Applied Computing and Intelligence

https://www.aimspress.com/journal/aci

Applied Computing and Intelligence
2(2): 115–132
DOI: 10.3934/aci.2022007
Received: 09 June 2022
Revised: 07 July 2022
Accepted: 12 September 2022
Published: 20 September 2022

Research article

Combining statistical, structural, and linguistic features for keyword extraction from web pages

Himat Shah and Pasi Fr änti*

School of Computing, University of Eastern Finland, Joensuu, Finland
*** Correspondence:** franti@cs.uef.fi

Academic Editor: Chih-Cheng Hung

Abstract: Keywords are commonly used to summarize text documents. In this paper, we perform a systematic comparison of methods for automatic keyword extraction from web pages. The methods are based on three different types of features: statistical, structural and linguistic. Statistical features are the most common, but there are other clues in web documents that can also be used. Structural features utilize styling codes like header tags and links, but also the structure of the web page. Linguistic features can be based on detecting synonyms, semantic similarity of the words and part-of-speech tagging, but also concept hierarchy or a concept graph derived from Wikipedia. We compare different types of features to find out the importance of each of them. One of the key results is that stop word removal and other pre-processing steps are the most critical. The most successful linguistic feature was a pre-constructed list of words that had no synonyms in *WordNet*. A new method called *ACI-rank* is also compiled from the best working combination.

Keywords: web mining; text analysis; keyword extraction; document object model tree

1. Introduction

Keywords are widely used to summarize text documents. They can be manually annotated by humans or automatically generated by computer. Automatic *keyword extraction* from web pages refers to the selection of a set of words from the document that best describe its content. The keywords can further be used for information retrieval, document retrieval, document clustering, document classifying, indexing, summarization and topic detection [1].

On one hand, extracting keywords from web pages is more difficult than from plain text because the additional information like menu and navigational bars, comments, adverts and all the formatting codes present in an HTML document can disturb the process. On the other hand, the HTML code provides additional clues about which words are more important than others. There exist many techniques for keyword extraction from plain text, but they usually do not pay attention to the structure of the page. In this paper, we focus on keyword extraction from web pages.

Figure 1 shows an example of a webpage with a complex, free form structure containing heterogeneous text from multiple languages scattered across the page. In this work, we focus on the candidate word selection and features used to score these candidates. Most existing techniques utilize three different types of features:

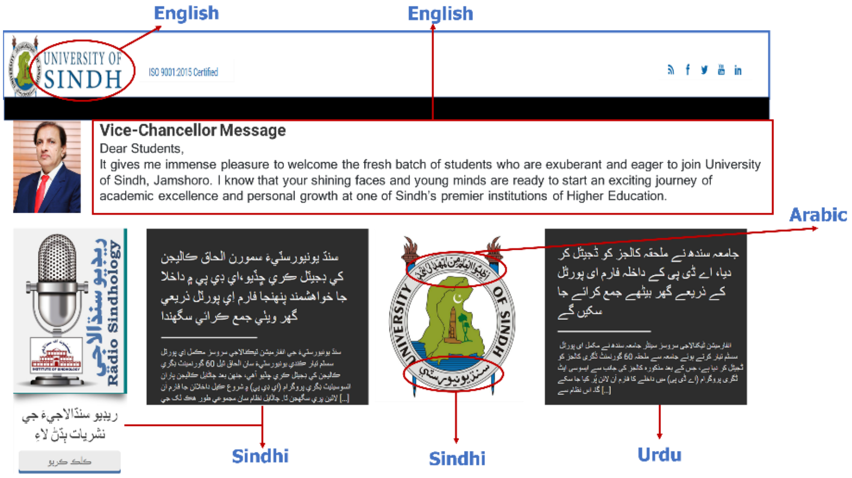
1. Statistical
2. Structural
3. Linguistic

Statistical features can be simple frequencies of the words with the idea that more frequent words are more important than less frequent ones. However, this would lead to choosing common words such as *the*, *and*, *for*. A normalization is therefore needed by giving more weight to words that are frequent in the given document but not so in other documents. A corpus like Wikipedia and frequencies of the word use in search engines can be used, but also simple *stop word* lists can be rather powerful.

Structural features utilize emphasis, links and meta information in the HTML code. For example, words included in the header tags are more likely to be good keywords. Simple formatting like **boldface**, *italic* and Capitalization may also reveal good keywords. Meta information itself may already include manually annotated keywords, but keywords are often included in the title and even part of the link of the web page (URL). The structure and spread of the words across the web page can provide further clues about the importance of the words.

Linguistic features utilize the semantic meanings of the words and their roles in the sentences. For example, nouns are more likely to be keywords than verbs and adjectives. There are also good tools available to analyze popular languages like English. However, finding language processing tools such as *part-of-speech* (POS) taggers can be challenging for smaller and grammatically more complex languages like Finnish. Simple solutions like stop words are easier, as they can be found for many languages. Language-independence would make the method more general.

In this paper, we perform systematic comparison of the most common features used in keyword extraction. We evaluate them on twelve datasets containing 2935 web pages. We start with the statistical features and construct a simple baseline from the best combination. We then study the effects of different structural and linguistic features on the accuracy. We propose a new method called *ACI-rank* from the best working combination while aiming at keeping the method simple and general. It is compared to the existing methods in the literature.



Vice-Chancellor Message

Dear Students,

It gives me immense pleasure to welcome the fresh batch of students who are exuberant and eager to join University of Sindh, Jamshoro. I know that your shining faces and young minds are ready to start an exciting journey of academic excellence and personal growth at one of Sindh's premier institutions of Higher Education.

سنڌ يونيورسٽيءَ سمورن الحاق ڪاليجن
کي ڊجيٽل ڪري ڇڏيو،اي ڊي پي ۾ داخلا
جا خواهشمند پنهنجا فارم اِي پورٽل ذريعي
گهر ويٺي جمع ڪرائي سگهندا

سنڌ يونيورسٽيءَ جي انفارميشن ٽيڪنالاجي سروسز مڪمل اي پورٽل
سسٽم تيار ڪندي يونيورسٽيءَ سان الحاق ٿيل 60 گورنمينٽ ڊگري
ڪاليجن کي ڊجيٽل ڪري ڇڏيو آهي، جنهن بعد ڇاڻيل ڪاليجن ۾ ان
ايسوسيئيٽ ڊگري پروگرام (اي ڊي پي) ۾ شروع ڪيل داخلائن جا فارم آن
لائين پري سگهجن ٿا. ڇاڻيل نظام سان مجموعي طور هڪ لک جي [...]

جامعہ سندھ نے ملحقہ کالجز کو ڈیجیٹل کر
دیا، اے ڈی پی کے داخلہ فارم ای پورٹل
کے ذریعے گھر بیٹھے جمع کرائے جا
سکیں گے

انفارمیشن ٹیکنالاجی سروسز سینٹر جامعہ سندھ نے مکمل ای پورٹل
سسٹم تیار کرتے ہوئے جامعہ سے ملحقہ 60 گورنمنٹ ڈگری کالجز کو
ڈیجیٹل کر دیا ہے، جس کے بعد مذکورہ کالجز کی جانب سے ایسوسی ایٹ
ڈگری پروگرام (اے ڈی پی) میں داخلے کا فارم آن لائن پُر کیا جا سکے
گا اس نظام سے [...]

Figure 1. Example of complex multi-lingual web page with heterogenous structure.

The rest of the paper is organized as follows. In Section 2, we define the problem and review the most relevant literature. We then study each of the three features as follows: statistical features in Section 3, structural features in Section 4 and linguistic features in Section 5. Experiments are presented in Section 6, and conclusions are drawn in Section 7.

2. Keyword extraction

Keyword extraction has numerous challenges:

- Diversity of the words
- Keywords may not always appear in the page as such
- Keywords vs. Key phrases
- Multi-lingual pages
- Multiple topics on the same pages
- Structure of the webpage

We focus on single keywords even if there are examples where key-phrases (*formula one*) would be more appropriate. Nevertheless, most of the methods would generalize to key-phrases via n-grams. The questions of how many keywords and the issues of having multiple-languages and multiple topics in the same page are also not considered.

The overall framework of the studied keyword extraction framework is summarized in Figure 2. The main parts are the cleaning and extraction of the text, selection of the candidate words, calculating the features and scoring. *Natural language processing* (NLP) can be highly useful but also time-consuming and limited to specific languages. It might even be difficult to decide which is the base language. From various NLP tools, we therefore consider only POS tagging and stop word lists.

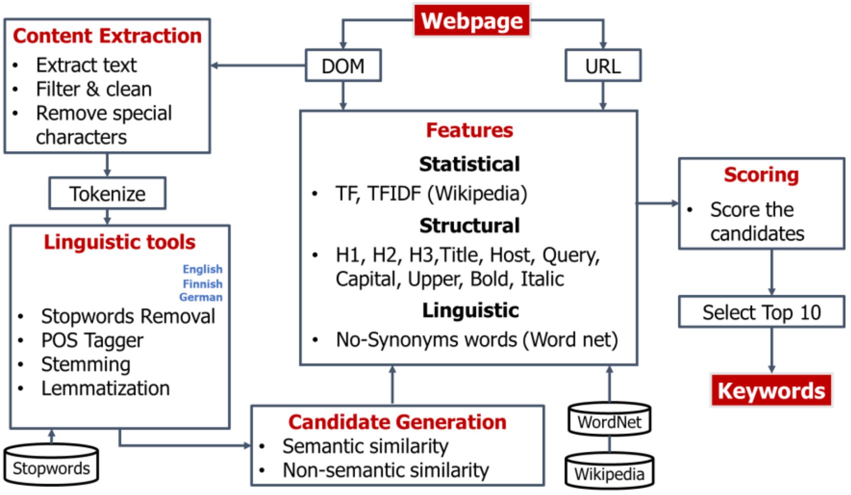


Figure 2. Framework for the keyword extraction combining statistical, structural and linguistic features.

2.1. Pre-processing

Pre-processing is one of the most critical steps in NLP because it shapes the results based on how we transform the input document into features [2]. Most typical pre-processing techniques are summarized in Table 1. A document is a collection of sentences. To extract the keywords, we need the natural language words inside the document. A typical pre-processing step is therefore to remove unnecessary information such as numbers and punctuation marks [1,2,8,18].

The remaining words can be further processed by stemming and lemmatization. A *lemma* is a chosen convention to represent a set of words (lexeme) originating from the same root and having the same meaning. For example, *break*, *breaks*, *broke*, *broken* and *breaking* all have roots to the same lemma, *break*. Lemmatization is the process or converting words into their lemmas.

Stem represents the root of a word carrying its lexical meaning. Unlike a lemma, a stem is always part of the original word and may not be a meaningful word itself. For example, the lemmas of the words *produced* and *producing* is *produce*, but their stem is *produc* because it is included in both words as such. In English, lemma and stem are often the same, but in other languages like Finnish they can differ more often.

Both stemming and lemmatization depend on language, and there exists plenty of different stemming algorithms. Stemming recognizes known suffixes of the words (e.g., *-ing*), and then chops the suffix off to obtain the stem. It has been widely used in keyword extraction [1,2,6,11,15,16,18] despite the drawback that the stem is not always a real word. This does not matter for algorithms but makes it less appealing for humans. According to [3], about 10% of English words would become non-real by stemming. The reasons for using stemming are that it is fast and easy to implement and does not require any dictionary. Many stemming algorithms for English exist, including *Porter stemmer*, *Snowball stemmer* and *Lancaster stemmer* [3]. Despite its better accuracy, lemmatization is less commonly used [1,14] than stemming mainly because it requires a dictionary.

Another common pre-processing method is stop word removal. *Stop words* are the most common words in the language, and they should therefore not be selected as keywords even if their frequency were high. Stop word lists must be built for each language separately. However, they are usually short (from a few dozen to a few hundred), and lists for many languages exist on the web[1].

Table 1. Summary of the pre-processing methods used.

Method	**References**	**Language dependency**	**Example**
Remove numbers and punctuation marks	1, 2, 8, 18	No	Numbers: 1,2,3 Punctuation marks: . , ? ! : ; “ & / =
Stemming	4, 8, 11, 9, 16, 18, 19	Yes	Original: *programs, programming, programmer, goes, corpora, studies* Stemmed: *program, program, program, goe, corpora, studi*
Lemmatization	1, 14	Yes	Original: *programs, programming, programmer, goes, corpora, studies* Lemmatized: *program, programming, programmer, go, corpus, study*
Stop words removal	1, 2, 4, 5, 8, 9, 12, 16, 18, 19, 36	Yes	English: *the*, *and*, *for*, *is*, *was* Finnish: *kuin* [as], *mina* [I], *hänen* [his], *että* [that]

[1] https://www.ranks.nl/stopwords

2.2. Candidate selection

Extracted keywords are words found in the web page, but which words should we consider as the candidates in the first place? Table 2 summarizes typical methods. Tokenization breaks each sentence into smaller units called tokens, which are usually the words. It is by far the most used method. Exceptions might be languages like Chinese, in which words describing a specific meaning are usually composed of two or more Chinese characters. Mere tokenization is therefore not enough. Unlike English and most Western languages, Chinese text lacks clear word separators. The *Jieba* tool has been used for Chinese text segmentation in [5].

It is also possible to use a pre-defined dictionary of possible words called a *bag of words* [6]. Then, only those candidate words existing in this dictionary are considered. The method in [7] relies on Wikipedia and string matching without the need for explicit tokenization.

POS tags and patterns have also been utilized. Typical keywords are nouns, and for this reason, many methods select only nouns as keywords. Some methods also allow adjectives [1,8,9,12,13,16] and some verbs [9,16]. To limit the keywords according to its POS tags is a bit na ïve, but it can improve the accuracy of simple baseline methods according to [10].

The problem is often considered as extraction of *key phrases* instead of single words. Combinations of noun + noun, noun + adjective and noun + verb have been considered [8,16,19,36]. *N-grams* are any fixed-length sequences of words and used for key phrase extraction in [3,6–8,15, 19,21,22,36]. *NP-chunks* are variable length sequences of words and differ from n-grams in that only pre-prepared combinations extracted using regular expressions are allowed [8]; see Figure 3.

Figure 4 summarizes the pre-processing and tokenization steps for a sample web page producing 23 candidate words. Four candidates have frequency of 2: *accessibility*, *BBC*, *homepage* and *victim*. Simple frequency is not enough to select the keywords, so the next step is to evaluate these candidates.

Table 2. Summary of the approaches for candidate generation.

Method	**References**	**Language dependency**	**Example**
Tokenize	1, 2, 4, 8, 9, 11, 12, 14, 15, 16, 18	No	Original: This is text Tokens: *This*, *is*, *text*
Nouns	8, 9, 12, 13 15, 16, 19	Yes	Original: people like to play best games. Nouns: *people*, *games*
Adjectives	7, 8, 13, 15, 16, 19	Yes	Original: people like to play best games. Adjectives: *best*
Verb	15, 16	Yes	Original: people like to play best games. Verb: *play* All other methods ignore verb as a candidate keyword
POS patterns	8, 16, 19, 36	Yes	Noun + Noun Noun + Adjective Noun + Verb
N-grams	3, 6, 7, 8, 15 19, 36	No	Compounds of multiple words. Special cases of n-grams include unigrams (n = 1), bigrams (n = 2). In [7], only lengths of n≥5 were considered.
NP-chunks	8	Yes	Chunking involves taking small pieces of information and grouping them into larger chunks.

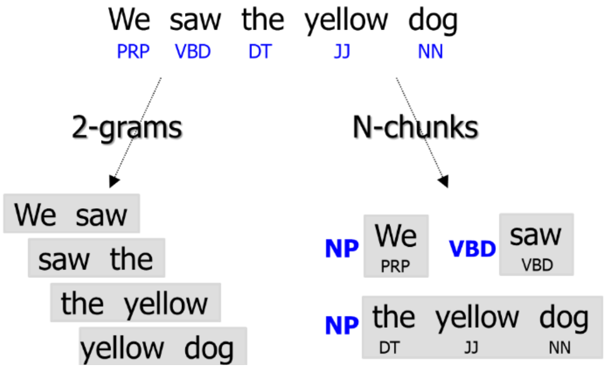


Figure 3. Examples of the N-grams (n = 2) and N-chunks (n is variable length) approaches.

WEBPAGE TEXT

BBC - Homepage

Homepage Accessibility links Skip to content Accessibility Help
BBC Account Notifications
French clergy abused 216,000 $ victims since 1950
The Church asks for forgiveness as an inquiry says it treated victims with "cruel indifference".
Europe cars

TOKENS

‘BBC’, ‘-‘, ‘Homepage’, ‘Homepage’ , ‘Accessibility’, ‘links’, ‘Skip’, ‘to’, ‘content’, ‘Accessibility’, ‘help’, ‘BBC’, ‘Account’, ’Notifications’, ‘French’, ‘clergy’, ‘abused’, ’216,000’, ‘$’, ‘victims’, ‘since’, ‘1950’, ‘The’, ‘Church’, ‘asks’, ‘for’, ‘forgiveness’, ‘as’, ‘an’, ‘inquiry’, ‘says’, ‘it’ ,‘treated’, ‘victims’, ‘with’, ‘"’,’cruel’, ‘indifference,’"’, ‘Europe’, ‘cars’. (**41**)

REMOVE NUMBERS AND PUNCTUACTION MARKS

BBC, Homepage, Homepage, Accessibility, links, Skip, to, content, Accessibility, help, BBC, Account, Notifications, French, clergy, abused, victims, since, The, Church, asks, for, forgiveness, as, an, inquiry, says, it, treated, victims, with, cruel, indifference, Europe, cars. (**35**)

REMOVE STOPWORDS

BBC, Homepage, Homepage, Accessibility, links, Skip, content, Accessibility, help, BBC, Account, Notifications, French, clergy, abused, victims, since, Church, asks, forgiveness, inquiry, says, treated, victims, cruel, indifference, Europe, cars. (**29**)

STEMMING

bbc, homepage, homepage, access, link, skip, content, access, help, bbc, account, notif, french, clergi, abus, victim, sinc, church, ask, forgiv, inquiri, say, treat, victim, cruel, indiffer, europ, car. (**7** non-words)

LEMMATIZATION

BBC, Homepage, Homepage, Accessibility, link, Skip, content, Accessibility, help, BBC, Account, Notifications, French, clergy, abused, victim, since, Church, asks, forgiveness, inquiry, say, treated, victim, cruel, indifference, Europe, car. (**1** mistake)

CANDIDATE GENERATION SEPARATE NOUNS

BBC, Homepage, Homepage, Accessibility, link, Skip, content, Accessibility, help, BBC, Account, Notifications, clergy, victim, Church, asks, forgiveness, inquiry, victim, cruel, indifference, Europe, car. (**23**)

Figure 4. Complete example of candidate generation process.

3. Statistical features

The most common feature is *term frequency* (TF), which simply selects the most common words in the web page. It has been used by many [2,4,5,8,13–15,19,27] because it is easy to implement by counting the appearances of the words in the page. Its main drawback is that the same words tend to be popular in all documents.

Removing stop words can compensate this deficiency, but this problem can also be attacked statistically using the so-called *inverse document frequency* (IDF). It counts how many documents contain the word. It helps to estimate the importance of the word so that a word that is frequent in all documents is less likely to be chosen. Vice versa, a word that is frequent only in the current web page is more likely to be a keyword. TF-IDF refers the joint use of TF and IDF.

For the BBC example in Figure 4, we get TF = 2 values for *BBC*, *Homepage*, *Accessibility* and *Victim*; and TF = 1 for the other candidates. We estimate their IDF-values by the number of Google search results they generate; see Table 4. *Victim* and *BBC* are the highest scoring words among those with TF = 2, and *clergy* and *indifference* among those with TF = 1. They all are potential keywords for this example. Wikipedia [11–14] and Bing search terms [19] have also been used for determining the IDF.

A more complex example using the *Formula 1* Wikipedia page is summarized in Table 5. TF-IDF helps, but we can also see that the combination *Formula one* would be a more meaningful key phrase instead of the single word *formula*. It would provide the highest scores: TF = 751, IDF-freq = 31, TF-IDF = 7262. The example also shows that the role of pre-processing is crucial. Overall, term frequency with IDF seems to work reasonably well with these examples.

Another statistical feature found in literature is the *first occurrence*, which is just the running index of the first appearance of a word in the document. The idea is that more important words appear earlier than the less important ones. The statistical features are summarized in Table 3.

Table 3. Summary of the statistical features.

Feature	**References**	**Type**	**Description**
Term Frequency (TF)	2, 4, 5, 8,13, 14, 15, 19, 36, 27	Numeric	The number of times term appears in a web page.
Inverse document frequency (IDF)	2, 4, 14, 19, 36	Numeric	The number of documents containing the word relative to all documents. Result is in log scale (-log n/N).
TF-IDF	2, 4, 10, 13, 14, 15, 36	Numeric	Product of the above two: TF-IDF = TF*IDF.
First occurrence of the word	4, 15, 36	Numeric	Location of the first appearance of the word. Integer number between 1 and the number of words in the page.

Table 4. Example TF-IDF calculations for the BBC example in Figure 4.
We used “the” word for the estimation of all documents, giving N = 25,270.

Page	**TF**	**IDF-freq.**	**TF-IDF**
victim	2	1,600	8.0
BBC	2	3,170	6.0
homepage	2	8,350	3.2
accessibility	2	13,550	1.8

clergy	1	94	8.1
indifference	1	165	7.3
forgiveness	1	461	5.8
cruel	1	633	5.3
church	1	2620	3.3
Other words: link (25270), content (25270), help (25270), account (19350), skip (12000), asks (9680), Europe (8090), car (5200), inquiry (3810), notifications (3670).			

Table 5. Example of normalization of the frequencies using data from Wikipedia.

Original text				Pre-processed text			
Word	**TF**	**IDF**	**TF-IDF**	**Word**	**TF**	**IDF**	**TF-IDF**
The	1,222	25,270	0	formula	320	6,710	612
.	605	n/a	-	one	268	6,310	536
of	469	25,270	0	championship	160	1,190	705
and	434	25,270	0	race	155	6,830	292
to	427	25,270	0	retrieved	153	10,520	193
in	365	25,270	0	prix	136	2,370	464
Formula	320	6,710	612	f1	135	1,450	557
a	269	25,270	0	drivers	122	3,910	328

4. Structural features

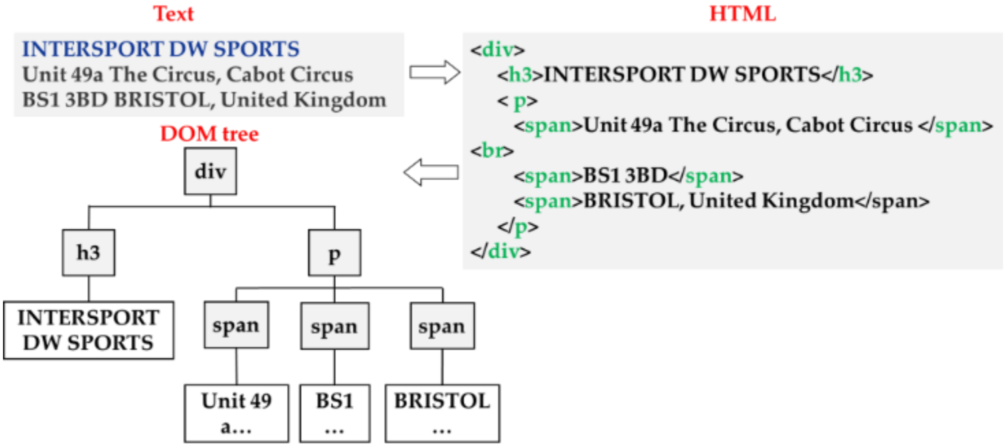
Structural features consider how the words are presented in the web page. Keywords are expected to have a stronger visual emphasis than normal words and therefore more often be used with the header tags (<h1> to <h6>) and within the title tag (<title>). A title tag is important for search bots, and therefore keywords are often added inside for that purpose. Keywords are often capitalized, either just the first letter or the entire word.

Keywords tend to appear also in the URL of the web page. We separate it into three meaningful parts: *host*, *path* and *query*. The host is the name of the web site, path is the directory structure used in the link, and query is the name of the actual web page. For example, the page[2] has candidate words *University Herald* in the host, *articles* in the path and the words *poor*, *britain*, *salt*, *rich*, *warvick*, *socio*, *economic* in the query part.

A *document object model* (DOM) is a tree-structured representation of the web page based on tags like <head>, <div>, <a> and <h1>. It divides the page into segments that can provide additional clues about the importance of the words; see Figure 5. The method in [16] assumes that the most important information appears in the beginning of the document and therefore analyzes only the first twenty DOM nodes. The method in [1] counts the number of DOM nodes in which the word appears. This assumes that less important words appear only locally in one node, whereas the keywords are more widely spread in the page.

[2] http://www.universityherald.com/articles/11104/20140827/poor-britain-salt-rich-warvick-socio-economic.htm

Other signs of importance are anchor tags (<a>) and meta tags (<meta>). Anchor tags are links to other web pages, while meta tags include additional information about the technical content of the page such as the character set and page description, but they also used for storing keywords. The most common structural features are summarized in Table 6.



```
<div>
   <h3>INTERSPORT DW SPORTS</h3>
   < p>
      <span>Unit 49a The Circus, Cabot Circus </span>
<br>
      <span>BS1 3BD</span>
      <span>BRISTOL, United Kingdom</span>
   </p>
</div>
```

Figure 5. Example of a piece of HTML code and the corresponding DOM tree.

Table 6. Summary of the structural features.

Feature	**References**	**Type**	**Description**
Header tags <h1>…<h6>	3, 5, 7	Binary / Numeric	Count of how many times a word uses <h1>…<h6>.
Part of URL	7, 19, 28	Binary	Whether the word is a part of the URL. Examples: Host: http://**bbc**.com Path: https://aimspress.com/journal/**aci** Query: https://www.bbc.co.uk/search?q=**queen**
Title: <title>	5, 7, 13, 19, 28	Binary / Numeric	Whether used in title tag or not (or count if repeated).
Anchor tag: <a>	14, 28	Numeric	Count of how many times inside an anchor tag.
Span & Meta tags: <meta> <span>	7, 13, 15, 28	Numeric	Count of how many times inside Span or Meta tag.
Capital initial char word	12	Numeric	Count of how many times the first letter is Capitalized. Examples: Car, Employee.
Capital all char word	12	Numeric	Count of how many times the entire word is Capitalized. Examples: CAR, EMPLOYEE
DOM	18, 19, 22, 23, 24, 25, 26, 16	Numeric	DOM tree represents the hierarchy of the page providing ways to analyze the relative location of the words; how early in the page, or how widely distributed.

5. Linguistic features

Language can be a very powerful tool to guide the keyword extraction, and linguistic features have been shown to significantly improve frequency-based methods [6,13]. For example, synonyms of the words have been utilized in [1,3,9,15]. A simple approach is to merge the counts of synonyms to get more reliable estimation of the important concepts in the document. The method in [1,12] does the opposite and assumes that important concepts are presented by the same keywords throughout the page for consistency, whereas synonyms are used more often for less important concepts to create variation. Chinese synonyms were used in [5], and *FarsNet* for Persian language was used in [2].

Semantic similarity takes the idea of synonyms further. The words do not need to have exactly the same meaning, but also words with similar meanings (*car*, *taxi*, *truck*) can increase their joint importance. However, the semantic meaning may differ depending on the context like "*call me a cab*" and "*My name is Cab*", so the use of semantics is not trivial. *Semantic relatedness* is also considered based on co-occurrence of the words. In [17], two words that occur frequently together can make a key-phrase.

The approach in [19] recognizes *named entities*, which are given higher emphasis in the evaluation. This is understandable, as named entities such as persons and places are often used as keywords in newspaper articles. The method in [7] builds a domain-specific concept hierarchy based on Wikipedia, and keywords are matched to these concepts. Starting from seed keywords, the method in [11] constructs a concept graph iteratively using Wikipedia's internal links. This graph is used when not enough keywords are found on a short-text page.

Parts of speech were listed in Section 2.2 already as a candidate word selection method. However, instead of a binary choice (to include or not), POS tags can also be used as a feature in the scoring process. They can be useful especially with trained classifiers but would require a good tagger. Several good taggers exist for the English language, but the accuracy for languages like Finnish with complex grammar is much weaker. The main drawback of using POS tags is that it makes the keyword extraction language dependent [20]. The most common linguistic features are listed in Table 7.

Table 7. Summary of the linguistic features.

Feature	**References**	**Type**	**Description**
Synonyms	1, 3, 5, 7, 9, 15, 19	String	Consider synonyms as evidence of the same keyword. For example: *internet-net*, *see-watch-look*.
Semantic similarity	13, 15, 19	Numeric	For example, car and cars are semantically related.
Co-occurrence	17	Numeric	Relationship between words is calculated by their distance in the document.
Named entity	19	String	Named entities such as people, organizations, and places. For example: "*Apple is selling iPhone in Europe*" include two such keywords: *Apple* (organization) and *Europe* (location).
Wikipedia	7, 11, 13, 14	Numeric / String	Wikipedia is used for creating concept hierarchy or graph.
POS tags	1, 8, 9, 15, 28	String	Parts of speech (POS) tags. Example: "*People play games*" have tags *people* = Noun, *play* = verb, *games* = noun.

6. Experiments

We next study the performances of the different components to find out which of them matters most. We use *f-score* based on standard *precision* and *recall*:

$$F\text{-}score = 2 \cdot precision \cdot recall \, / \, (precision + recall) \tag{1}$$

Precision and recall are counts of correctly extracted keywords relative to all ground truth keywords (precision) and relative to all extracted keywords (recall). The higher the precision, the more correct keywords were found; and vice versa, the higher the recall, the less incorrect keywords were given. We refer to the f-score as *hard measure* in the rest of this paper. In all experiments, we extract 5 keywords for Mopsi datasets and 10 keywords for the rest.

The hard measure recognizes only exact matches and may not give a realistic picture of the performance. For example, consider the ground truth {students, **university**, tuition, opportunities} and the extracted keywords {study, **university**, lecture, chances, fees}. Not only do the number of keywords differ, but there is only one exact match despite the result otherwise being good. For this reason, we also use the soft variant of precision and recall [34]. We refer to this as *soft measure* and use it for the final comparison in Table 13 to get better understanding of the real accuracy level.

6.1. Datasets used

We used twelve datasets, summarized in Table 8. They are mainly collected from English and Finnish newspaper web sites but also German web sites and user-collected web pages in the *Mopsi services* platform. The newspaper web pages have ground truth keywords stored in their meta tags, annotated by the media itself for journalistic use. These web pages have uniform structure, which makes them easier to process. The German and Mopsi have more variations. The ground truth keywords in the Mopsi datasets have been manually annotated and sometimes do not even exist in the web page as such. The keywords may also use both English and Finnish in a mixed manner, which makes the datasets more challenging.

Figure 6. Summary of the datasets used.

Statistics of the data are summarized in Table 8. The average numbers of keywords in the cases of the newspaper datasets are **9.5** (English) and **7.8** (Finnish), with **16.2** in the case of the German datasets, but only **2.5** in the case of the Mopsi datasets. The latter two datasets also have a lot of annotated keywords that do not appear in the web page: German (64%), Mopsi (48%). In the newspaper web pages, the number of non-present keywords is low, usually below 10%.

Table 8. Summary of the data sets[3].
We will later abbreviate the sets by their first two letters (GU = Guardian, HE = Herald, and so on).

Language	Name	Data source	Pages	Keywords (average)	Keywords not in text	Stop-words
English	Guardian	theguardian.com	421	13.4	12.3%	7.3%
	Herald	universityherald.com	300	9.0	9.9%	2.1%
	Indian	indianexpress.com	329	6.1	6.3%	1.4%
	Mac	macworld.com	204	7.5	1.4%	1.4%
Finnish	Kaksplus	kaksplus.fi	200	5.4	4.3%	0.6%
	Kotiliesi	kotiliesi.fi	210	6.5	5.0%	0.3%
	Ruoka	ruoka.fi	200	7.4	2.5%	1.1%
	Taloussanomat	taloussanomat.fi	210	9.8	5.8%	0.7%
	Urheilulehti	urheilulehti.fi	200	6.6	10.8%	0.3%
	Uusisuomi	uusisuomi.fi	200	10.9	8.3%	0.6%
German	German	*multiple URLs*	81	16.2	63.8%	6.3%
English & Finnish	Mopsi	*multiple URLs*	381	2.5	47.9%	0.1%

6.2. Statistical features

Results for term frequency are summarized in Table 9, with and without pre-processing and stop word removal. Our first observation is that the pre-processing is essential to useful results by the statistical features alone. The results are still rather modest though. Our second observation is that stop word removal works equally well to TF-IDF. For the Finnish and German datasets, we used two stop word lists: English and the primary language of the web page (Finnish or German).

Table 9. Accuracy of the statistical measures. *TF+SW+PP is used as our baseline later.
TF = term frequency, SW = stop word removal, PP = preprocessing

	GU	HE	IN	MAC	KA	RU	UR	UU	KO	TA	Mopsi	GER	**Average**
TF	0.02	0.06	0.02	0.02	0.05	0.07	0.02	0.01	0.05	0.03	0.01	0.01	**0.01**
TF + SW	0.08	0.03	0.03	0.04	0.09	0.07	0.02	0.02	0.02	0.03	0.02	0.02	**0.02**
TF + SW + PP*	**0.20**	**0.26**	**0.24**	**0.27**	**0.16**	**0.21**	**0.20**	**0.09**	**0.18**	**0.18**	**0.10**	**0.17**	**0.15**
TF-IDF + SW + PP	0.23	0.26	0.25	0.20	0.16	0.21	0.20	0.09	0.19	0.19	0.10	0.17	**0.15**
TF-IDF + Wikipedia	0.15	0.42	0.22	0.20	-	-	-	-	-	-	-	-	**-**

[3] http://cs.uef.fi/mopsi/MopsiSet/, http://cs.uef.fi/mopsi/newspaper/, http://cs.uef.fi/mopsi/newspaper/GermanSet/

6.3. Candidate selection and other features

Next, we test the impact of the other features. Results are summarized in Table 10. Some variants are tested only with English datasets. Here, we observe that the individual formatting features have only a minor effect on the result; but when used together, they improve the f-score from 0.16 to 0.19. Title and URL seems to have the biggest impact among the individual features.

Linguistic features improved the accuracy on English datasets remarkably, from 0.27 to 0.33, on average. Among different features, using only nouns and no-synonyms improve the most. Stemming and lemmatization were counter-productive and decreased the performance. However, this might be partly due to the evaluation method (hard measure) requiring exact match. As soon as the words are stemmed or lemmatized, their original forms change. In the case of the English datasets, we also tested the named entity feature. We determined whether the word refers to a place, person or organization. This feature improved the baseline method but not when combined with other features.

6.4. Summary of results

Based on the results, we construct two combinations in this paper: *baseline* (see Table 9) and the best performing combination, called *ACI-rank* (see Table 11). Frequency is used as such (baseline) and with IDF-value from Wikipedia (ACI-rank). Then no-synonym feature is a binary feature with only values 0 and 1. The rest of the features are the counts of the appearance of the feature. The scoring is simply the sum of the counts as such. The results are compared against the existing method, summarized in Table 12.

Table 10. Accuracy of the statistical measures: Baseline = TF + SW + PP.

	GU	HE	IN	MAC	KA	RU	UR	UU	KO	TA	Mopsi	GER	**Average**
Formatting features													
Baseline	0.23	0.26	0.25	0.20	0.16	0.21	0.20	0.09	0.18	**0.09**	0.10	0.17	0.16
Base + <H1><H2><H3>	0.16	0.31	0.26	0.24	0.15	0.21	0.17	0.09	0.17	0.08	0.10	0.17	0.16
Base + Title	0.16	0.43	0.24	0.20	0.13	0.20	0.12	0.08	0.12	0.08	0.12	**0.20**	0.17
Base + URL host + query	0.18	0.39	0.25	0.21	0.17	0.21	0.21	0.09	0.19	**0.09**	0.12	**0.20**	0.18
Base + **Bold** + *italic*	0.17	0.39	0.26	0.14	0.17	0.22	0.13	0.04	0.14	0.06	0.11	0.18	0.16
Base + Cap + UPPER	0.18	0.42	0.25	0.23	0.16	0.20	0.13	0.08	0.10	0.04	0.10	0.17	0.16
Base + All format features	0.21	0.40	0.26	0.20	**0.19**	**0.23**	**0.16**	**0.08**	**0.22**	0.06	**0.16**	0.19	**0.19**
Linguistic features													
Base + format + Stem	0.14	0.34	0.18	0.10	0.11	0.13	0.06	0.07	0.14	0.06	0.06	0.11	0.11
Base + format + Lemma	0.16	0.40	0.16	0.18	0.14	0.15	0.13	0.08	0.13	0.06	0.10	0.16	0.15
Base + only (N)	0.16	0.36	0.26	0.25	-	-	-	-	-	-	-	-	-
Base + (N) +(V) + (A)	0.17	0.44	0.26	0.26	-	-	-	-	-	-	-	-	-
Base + (N) + NoSyn	0.21	0.51	0.25	0.27	-	-	-	-	-	-	-	-	-
Base + (N) + NamedEntity	0.17	0.34	0.22	0.25	-	-	-	-	-	-	-	-	-
Base+ Format+(N)+NoSyn	**0.22**	**0.53**	**0.30**	**0.25**	-	-	-	-	-	-	-	-	-

Table 11. Summary of components used in the proposed ACI-rank method.

Structural (DOM)		
1	H1	Appearance in <h1> tag
2	H2	Appearance in <h2> tag
3	H3	Appearance in <h3> tag
4	Title	Appearance in <title> tag
5	URL-Host	Appearance in host part of URL
6	URL- Query	Appearance in query part of URL
7	Capital	The word appears to be capital
8	Upper	The word appears to be upper
9	Bold	The word appears to be bold
10	Italic	The word appears to be italic
Linguistic		
11	No-Synonym word (WordNet)	Word Appearance in the list of No-Synonym words
12	Named Entity	Named Entity: Person, Organization, Location.
Statistical		
13	Term frequency (TF)	Word frequency in the text
14	TF-IDF score (Wiki)	Score of a word in Wikipedia's TF-IDF

We compare our baseline and the proposed ACI-rank against existing methods shown in Table 12. The results in Table 13 are summarized so that *News1* is the average of the four English newspaper datasets, and *News2* the average of the six Finnish newspaper datasets. Soft evaluation results are also provided, as they provide a more realistic view of how good (or bad) the methods really are.

According to the soft measure, the proposed ACI-rank works best among the unsupervised methods (0.47) and close to the supervised approach, WebRank (0.44). In case of well-structured newspaper datasets, WebRank is better, whereas the proposed method is clearly superior on the most heterogenous Mopsi datasets. We also see that the difference from the mere frequency-based baseline method (0.37) is significant. It shows that the web HTML-based structural features are important.

It is also worth noting that the results with Finnish newspaper datasets were significantly worse than those of English newspaper data because the linguistic features were not used. Notable differences were seen with results of KeyBert, Yake and WebRank. However, the result of the frequency-based baseline deteriorated even it had access to stop words of Finnish and German and did not use any other linguistic feature.

Table 12. Existing methods from literature.

	TextRank	**Yake**	**KeyBert**	**CL-rank**	**D-rank**	**H-rank**	**WebRank**
Data used	Text	Text	Text	Text	Text + DOM	Text	Text + DOM
Language	English	English	English	English	Any	English	Any
Pre-processing	Stem+ lemma	Stem+ lemma	Stem+ lemma	Stem+ lemma	-	Stem+ lemma	-
Frequency	TF	TF	TF	TF / cluster	TF / position	TF / cluster	TF / position
Linguistic features	Nouns	Nouns + Adj+Verbs	Nouns	Nouns	-	Nouns + Adj+Verbs	-
WordNet	Synonyms	Synonyms	-	Synonyms	-	Synonyms	-
Supervised	-	-	-	-	-	-	Yes

Table 13: Comparison to existing methods. Red refers to the best overall result, and blue refers to best result among the unsupervised methods.

	Hard measure					**Soft measure**				
	News1	News2	Mopsi	GER	**Average**	News1	News2	Mopsi	GER	**Average**
TextRank [28]	0.23	0.07	0.05	0.11	0.12	0.44	0.36	0.19	0.37	0.34
Yake [29]	0.18	0.09	0.03	0.10	0.10	0.45	0.30	0.12	0.30	0.29
KeyBert [35]	0.10	0.08	0.09	0.05	0.08	0.28	0.16	0.14	0.11	0.16
CL-rank [1]	0.29	0.14	0.06	0.12	0.15	0.49	0.39	0.19	0.37	0.36
D-rank [18]	0.30	0.13	0.12	**0.21**	0.19	0.49	**0.41**	0.25	**0.45**	0.40
H-rank [9]	0.22	-	-	-	-	0.53	-	-	-	-
WebRank [30]	**0.40**	**0.26**	0.04	**0.21**	**0.23**	0.60	**0.47**	0.23	**0.46**	0.44
Baseline (**new**)	0.23	0.15	0.10	0.17	0.16	0.51	0.32	0.25	0.38	0.37
ACI-rank (**new**)	**0.33**	**0.16**	**0.14**	0.18	**0.20**	**0.68**	0.39	**0.36**	0.44	**0.47**

7. Conclusions

We have studied keyword extraction from web pages. Simple term frequency with stop word removal works reasonably, but pre-processing is important. Average results of the frequency-based baseline were 0.16 (hard) and 0.37 (soft). Further improvement was achieved by adding formatting and linguistic features, with the average results of 0.20 (hard) and 0.47 (soft). The new method, called ACI-rank, reaches the best results and is rather close to a supervised method (0.23 and 0.44). We expect that it can be improved even further by adding some of the more sophisticated ideas like concept graphs.

Future work includes applying clustering based on semantic or syntactic similarity [31] instead of the simple no-synonyms approach. Ideas from other summarization tasks, like title extraction [32] and representative image selection [33], could also be adopted to improve keyword extraction or to construct a complete content summarization that would cover all these three tasks. Many components used were rather simple, and the scoring of their combination was a bit naïve. We simply did not find significantly better combinations, and significant further improving seemed to require a machine learning based training approach. This is also a point of future work.

Conflict of interest

All authors declare that there is no conflict of interests in this paper.

References

1. M. Rezaei, N. Gali, P. Fränti, CLRank: A method for keyword extraction from web pages using clustering and distribution of nouns, *IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology* (*WI-IAT*), **1** (2015), 79–84. https://doi.org/10.1109/WI-IAT.2015.64
2. S. Lazemi, H. Ebrahimpour-Komleh, N. Noroozi, PAKE: a supervised approach for Persian automatic keyword extraction using statistical features, *SN Appl. Sci.*, **1** (2019), 1–4. https://doi.org/10.1007/s42452-019-1627-5
3. S. Vijaya Shetty, S. Akshay, S. Reddy, H. Rakesh, M. Mihir, J. Shetty, Graph-Based Keyword Extraction for Twitter Data, *Emerging Research in Computing, Information, Communication and Applications*, (2022), 863–871. https://doi.org/10.1007/978-981-16-1342-5_68
4. B. Armouty, S. Tedmori, Automated keyword extraction using support vector machine from Arabic news documents, *IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology* (*JEEIT*), (2019), 342–346. https://doi.org/10.1109/JEEIT.2019.8717420
5. P. Sun, L. Wang, Q. Xia, The Keyword Extraction of Chinese Medical Web Page Based on WF TF IDF Algorithm, *Ininternational conference on cyber enabled distributed computing and knowledge discovery* (*CyberC*), (2017), 193–198. https://doi.org/10.1109/CyberC.2017.40
6. A. Onan, S. Korukoğlu, H. Bulut, Ensemble of keyword extraction methods and classifiers in text classification, *Expert Syst. Appl.*, **57** (2016), 232–247. https://doi.org/10.1016/j.eswa.2016.03.045
7. W. Zhang, D. Wang, G. R. Xue, H. Zha, Advertising Keywords Recommendation for Short

Text Web Pages using Wikipedia, *ACM T. Intel. Syst. Tec.*, **3** (2012), 1–25. https://doi.org/10.1145/2089094.2089112

8. A. Hulth, Improved automatic keyword extraction given more linguistic knowledge, *Proceedings of the conference on empirical methods in natural language processing*, (2003), 216–223. https://doi.org/10.3115/1119355.1119383
9. H. Shah, M. U. Khan, P. Fränti, H-rank: a keywords extraction method from web pages using POS tags, *IEEE 17th International Conference on Industrial Informatics* (*INDIN*), **1** (2019), 264–269. https://doi.org/10.1109/INDIN41052.2019.8972331
10. D. Khyani, B. S. Siddhartha, N. M. Niveditha, B. M. Divya, An Interpretation of Lemmatization and Stemming in Natural Language Processing, *Journal of University of Shanghai for Science and Technology*, **22** (2021), 350–357.
11. Nie H, Yang Y, and Zeng D, Keyword Generation for Sponsored Search Advertising: Balancing Coverage and Relevance, *In IEEE intelligent systems*, vol. 34, number 5, pp. 14–24, 2019. https://doi.org/10.1109/MIS.2019.2938881
12. O. Alqaryouti, H. Khwileh, T. Farouk, A. Nabhan, K. Shaalan, Graph based keyword extraction, *Intelligent natural language processing: trends and applications*, **740** (2018), 159–172. https://doi.org/10.1007/978-3-319-67056-0_9
13. W. Zhang, W. Feng, J. Wang, Integrating semantic relatedness and words intrinsic features for keyword extraction, *Twenty third international joint conference on artificial intelligence* (*IJCAI*'13), (2013), 2225–2231.
14. J. Xu, Q. Lu, Z. Liu, Aggregating skip bigrams into key phrase-based vector space model for web person disambiguation, *In KONVENS*, (2012), 108–117.
15. T. D. Nguyen, M. Y. Kan, Keyphrase extraction in scientific publications, *Proceedings of the 10th international conference on Asian digital libraries*, (2007), 317–326. https://doi.org/10.1007/978-3-540-77094-7_41
16. A. Gupta, A. Dixit, A. K. Sharma, A novel statistical and linguistic features-based technique for keyword extraction, *International conference on information systems and computer networks* (*ISCON*), (2014), 55–59. https://doi.org/10.1109/ICISCON.2014.6965218
17. D. Cai, S. Yu, J. R. Wen, W. Y. Ma, VIPS: a vision-based page segmentation algorithm, *In Microsoft technical report*, MSR-TR-2003-79, 2003.
18. H. Shah, M. Rezaei, P. Fränti, DOM based keyword extraction from webpages, *In proceedings of international conference on artificial intelligence, information processing and cloud computing* (*AIIPCC*), (2019), 1–6. https://doi.org/10.1145/3371425.3371495
19. P. Liu, J. Azimi, R. Zhang, Automatic keywords generation for contextual advertising, *In Proceedings of the 23rd International Conference on World Wide Web*, (2014), 345–346. https://doi.org/10.1145/2567948.2577361
20. S. Siddiqi, A. Sharan, Keyword and keyphrase extraction techniques: a literature review, *In international journal of computer applications*, **109** (2015), 18–23. https://doi.org/10.5120/19161-0607
21. M. Grineva, M. Grinev, D. Lizorkin, Extracting key terms from noisy and multi-theme documents, *In Proceedings of the 18th international conference on World Wide Web*, (2009), 661–670. https://doi.org/10.1145/1526709.1526798
22. F. Lei, M. Yao, Y. Hao, Improve the performance of the webpage content extraction using webpage segmentation algorithm, *In proceedings of international forum on computer science-technology and applications*, (2009), 323–325.

https://doi.org/10.1109/IFCSTA.2009.84

23. D. Cai, S. Yu, J. R. Wen, W. Y. Ma, Extracting content structure for web pages based on visual representation, *In Asia-Pacific Web Conference*, (2003), 406–417. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-36901-5_42
24. G. Salton, C. S. Yang, C. T. Yu, A theory of term importance in automatic text analysis, *Journal of the American society for Information Science*, **26** (1975), 33–44. https://doi.org/10.1002/asi.4630260106
25. J. Pasternack, D. Roth, Extracting article text from the web with maximum subsequence segmentation, *Proceedings of the* 18*th international conference on world wide web*, (2009), 971–980. https://doi.org/10.1145/1526709.1526840
26. S. Gupta, G. Kaiser, D. Neistadt, P. Grimm, DOM-based content extraction of html documents, *Proceedings of the 12*th *international conference on World Wide Web*, (2003), 207–214. https://doi.org/10.1145/775152.775182
27. M. Krapivin, A. Autayeu, M. Marchese, E. Blanzieri, N. Segata, Keyphrases extraction from scientific documents: improving machine learning approaches with natural language processing, *International Conference on Asian Digital Libraries*, (2010), 102–111. https://doi.org/10.1007/978-3-642-13654-2_12
28. R. Mihalcea, P. Tarau, TextRank: Bringing order into texts, *In proceedings of* (*EMNLP04*) *conference on empirical methods in natural language processing*, (2004), 404–411.
29. R. Campos, V. Mangaravite, A. Pasquali, A. M. Jorge, C. Nunes, A. Jatowt, YAKE! Collection-Independent Automatic Keyword Extractor, *European conference on information retrieval*, **10772** (2018), 806–810. https://doi.org/10.1007/978-3-319-76941-7_80
30. H. Shah, R. Mariescu-Istodor, P. Fränti, WebRank: Language-Independent Extraction of Keywords from Webpages, *IEEE International Conference on Progress in Informatics and Computing* (*PIC*), (2021), 184–192. https://doi.org/10.1109/PIC53636.2021.9687047
31. N. Gali, R. Mariescu-Istodor, D. Hostettler, P. Fränti, Framework for syntactic string similarity measures, *Expert Syst. Appl.*, **129** (2019), 169–185. https://doi.org/10.1016/j.eswa.2019.03.048
32. N. Gali, R. Mariescu-Istodor, P. Fränti, Using linguistic features to automatically extract web page title, *Expert Syst. Appl.*, **79** (2017), 296–312. https://doi.org/10.1016/j.eswa.2017.02.045
33. N. Gali, A. Tabarcea, P. Fränti, Extracting Representative Image from Web Page, *In WEBIST*, (2015), 411–419. https://doi.org/10.5220/0005438704110419
34. P. Fränti and R. Mariescu-Istodor, Soft precision and recall. Manuscript. Software available from: https://cs.uef.fi/sipu/soft/SoftEval/
35. M. Grootendorst, KeyBERT: minimal keyword extraction with BERT. Available from: https://github.com/MaartenGr/KeyBERT.
36. A. Awajan, Keyword extraction from Arabic documents using term equivalence classes, *ACM Transactions on Asian and Low-Resource Language Information Processing* (*TALLIP*), **14** (2015), 1–18. https://doi.org/10.1145/2665077