Abimbola, R.O., Olaniran, A.T., Amoo, A.O., Mabude, C.N. (2016).** Performance Evaluation of an Improved Model for Keyphrase Extraction in Documents. Computer Science and Information Technology, Vol. 4, No. 1, pp. 33–43. DOI: 10.13189/csit.2016.040106.
6. **Beliga, S., Martinčić-Ipšić, S. (2017).** Network-Enabled Keyword Extraction for Under-Resourced Languages. **Calì, A., Gorgan, D., Ugarte, M. (Eds.)**, Semantic Keyword-Based Search on Structured Data Sources pp. 124–135.
7. **Berend, G. (2011).** Opinion Expression Mining by Exploiting Keyphrase Extraction, pp. 1162–1170.
8. **Boudin, F. (2016).** PKE: an open source python-based keyphrase extraction toolkit. Proceedings of COLING´16, the 26th International Conference on Computational Linguistics: System Demonstrations, pp. 69–73.
9. **Bougouin, A., Boudin, F., Daille, B. (2016).** Keyphrase Annotation with Graph Co-Ranking. Proceedings of COLING´16, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 2945–2955.
10. **Bougouin, A., Boudin, F., Daille, B. (2013).** Topicrank: Graph-based topic ranking for keyphrase extraction. International Joint Conference on Natural Language Processing, pp. 543–551.
11. **Camacho, M. (2015).** Detección de fragmentos de texto como candidato a hipervínculo. Universidad Autónoma del Estado de México.
12. **Campos, R., Mangaravite, V., Pasquali, A., Jorge, A., Nunes, C., Jatowt, A. (2020).** YAKE! Keyword extraction from single documents using multiple local features. Information Sciences, Vol. 509, pp. 257–289.
13. **Denicia, M.C. (2007).** Respondiendo Preguntas de Definición mediante el Descubrimiento de Patrones Léxicos. Instituto Nacional de Astrofísica, Óptica y Electrónica.
14. **García-Hernández, R.A., Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A. (2006).** A New Algorithm for Fast Discovery of Maximal Sequential Patterns in a Document Collection. **Gelbukh, A. (Ed.)**, Computational Linguistics and Intelligent Text Processing, pp. 514–523.
15. **Kathait, S.S., Tiwari, S., Varshney, A., Sharma, A. (2017).** Unsupervised Key-phrase Extraction using Noun Phrases.
16. **Kim, S.N., Medelyan, O., Kan, M.Y., Baldwin, T. (2010).** SemEval´10 Task 5 : Automatic Keyphrase Extraction from Scientific Articles. pp. 21–26.
17. **Lahiri, S., Mihalcea, R., Lai, P.H. (2017).** Keyword extraction from emails. Natural Language Engineering, Vol. 23, No. 2, pp. 295–317. DOI: 10.1017/S1351324916000231.
18. **Lee, L.H., Lee, K.C., Tseng, Y.H. (2017).** The NTNU System at SemEval-2017 Task 10: Extracting Keyphrases and Relations from Scientific Publications Using Multiple Conditional Random Fields. Proceedings of the 11th International Workshop on Semantic Evaluation, pp. 951–955. DOI: 10.18653/v1/S17-2165.
19. **Lin, H., Yang, C., Lee, H., Lee, L. (2018).** Domain Independent Key Term Extraction from Spoken Content Based on Context and Term Location Information in the Utterances. International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6044–6048. DOI: 10.1109/ICASSP.2018.8462252.
20. **Liu, S., Shen, F., Chaudhary, V., Liu, H. (2017).** MayoNLP at SemEval´17 Task 10: Word Embedding Distance Pattern for Keyphrase Classification in Scientific Publications. Proceedings of the 11th International Workshop on Semantic Evaluation, pp. 956–960. DOI: 10.18653/v1/S17-2166
21. **Medelyan, O., Frank, E., Witten, I.H. (2009).** Human-competitive Tagging Using Automatic Keyphrase Extraction. Proceedings of the Conference on Empirical Methods in Natural Language, Vol. 3, pp. 1318–1327.

22. **Meng, R., Zhao, S., Han, S., He, D., Brusilovsky, P., Chi, Y. (2017).** Deep Keyphrase Generation. Vol. 1, pp. 582–592. DOI:10.18653/v1/P17-1054.
23. **Mihalcea, R., Tarau, P. (2004).** Textrank: Bringing order into text. Proceedings of the Conference on Empirical Methods In Natural Language, pp. 404– 411.
24. **Najadat, H.M., Hmeidi, I.I., Al-Kabi, M.N., Issa, M. M. B. (2016).** Automatic Keyphrase Extractor from Arabic Documents. International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 7, No. 2. DOI: 10.14569/IJACSA. 2016. 070226.
25. **Ortiz, R., Pinto, D., Tovar, M., Jiménez-Salazar, H. (2010).** BUAP: An Unsupervised Approach to Automatic Keyphrase Extraction from Scientific Articles. Procceding of the 5th International Workshop on Semantic Evaluation, pp.174–177.
26. **Palacios, C.P.O. (2008).** Métodos Basados en Patrones Léxicos para la Extracción de Información. Instituto Nacional de Astrofísica, Óptica y Electrónica.
27. **Porter, M.F. (1980).** An algorithm for suffix stripping. Program, Vol. 14, pp. 130–137. DOI: 10.1108/00330330610681286.
28. **Salton, G., Buckley, C. (1988).** Term-weighting approaches in automatic text retrieval. Information Processing & Management, Vol. 24, No. 5, pp. 513–523. DOI: 10.1016/0306-4573(88)90021-0.
29. **Sarkar, K., Nasipuri, M., Ghose, S. (2010).** A New Approach to Keyphrase Extraction Using Neural Networks. IJCSI International Journal of Computer Science Issues.
30. **Siddiqi, S., Sharan, A. (2015).** Keyword and Keyphrase Extraction Techniques: A Literature Review. International Journal of Computer Applications, Vol. 109, No. 2, pp.18–23.
31. **Siddiqi, S., Sharan, A. (2016).** Keyword extraction from single documents using mean word intermediate distance. International Journal of Advanced Computer Research. DOI: 10.19101/ ijacr.2016.625003
32. **Sterckx, L., Demeester, T., Deleu, J., Develder, C. (2018).** Creation and evaluation of large keyphrase extraction collections with multiple opinions. Language Resources and Evaluation, Vol. 52, No. 2, pp. 503–532. DOI: 10.1007/s10579-017-9395-6.
33. **Thomas, J.R., Bharti, S.K., Babu, K.S. (2016).** Automatic Keyword Extraction for Text Summarization in e-Newspapers. Proceedings of the International Conference on Informatics and Analytics, DOI: 10.1145/2980258.2980442.
34. **Treeratpituk, P., Teregowda, P., Huang, J., Giles, C.L. (2010).** SEERLAB: A System for Extracting Keyphrases from Scholarly Documents. pp. 182– 185.
35. **Turney, P. (1999).** Learning to Extract Keyphrases from Text. National Research Council, 45.
36. **Turney, P.D. (2003).** Coherent Keyphrase Extraction via Web Mining. Proceedings of the 18th International Joint Conference on Artificial Intelligence, pp. 434–439.
37. **Ulyanova, U., Petrochenko, L. (2017).** Key Words in the Missing Manual: the Problem of Identification. Vestnik Volgogradskogo gosudarstvennogo universiteta. Serija 2. Jazykoznanije, Vol. 16, No. 2, pp. 68–81. DOI: 10.15688/jvolsu2.2017.2.7
38. **Wan, X., Xiao, J. (2008).** CollabRank: Towards a Collaborative Approach to Single-Document Keyphrase Extraction. Proceedings of the 22nd International Conference on Computational Linguistics, pp. 969–976.
39. **Xie, F., Wu, X., Zhu, X. (2014).** Document-Specific Keyphrase Extraction Using Sequential Patterns with Wildcards. IEEE International Conference on Data Mining, pp. 1055–1060. DOI: 10.1109/ICDM. 2014.105.
40. **Ying, Y., Qingping, T., Qinzheng, X., Ping, Z., Panpan, L. (2017).** A Graph-based Approach of Automatic Keyphrase Extraction. Procedia Computer Science, Vol. 107, pp. 248–255. DOI: 10.1016/j.procs.2017.03.087.
41. **Zhang, C. (2008).** Automatic Keyword Extraction from Documents Using Conditional Random Fields.
42. **Kolesnikova, O. (2020).** Automatic Detection of Lexical Funations in Context. Computación y sistemas, Vol. 24, No. 3, pp. 1337–1352.
43. **Ameer, I., Ashraf, N., Sidorov, G., Goméz, H. (2020).** Multi-label Emotion Classification using Content-Based Features in Twitter. Computación y Sistemas, Vol. 24, No. 3, pp. 1159–1164.

Article received on 03/10/2020 accepted on 23/11/2020.
Corresponding author is Yulia Ledeneva.