ISSN 1846-6168
UDK 62

# THE ANALYSIS OF KEYWORD OCCURRENCES WITHIN SPECIFIC PARTS OF MULTIPLE ARTICLES — THE CONCEPT AND THE FIRST IMPLEMENTATION

# ANALIZA POJAVLJIVANJA KLJUČNIH RIJEČI U SPECIFIČNIM DIJELOVIMA ČLANAKA — KONCEPT I PRVA IMPLEMENTACIJA

***Tomislav Horvat, Ladislav Havaš, Robert Logožar***

Original scientific paper

***Abstract.*** *The authors present the concept and implementation of the keyword occurrence analysis that is based on the separate counting of (key)words in the four main parts of scientific and other types of articles: the title—followed by a list of authors, abstract, list of keywords, and the main text. Also, the analysis is meant to be applied to more than one article at once. As such, it can serve the researchers, writers, editors, and others for the reviewing and statistical exploration of the articles. For discrimination of the article parts, the already existing, implicit tagging must be minimally assisted by the user. That is to be done on collection and conversion of the texts in one or more txt-format input files. The proposed analysis of keywords occurrences is implemented in a web-based PHP – MySQL application. It is already used for the practical testing and provides a basis for the future improvements.*

***Keywords:*** *article structure, in-text tagging, keyword counting, keyword occurrence analyzer.*

Izvoran znanstveni rad

***Sažetak.*** *Autori predstavljaju koncept i implementaciju analize pojavljivanja ključnih riječi koja se temelji na zasebnom brojanju (ključnih) riječi u četiri glavna dijela znanstvenih i inih članaka: njihovom naslovu—iza kojeg slijedi lista autora, sažetku, listi ključnih riječi i glavnom dijelu teksta. Također, ova je analiza zamišljena za primjenu na više od jednog članka odjednom. Kao takva ona može poslužiti istraživačima, piscima, urednicima i drugima za pregledavanje i statističku obradu članaka. Za razlikovanje pojedinih dijelova članaka, uz već postojeće, implicitno označavanje, potrebna je minimalna pomoć korisnika u fazi prikupljanja i konverzije tekstova u jednu ili više ulaznih datoteka txt formata. Predložena analiza pojavljivanja ključnih riječi implementirana je u spletnoj PHP – MySQL aplikaciji. Ona već služi za praktično testiranje novog koncepta i predstavlja osnovu za daljnja usavršavanja.*

***Ključne riječi:*** *struktura članka, označavanje teksta, brojanje ključnih riječi, analizator pojavljivanja ključnih riječi.*

## 1. INTRODUCTION

Counting the occurrences of words is one of the simplest statistical analyzes that can be easily done even on entirely unstructured natural language texts. In the field of *text* (*data*) *mining*, also known as *text analytics* [1], it is considered as quite a basic approach, which is reduced to the pure statistics of the used vocabulary. However, the simplicity of the method—comparing to the sophisticated methods for extraction of semantic contents and knowledge—does not diminish its value in practical use. Quite the contrary, writers, reviewers, editors, and linguists find the word frequency counters as an indispensable tool for the analysis of different kinds of texts [2].

In this paper we outline the idea, concept and implementation of a new approach to counting the occurrences of words. It is designed for analysis of more than one article at once, and it focuses on the statistics of the selected words only—the keywords defined in the articles and the additional (key)words defined by the user. These words are then counted separately in different parts of each article: its title, abstract, list of keywords and the main text, i.e. in those of the mentioned parts that are available to the user.

Motivation for such analysis emerged when one of the authors of this paper was collecting materials for his research, and was confronted with the need to investigate the appropriateness and relevance of dozens and hundreds of scientific papers for the topic of his study. Also, other uses of such analysis can be readily suggested. They all can be summarized as follows:

- The use by researchers in some scientific or professional area;
- The use by writers of review articles on a particular topic or area;
- The use by editors of scientific and other journals;
- Linguistic, ontological, and similar studies.

### 1.1 Originality of the proposed analysis

After realizing the need for a new information retrieval tool, the authors of this paper made a preliminary check if it, or at least something similar, already exists. However, the search over Internet discovered only the usual word frequency counters, which provide the statistics of *all* the words that are used in a *single* text, treated as a unity. These programs, implemented either as web-applications or stand-alone programs [3], quite often offer very useful statistical functions and other features, but none among dozens of those checked by us could perform the proposed analysis. Also, such analysis was not found possible in any of the investigated scientific databases (two examples are in [4]), which would otherwise provide an excellent platform for the job.

The negative result of the above search is quite a surprising one, especially because of the relative simplicity of the outlined idea and the fact that no sophisticated data mining methods are needed for its realization. In fact, it is good to note that the desired analysis of the standard textual files may be done with the help of text processors, and for the HTML files, in all modern web browsers. For that job, it is convenient that the *Find* (*string*) function immediately gives the number of occurrences of a specified word or phrase in the observed text. However, by doing a word count analysis this way, much extra processing has to be done manually by the user. That includes the selection and the manipulation of desired parts of the text separately for each article, as well as recording of the obtained counts for every (key)word in a user-created table.

So, because we could not find any application that would perform the above-described analysis, we assumed that its concept was original. That posed a high motivation for us to proceed with its formal description and implementation in an original program.

### 1.2 Outline of this paper

After we have given a short description of the new kind of keyword occurrence analysis and shown its originality in the previous parts, in section 2 we describe its concept more formally. There we define the specifics of the analysis and the requirements on the organization and format of the implicit and additional tagging of the article parts. Section 3 presents the relation schemas that are derived directly from the desired results. In section 4, a brief overview of the application program is given, and an excerpt of its PHP code is shown. Section 5 provides an example of the use of our analyzer. The paper is concluded with section 6, which also gives a brief outlook on the possible improvements of the analysis and its implementation.

## 2. THE CONCEPT, RESTRICTIONS, AND REQUIREMENTS

### 2.1 General features

In the proposed analysis we count the occurrences of the selected words only, which can be of two types:

- *Keywords*– defined “implicitly” by the authors of analyzed articles, in the articles’ lists of keywords.
- *Additional* (*key*)*words* – defined explicitly by the performer, or the user, of the analysis. These words are added to the set of implicitly defined keywords, from the previous point.

When no distinction between the words of the above two kinds is necessary, both of them will be simply called *keywords*.

The two essential features of our analysis are:

I. Counting of the keywords is done on an arbitrary number of articles that are chosen by the user;

II. Occurrences of the selected keywords are counted separately in the following four main parts of a (scientific) article:
   1. Title (followed by a list of authors);
   2. Abstract;
   3. List of keywords;
   4. Main text.

### 2.2 Restriction of input files to a unified format

In order to simplify the parsing of and searches through the multiple textual files, we impose the following restriction:

- The input files must be in the (common) txt format.

Such constraints can be justified, especially in the early versions of the program. Similar restrictions of the allowed file formats are found in the aforementioned standard word frequency counters. In the case of the web-based applications [3], the adjustment to the HTML format is made upon copying the text from the operation system’s clipboard. Even the professionally sold software mentioned in [3] requires the input document files to be in the txt format.

The conversion to the txt files can be easily done from both the specialized text-processor formats and from the PDF—the today’s prevailing computer format for presentation and dissemination of articles of all kinds. The former conversion can be done in the text processors alone. The latter can be done in various PDF editors and in many specialized, stand-alone or web-based applications, which can be easily found on Internet.[1]

### 2.3 Structure of (scientific) articles and their internal marking

A scientific paper that follows the today’s standard structure will have all four parts which are listed at the end of the subsection 2.1. Less structured articles may be missing some of those parts, but never the title—the part marked with number 1 in our division.

The title, together with the name(s) of the article’s author(s), essentially defines every article. Furthermore, the main text serves the very purpose of the article. So, the title and the main text must be present in every article of each kind. Regarding the availability of these parts to the user, only the title must be provided for our analysis. The main text may not be needed for some surveys or may not be available, e.g. because the full version of the article it belongs to must be purchased.

The abstract is almost always both present and available, for free. In popular and newspaper articles, a leading paragraph can and often does serve the purpose of a more or less formal abstract. On the other hand, the keywords may be missing quite often. Some of the most reputable scientific journals do not have them. Also, the keywords are almost never present in popular magazines or news articles.

The wide variety of the appearances of abstracts and keyword lists in articles is followed by a similar variety of their already-existing, *internal marking*, or *tagging*. The internal marks of the article parts are somewhat

[1] A suitable query for them is “pdf to txt converter.”

analogous to the *tags* in the field of information storage and retrieval and text mining [1]. Obviously, if an abstract is informal, it will not be marked in any way. The formal abstracts are usually preceded by the words *Abstract* or—in the texts of older style—*Summary*. (In the Latin writing order, they are placed to the left of or above the abstract). However, even the formal abstracts may not be designated by any special words, but only by a distinct formatting or a significant placement—e.g. just below the article's title and the name(s) of the author(s). In contrast to that, the lists of keywords—when they exist—are always appropriately marked by the word *Keywords*, sometimes written as *KeyWords*.

Despite the apparent complexity of the possible article structure and diversity of its internal marking, the structure of articles of all forms can be efficiently parsed (resolved) by the use of already existing or slightly modified marks of the article parts and, if needed, by some additional tagging that is done during the preparation of the txt files (see 2.5). For that, all text editors that accept clipboard contents can be used.

To summarize, minimally required information about an article is its title, and all other parts of the article can be considered as optional (see Table 1). It is expected that the user will want to make his or her collection of papers as consistent as possible, e.g. by providing at least parts 1, 2 and 3 for all of them, and by, perhaps, adding the main text (part 4) for those articles that are available in full.

## 2.4 Search through multiple articles

Searching for the keywords and counting their occurrences in *multiple* articles at once is one of the very foundations of our analysis. To make this concept as flexible as possible for the user, we request that the implemented program must be able to process an arbitrary number of txt files placed in the application's working directory, each of which can contain either:

i. Only one, separate article or its selected parts, or
ii. Several articles or their selected parts, which are placed one below the other.

If there are multiple files, they will be processed in alphabetical order, which is also the order of their appearance in the application's working directory.

## 2.5 Search in the specific parts of the articles

The above deliberation leads us to the tagging scheme shown in Table 1. The idea is to use the standard part names as much as possible, in order to minimize the assistance needed by the user. Thus, the words *Abstract* and *Keywords* will be the tags of the parts 2 and 3, respectively, i.e., more precisely, the tags' root words.

The tagging is mandatory only for the title and only if the article follows another article. That is, if the article is the first one in a file, the title tag can be omitted. The *designated tag* for title is the string *TitleS*, where *S* must be formed according to the regular expression

$$S = ('.' + ':') + (\varepsilon + '\ '^{*} + '.' + ':')P. \qquad (1)$$

Here $\varepsilon$ is the empty string, and $P$ denotes the character or sequence of characters for entering, or creating, a new "line of text," which will be normally seen as a new paragraph.[2] The requested, i.e. allowed punctuation and the blank character are in single quotes. The plus (OR) operator denotes the possibility of choice between two or more characters or strings, and the asterisk means that the denoted string may be repeated. Writing a string next to its predecessor (as $P$ in our case) means the concatenation of the two. So, according to Eq. (1), after the word *Title* either: (i) a dot or a semicolon must follow, or (ii) nothing ($\varepsilon$), or one or more blank spaces, or a full stop, or a semicolon, after which a new paragraph must be entered. The analogous rules are applied for the tags of the other article parts (Table 1).

As for the remaining, non-mandatory article parts—if they are not present, their tags will not be required. If the parts are present, the assistance needed from the user is also minimized. These parts will have to be tagged only if their recognition is required. To fulfill this request, the lengths of all parts that precede an *untagged* part must be known in advance. Vice-versa, the part preceding a properly tagged part can have an arbitrary, unknown length. The last part (4) can have an arbitrary length, because it can precede only the next article's title, which—as stated before—must be tagged. This tagging scheme is presented in Table 1. In the current version of our application, a bit less general solution is implemented: the parts that precede the non-mandatory article parts are always restricted to a single paragraph.

If the prescribed, English, tagging scheme is obeyed, the proposed analysis could be in general done on articles written in any language. The former restriction to the txt files only limits that to the Latin-alphabet-based languages, whose texts are ASCII-encodeable.

**Table 1**. The article parts and their tagging scheme. *S* is a string formed according to Eq. (1). The asterisk denotes that the predefined part length could be alleviated if the subsequent (non-mandatory) part is correctly tagged.

| **Part** **& presence** | **Tag** | **The tag usage and comments** |
|---|---|---|
| 1. Title, mandatory, 1 paragraph* | *TitleS* | Mandatory, unless the article, and thus also its title, appear at the beginning of the txt file [always valid for the files of type (i) in 2.4]. |
| 1a. List of authors, optional, 1 paragraph* | *AuthorsS* | Optional. The tag must precede the list of authors if their names are required in the description of the article (Table 2a). If omitted, the authors' names will not be extracted. |
| 2. Abstract, optional, 1 paragraph* | *AbstractS*, *SummaryS* | Optional. The tag must precede this part if it is required to be recognized. If missing, the text will be ignored. |
| 3. List of keywords, opt., 1 par.* | *KeywordsS* | Optional. Analogous to the usage of the tag of part 2. *A note*: the keywords are separated by ',' or ';'. |
| 4. Main text optional, multi-par. | *MaintexS*, *Main textS* | Optional. Analogous to the usage of the tag of part 2. |

[2] This is accomplished by special characters, like EOL = *End Of Line*, or by the combination LFCR, consisting of LF = *Line Feed* and CR = *Carriage Return*.

## 3. A GLIMPSE TO THE RELATIONAL MODEL AND ITS IMPLEMENTATION

Here we just shortly sketch the main relations that are needed for the implementation of the database that will support our analysis.

### 3.1 Main relation schemas

The above-outlined concept gives a rise to the two basic relation schemas that are needed for our analysis. They are given in Table 2a and Table 2b.[3] The first one describes an article. An article is uniquely and non-redundantly identified by the number (*No.*) of its appearance in the scan of the subsequent input txt files, and can thus serve as the primary key (underlined). Its range of values corresponds to that of the unsigned integer data type. The value *No.* = 0 is reserved for a non-existing, "dummy" article, with no parts of its own, except for the list of keywords that is composed of the additional (key)words entered by the user.

As emphasized earlier, the minimal requirement for a text to be recognized as an article is that it has a title. That is, for a 6-tupple describing an article to be valid, the textual value of the attribute *Title* must be defined. If the tag *AuthorsS* is not placed after the article title, the authors' names will not be extracted in the body of the relation (see Table 1). The remaining attributes' values are of logical type, and they show if the abstract, the list of keywords, and the main text are present, i.e. recognized, in the article's text.

The second relation schema describes the keyword occurrence counts within the four article parts (Table 2b). That is done separately for the parts of the article that the keyword originates from—which bear the number *No.* (marked as *orig.*), and for the same parts of all other articles that are present in the application's working directory (marked as *othr.*). Thus, the relation consists of 10-tupples. A careful reader will notice that if this distinction was not made, i.e. if only the total number of keyword occurrences within the specified article parts were counted, the relation would lead to a redundant set of 6-tuples. To put the latter relation in the third normal form, the article number attribute (*No.*) would have to be removed, leaving the *Keyword* as the single-attribute key of the new 5-tuple relation.

The above relation schemas can also be formed by conversion from the Entity-Relationship (ER) model of the described analysis. The ER model [7] is based on the three classes of objects: entities, their attributes, and the connections among the entities. According to the

**Table 2a.** Relation schema for the article. The attributes are written in italics. Uint is an unsigned integer.

| ARTICLE (SCIENTIFIC PAPER) | | | | | |
|---|---|---|---|---|---|
| <u>No.</u> | *Title* | *Authors* | *The presence of:* | | |
| | | | *Abstract* | *Keywords* | *Main text* |
| Uint | A paragraph of text | A par. of text ∪ NULL | Logical type ∪ NULL | | |

**Table 2b.** Relation schema for the keyword occurrence counts in the specific parts of: originating article (*orig.*) with number *No.*, and of all other articles (*othr.*).

| Keyword Occurrence Counts In: | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| <u>No.</u> | <u>Keyword</u> | *Title(s)* | | *Abstract(s)* | | *KyW. Lst(s)* | | *Main txt(s)* | |
| | | *orig.* | *othr.* | *orig.* | *othr.* | *orig.* | *othr.* | *orig.* | *othr.* |
| Uint | One or more words | Uint (unsigned integer) ∪ NULL | | | | | | | |

the well-known rules of conversion from the ER to the relational model, each entity becomes a relation, and the entity attributes become the relation attributes (the columns of R-table). For the transformation of connections of several other types, the methodology is given e.g. in [8]. In our case, "many to many" connection between the entities *Article* and *Keyword* is transformed into the relation schema of Table 2b, with the combined primary key <u>*No.+Keyword*</u>.

### 3.2 The generalities of implementation

The design and implementation of a database are generally complex processes that must be performed on the conceptual (semantic), implementational, and physical level. If we go in the reverse order, the process requires definition of the following:[4]

- Hardware and operation system for stand-alone applications and the web server for the web-based applications;
- Data model and database;
- Programming language.

A choice that assures portability over most of today's web servers, and that can be also realized to run as a "stand-alone" application on most of today's operation systems, is the use of MySQL database and PHP scripting language. It is the de-facto standard choice for an open-source relational database management system (RDBMS). In our concrete case, the database and application are assisted by Apache web server, which also provides the necessary data storage and access to the Wide Area Network (WAN) services. As discussed above, the data model was initially defined by the relation schemas depicted by Table 2a and Table 2b, which both result from the demands of the proposed analysis. A full description of its ER and relational models will be given in the subsequent publications.

The implemented application is named *KeyWords Occurrence Analyzer*, with acronym KWOA (see Figure 2). In further text, it will also be shortly called the *analyzer*. For its recent URL placement, the readers are kindly asked to contact the authors of this paper.

[3] Here follows a short reminder of the used terminology. The *relation* is a set of *n*-tuples, each defining *n* values, described by *n attributes*. The relation can be depicted as a *body* of the *table* to which the set of *n* attributes is the *heading*. In [5] this is called *R-table*. The heading defines the order of attributes, so that the relation can be also defined as a heading of a table paired with its body. A heading with the constraints on its attributes' values is called *relation schema* (not to be confused with *relational*, or *database, schema* [6]). To differentiate the (body of the) table holding a relation from other, non-relational forms of tables, a redundant term *relational table* is sometimes used in the database professional jargon.

[4] A systematic elaboration of this process can be found in [9].

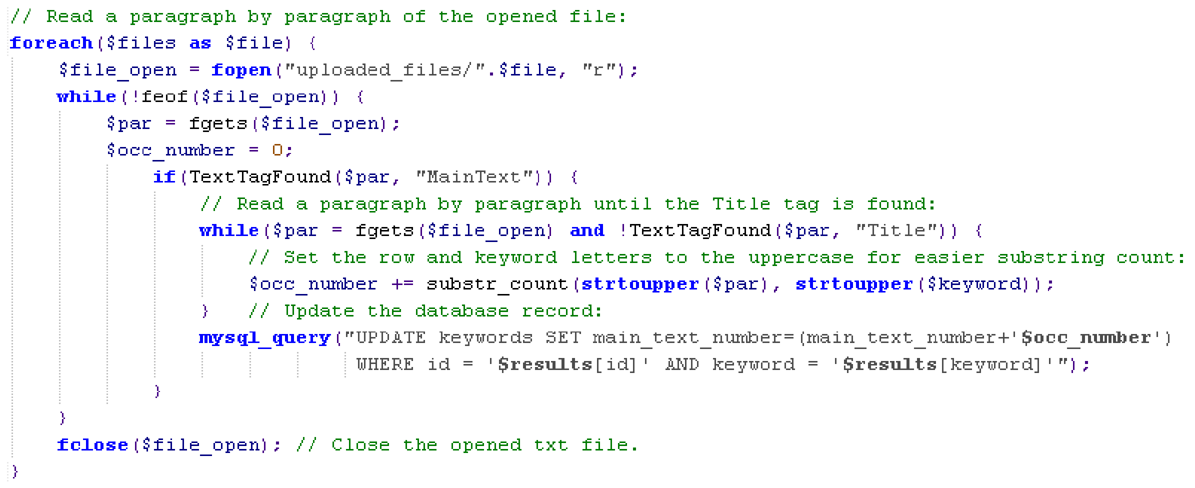
## 4. A SHORT OUTLINE OF THE PROGRAM

The present version of KWOA PHP program is executed in the following four steps:

1. Do a passage through the pre-prepared txt files uploaded to the working directory (Figure 2), in the alphabetical order, to find the titles, authors, and abstracts, and store them in the database-version of the Table 2a (the latter two if they are tagged, the abstracts goes in the column that is appended to the table). Also, find the keywords and store them in the implementational version of the Table 2b.
2. For each of the keywords stored in the database-version of Table 2b, count its occurrences within the titles and abstracts (stored as in the Table 2a) and within the lists of keywords (stored as in the Table 2b), and then—in another pass through the input textual files—within the articles' main texts.
3. Do an additional passage through the input files to extract the numbers of articles in whose parts the keywords appear. This data are displayed in the third table of the application (*not* shown in sec. 5).
4. The diagrams of the occurrence counts are formed for the (ten) most frequent keywords.

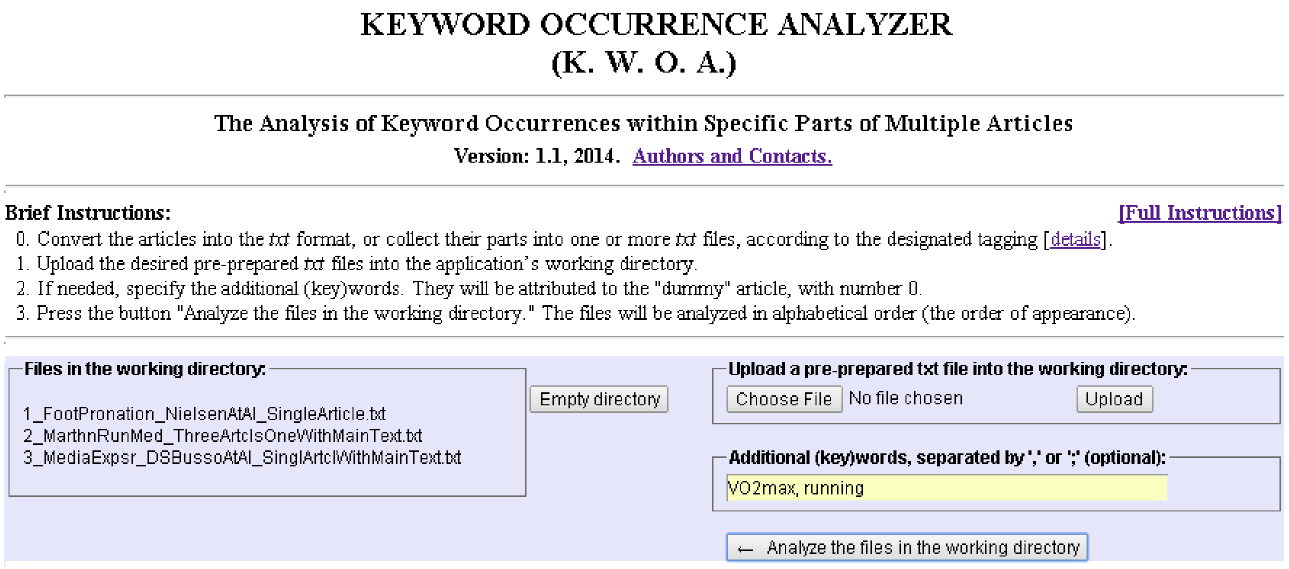
A part of the step 2 program code that finds occurrences of keywords in the articles' main texts is shown in Figure 1. The TextTagFound function searches for the main text tag according to the scheme depicted in Table 1 and Eq. (1). This, case-insensitive, search is done by using the regular expressions, which are in PHP provided by preg_match function. [As also reported by others, the preg_match function causes problems on encountering some nonstandard symbols.] In the while loop, the function substr_count counts the occurrences of keywords in the main text for all available paragraphs or until the next designated title tag (*TitleS*) is found. Then mysql_query function updates the counts for that keyword in the database version of Table 2b.

The current version of the program requires revision, optimization, and further testing. The main possible improvements will be briefly discussed in sec. 6.



```
// Read a paragraph by paragraph of the opened file:
foreach($files as $file) {
    $file_open = fopen("uploaded_files/".$file, "r");
    while(!feof($file_open)) {
        $par = fgets($file_open);
        $occ_number = 0;
            if(TextTagFound($par, "MainText")) {
                // Read a paragraph by paragraph until the Title tag is found:
                while($par = fgets($file_open) and !TextTagFound($par, "Title")) {
                    // Set the row and keyword letters to the uppercase for easier substring count:
                    $occ_number += substr_count(strtoupper($par), strtoupper($keyword));
                }   // Update the database record:
                mysql_query("UPDATE keywords SET main_text_number=(main_text_number+'$occ_number')
                             WHERE id = '$results[id]' AND keyword = '$results[keyword]'");
            }
    }
    fclose($file_open); // Close the opened txt file.
}
```

**Figure 1.** PHP code for counting the keyword occurrences in the articles' main texts.



KEYWORD OCCURRENCE ANALYZER
(K. W. O. A.)

The Analysis of Keyword Occurrences within Specific Parts of Multiple Articles
Version: 1.1, 2014. Authors and Contacts.

Brief Instructions: [Full Instructions]
0. Convert the articles into the *txt* format, or collect their parts into one or more *txt* files, according to the designated tagging [details].
1. Upload the desired pre-prepared *txt* files into the application's working directory.
2. If needed, specify the additional (key)words. They will be attributed to the "dummy" article, with number 0.
3. Press the button "Analyze the files in the working directory." The files will be analyzed in alphabetical order (the order of appearance).

Files in the working directory:
1_FootPronation_NielsenAtAl_SingleArticle.txt
2_MarthnRunMed_ThreeArtclsOneWithMainText.txt
3_MediaExpsr_DSBussoAtAl_SinglArtclWithMainText.txt

Upload a pre-prepared txt file into the working directory:
No file chosen

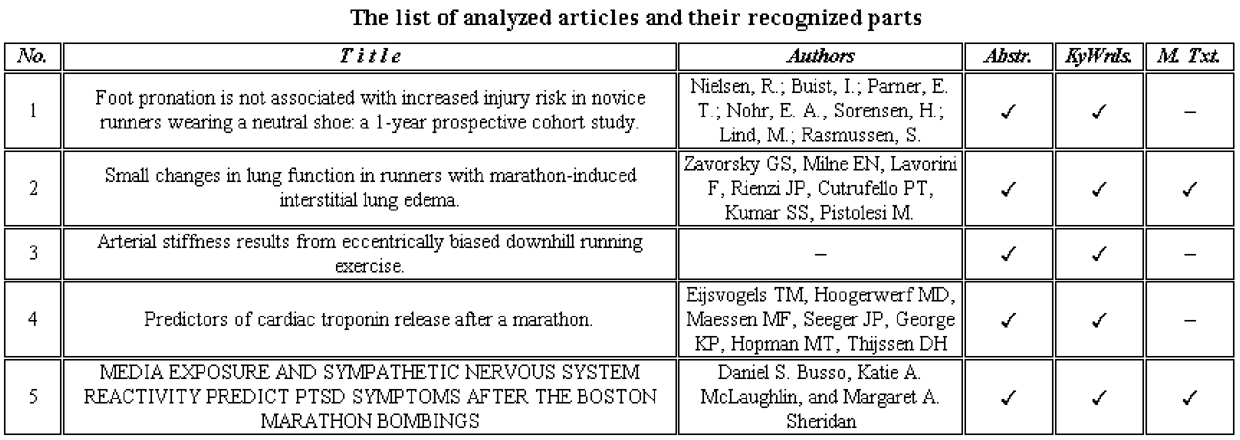
Additional (key)words, separated by ',' or ';' (optional):
VO2max, running

**Figure 2.** The print-screen of the top part of KWOA application, with its command section. Three pre-prepared *txt* files are uploaded to the application's working directory, and two additional (key)words are added.

## 5. AN EXAMPLE OF USE

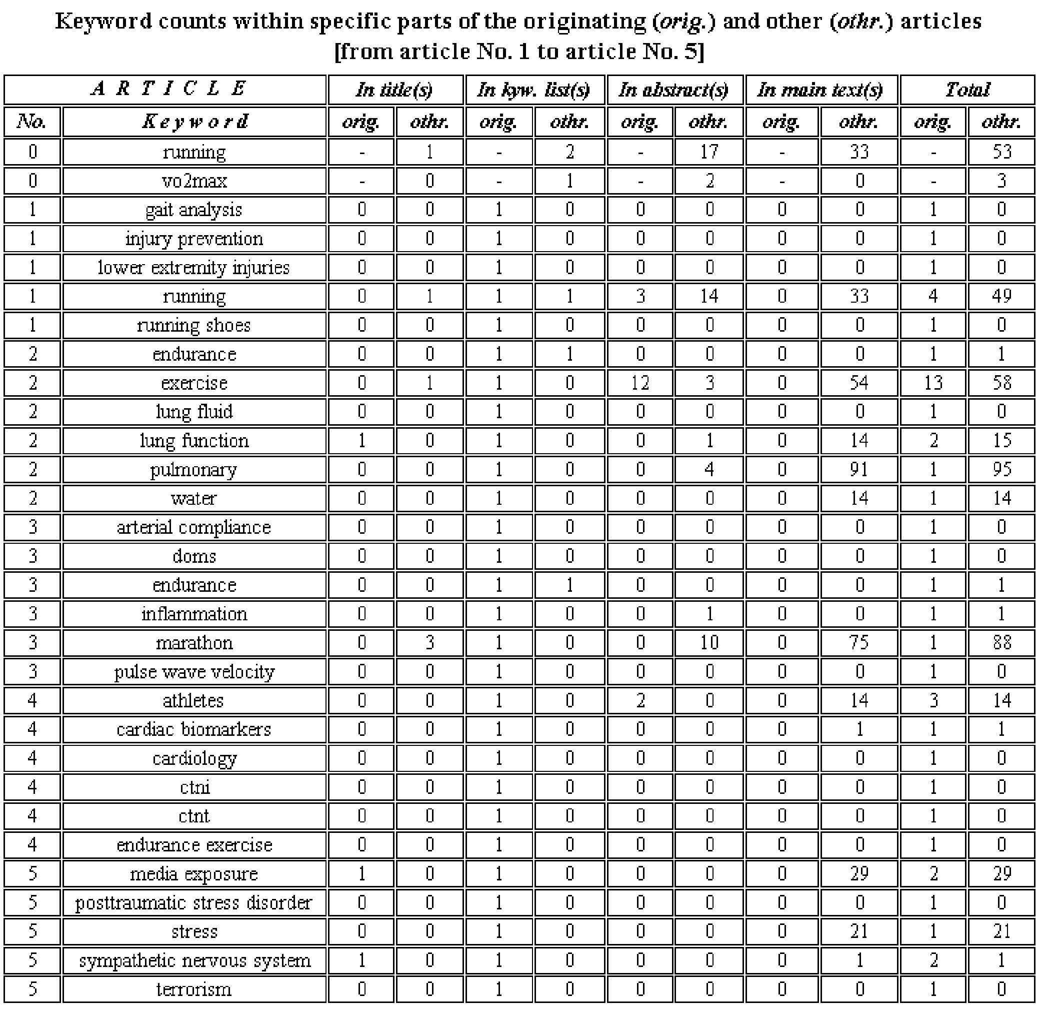
As an example of the use of KWOA application, we give the analysis of scientific papers that deal with the medical aspects of the marathon running and training, extracted from the PubMed scientific database [10].

To perform the keyword occurrence analysis, the user should follow the brief instructions given in the top part of the application (Figure 2). The three txt files prepared for this example comprise two whole articles and three article excerpts containing parts 1 to 3 (confer Table 1). Figure 3a shows the table that gives the description of the articles according to the relation schema given in Table 2a. In paper *No.* 3, the list of authors was not (properly) tagged by the user so that its authors are missing. Figure 3b displays the second table of the KWOA application. It reproduces the data according to our crucial relation schema, shown in Table 2b. In this, presentational, form of the table, two additional columns are added. They summarize the total keyword counts that are found


**The list of analyzed articles and their recognized parts**

| *No.* | *T i t l e* | *Authors* | *Abstr.* | *KyWrds.* | *M. Txt.* |
|---|---|---|---|---|---|
| 1 | Foot pronation is not associated with increased injury risk in novice runners wearing a neutral shoe: a 1-year prospective cohort study. | Nielsen, R.; Buist, I.; Parner, E. T.; Nohr, E. A., Sorensen, H.; Lind, M.; Rasmussen, S. | ✓ | ✓ | – |
| 2 | Small changes in lung function in runners with marathon-induced interstitial lung edema. | Zavorsky GS, Milne EN, Lavorini F, Rienzi JP, Cutrufello PT, Kumar SS, Pistolesi M. | ✓ | ✓ | ✓ |
| 3 | Arterial stiffness results from eccentrically biased downhill running exercise. | – | ✓ | ✓ | – |
| 4 | Predictors of cardiac troponin release after a marathon. | Eijsvogels TM, Hoogerwerf MD, Maessen MF, Seeger JP, George KP, Hopman MT, Thijssen DH | ✓ | ✓ | – |
| 5 | MEDIA EXPOSURE AND SYMPATHETIC NERVOUS SYSTEM REACTIVITY PREDICT PTSD SYMPTOMS AFTER THE BOSTON MARATHON BOMBINGS | Daniel S. Busso, Katie A. McLaughlin, and Margaret A. Sheridan | ✓ | ✓ | ✓ |

**Figure 3a.** The first table in KWOA application, which resembles the Table 2a. In the three input files visible in Figure 2, five scientific papers are found. In the third of them, the authors were not tagged and are missing.


**Keyword counts within specific parts of the originating (*orig.*) and other (*othr.*) articles [from article No. 1 to article No. 5]**

| *A R T I C L E* | | *In title(s)* | | *In kyw. list(s)* | | *In abstract(s)* | | *In main text(s)* | | *Total* | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *No.* | *K e y w o r d* | *orig.* | *othr.* | *orig.* | *othr.* | *orig.* | *othr.* | *orig.* | *othr.* | *orig.* | *othr.* |
| 0 | running | - | 1 | - | 2 | - | 17 | - | 33 | - | 53 |
| 0 | vo2max | - | 0 | - | 1 | - | 2 | - | 0 | - | 3 |
| 1 | gait analysis | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | injury prevention | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | lower extremity injuries | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | running | 0 | 1 | 1 | 1 | 3 | 14 | 0 | 33 | 4 | 49 |
| 1 | running shoes | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 2 | endurance | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 2 | exercise | 0 | 1 | 1 | 0 | 12 | 3 | 0 | 54 | 13 | 58 |
| 2 | lung fluid | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 2 | lung function | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 14 | 2 | 15 |
| 2 | pulmonary | 0 | 0 | 1 | 0 | 0 | 4 | 0 | 91 | 1 | 95 |
| 2 | water | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 14 | 1 | 14 |
| 3 | arterial compliance | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 3 | doms | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 3 | endurance | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 3 | inflammation | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| 3 | marathon | 0 | 3 | 1 | 0 | 0 | 10 | 0 | 75 | 1 | 88 |
| 3 | pulse wave velocity | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 4 | athletes | 0 | 0 | 1 | 0 | 2 | 0 | 0 | 14 | 3 | 14 |
| 4 | cardiac biomarkers | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 4 | cardiology | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 4 | ctni | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 4 | ctnt | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 4 | endurance exercise | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 5 | media exposure | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 29 | 2 | 29 |
| 5 | posttraumatic stress disorder | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 5 | stress | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 21 | 1 | 21 |
| 5 | sympathetic nervous system | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 2 | 1 |
| 5 | terrorism | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

**Figure 3b.** The second table of KWOA application. It reproduces the Table 2b, but has also two additional columns—for the total keyword counts (note that they infringe the third normal form).

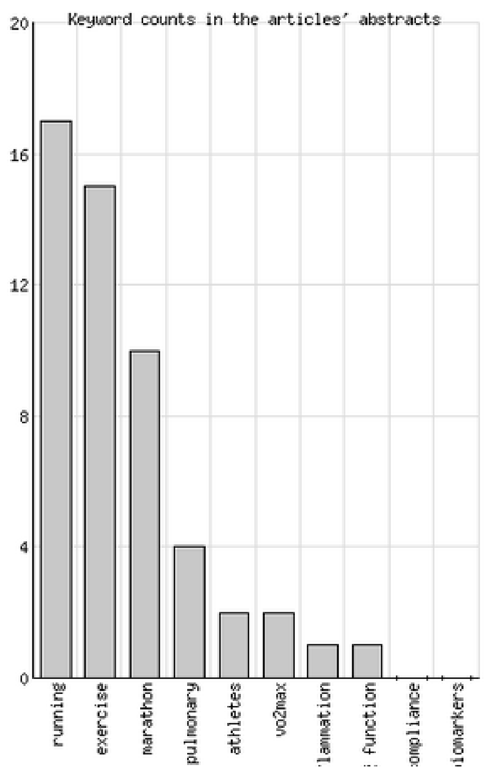


**Figure 4a.** Keyword occurrence counts within the keyword lists of the collected articles.

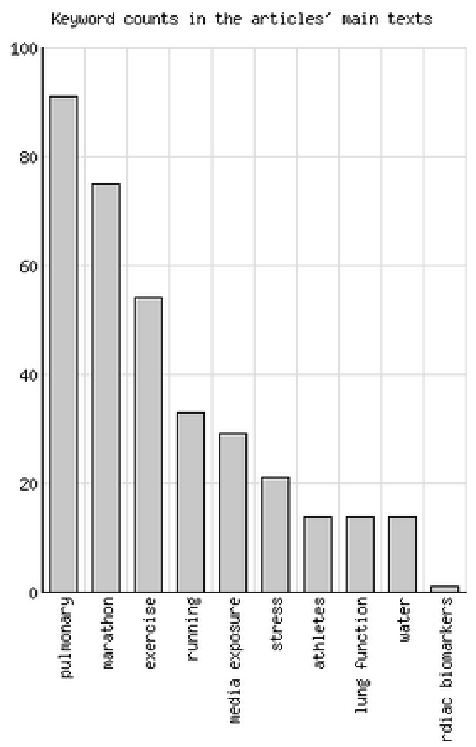


**Figure 4b.** Keyword occurrence counts within the main texts of the collected articles.

within the keyword's own and in the other articles. The analyzer's third table (see sec. 4, step 3) is not shown.

Finally, the KWOA application gives the frequency histograms for the ten most often used keywords, as found within the four parts of the analyzed articles. Here, they are pictured for the counts within keyword lists and the main texts, in Figure 4a and Figure 4b, respectively.

## 6. CONCLUSION AND FUTURE IMPROVEMENTS

The refinement of the keyword occurrence counts to the specific parts of possibly multiple (scientific) articles gives the analysis of the keyword frequency a new qualitative dimension. The overall significance of some (key)word within a set of investigated articles can be much more appropriately judged if its appearance is tracked separately in each of the four standard parts of a scientific or other type of text, as defined in Table 1.

For the present version of our keyword occurrence analyzer, the user must pre-prepare one or more textual files in the txt format. That can be done directly by conversion of the articles' documents to that format, or by collecting them or their parts in a suitable editor. Such preparation is not uncommon in today's word frequency counters. According to the desired distinction and availability of the four main article parts, the user should adjust or assist the already existing, implicit tagging of otherwise weakly or semi-structured text. The tagging scheme, given in Table 1, is made to be highly flexible. The only mandatory part of each article is its title, expected to be placed at the beginning of the article. It is required to be properly tagged, but only if the article is not the first one in a file. Tagging of the parts that are missing or whose distinction is not required may be omitted. For example, if the proper tags *AbstractS* and *KeywordsS* are already present [for *S* see Eq. (1)], the user must only take care that the articles' beginnings are properly marked (or distinguished by the file starts). Although "labor-intensive," the manual tagging assures both simplicity and high reliability of the KWOA application. Similar human-assisted solutions were and are still used in the systems that require high precision (see e.g. sec. II.2.5 in [1]).

The used tagging scheme could be reconsidered and redefined for even greater flexibility of the allowed article structure. For example, if a dedicated passage through an article within a file is made to investigate what tags are present, this will allow more flexible treatment of the cases when different numbers of tags are found. For example, if all parts (five if counting 1a) are properly tagged, their order could be fully arbitrary. If one or two non-mandatory tags are missing, the orders of article parts different from the standard one might still be allowed, but their resolution would be more complex.

For the implementation of our analysis, the combination of PHP scripting programming language and MySQL relational database is used. The overall achieved functionality of KWOA application is satisfactory. Along with the testing, it is already being used for its practical purpose outlined in section 1.

The proposed keyword occurrence analysis provides a broad foundation for several further improvements. One of the most inviting and, of course, the most difficult, is the simplification of the preparation of input files. Namely, this process requires not only the conversion of compiled articles into the unified, txt format, but also the inspection and possible manual completion of their internal tagging. Thus, if the original natural language text is not formatted in HTML—which already provides the tagging infrastructure, that could also be usable for our analysis—its format will generally be loosely structured.

So, to automate the tagging process at least partially would require a careful refinement of the rules for the recognition of the specific article parts. In the better structured, scientific papers, parsing of the text parts could be easier. For example, their main body most usually appears after the parts 1 to 3, and typically begins with a paragraph (section title) that starts with the word *Introduction*. Similar typical words may present *loose implicit markers*, which could help in locating the parts that are not more firmly tagged. However, if unassisted by humans, this method may be prone to failures.

It is obvious that additional and, if possible, diversified criteria for the automatic article's part discrimination would be very helpful. A method quite opposite to the lexical analysis used above is examination of the article's graphical appearance. For example, the titles—regardless of where they appear in an article, book, or a thesis—are usually written with the fonts of the largest size. Such formatting details are to be extracted from the article's original, PDF, HTML, or other file format. The conclusions based on them should be compared with the findings of other methods.

In the approach described above, recognition of the articles' parts could be organized as a rule-based expert system. If it is integrated in a practical software tool that could be easily and interactively supervised by the user, we predict that it could be both precise and highly efficient. Also, the (advanced) text mining methods for syntactic analysis could help the better determination of titles and subtitles. These parts can be recognized by their usual lack of the verb phrase. Finally, as for the usage of the (complex) semantic analysis, we are not sure that it would be either accurate or useful for the discrimination of the article's parts. But deliberation of that topic goes beyond the scope of this paper.

## 7. REFERENCES*

[1] Feldman, R.; Sanger, J.: The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data, Cambridge University Press, New York, NY, 2007.

[2] Jackson, H.; Amvela, E. Z.: Words, meaning, and vocabulary: an introduction to modern English lexicology, Continuum, London, 2004.

[3] a: Online Word Counter: http://www.textfixer.com/tools/online-word-counter.php, Word Freq. Counter: http://www.writewords.org.uk/word_count.asp;
b: Hermetic Word Frequency Counter: http://www.hermetic.ch/wfc/wfc.htm.

[4] a: ResearchGate: http://www.researchgate.net;
b: Croatian Scientific Bibliography: http://bib.irb.hr/.

[5] Codd, E. F.: The Relational Model for Database Management: Version 2, Addison-Wesley, Reading, Massachusetts, 1990.

[6] Wikipedia article: Relation (database), Relational table), http://en.wikipedia.org.

[7] Chen, P.: The Entity-Relationship Model—Toward a Unified View of Data, ACM Transactions on Database Systems, Vol. 1, No. 1 (1976) 9-36.

[8] Skočir, Z.; Matasić, I; Vrdoljak, B.: Organizacija obrade podataka (*the translation of the title from Croatian:* The Organization of Data Processing), Merkur A.B.D., Zagreb, 2007.

[9] Havaš, L.: Informacijski sustav za podršku u odlučivanju u treningu sportaša (*English title:* Information system for athlete's training decision support), Ph.D. Thesis, University of Zagreb, Faculty of Electrical Engineering and Computing, 2014.

[10] PubMed, US National Library of Medicine, National Institutes of Health: http://www.ncbi.nlm.nih.gov/pubmed.

* All cited Web sites and Web pages were accessible, and their URLs were correct, in December 2014.

Authors' contacts:

**Tomislav Horvat, B.Sc.**
University North, Dpt. of Electrical Engineering
104. brigade 3, HR-42000 Varaždin
tomislav.horvat@unin.hr

**Ladislav Havaš, Ph.D.**
University North, Dpt. of Electrical Engineering
104. brigade 3, HR-42000 Varaždin
ladislav.havas@unin.hr

**Robert Logožar, Ph.D.**
University North, Dpt. of Multimedia.
104. brigade 3, HR-42000 Varaždin
robert.logozar@unin.hr